tenon.io

Gutenberg

Accessibility Audit Report

Table of Contents

Summary and Methodology

At its core, Web accessibility is about ensuring that systems are designed to support users with disabilities to ensure unassisted access as well as access when aided by assistive technologies. Each of the core technologies of the Web have been designed to allow native platforms to access the underlying information necessary to allow client software to expose the information necessary for users to understand web content as well as the nature and operation of user interface controls. Improperly designed and developed websites introduce usability and operability problems that cumulatively decrease accessibility for users. This accessibility audit report provides details on 90 total issues logged against 16 representative samples of the tested system.

Methodology

Tenon's accessibility audit methodology used during the Gutenberg audit leverages a variety of automated and manual test techniques up to and including code review and inspection with the following platforms:

- JAWS on Windows with Internet Explorer and Firefox
- NVDA on Windows with Internet Explorer and Firefox
- Windows High Contrast Mode
- VoiceOver on macOS with Safari and Chrome

Detailed Findings by Content Type

Below is a detailed list of all of the issues uncovered during testing. The issues are organized by content type and each issue contains the following information:

- **Issue number & title:** allows for easy referencing of the issues as well as a concise summary of the specific issue.
- **Summary information:**
  - *Severity:* How bad the issue is, organized by Low, Medium, and High
  - *Components Affected:* Which component(s)s had the issue. In cases where the value is listed here as "Global", it is either an issue that exists in global code (such as templates or CSS) or the issue was pervasive enough to be considered sitewide in nature
  - *Affected Populations:* Which disability population(s) will have problems using the system due to this issue
- **Description:** a long-form description of the problem, including screenshot(s) where relevant
- **Issue Code:** A snippet of code (where relevant) that is causing the problem. Code samples have been simplified so that only the most important part of the violating code is shown.
- **Remediation Guidance:** A description of how to fix the problem, including recommended code
- **Relevant WCAG Success Criterion:** A listing of the specific WCAG Success Criterion that this issue maps to.
- **References:** Any relevant links at which you can get more information.

Note on the severity of issues

Among the more important things to understand about the Web Content Accessibility Guidelines (WCAG) version 2.0 is that "Level" is not an appropriate indicator of importance or priority. Although laypersons and persons in the legal field have seemed to indicate a preference for Level AA compliance, there are good reasons to disregard Level altogether when it comes to determining if a system is accessible or not. Severity, Volume, and Affected Populations are also important factors in judging accessibility.

So-called "minor" issues with Low severity can still create major problems when you consider their cumulative impact on usability. Fundamentally, accessibility is merely a more technical and more focused field within Usability and the impact to users more profound when assistive technologies are thrown into the mix. For these reasons, we strongly recommend against disregarding issues listed as having "Low" severity, as this judgment is only in the context of the issue at the micro level and not in context of the cumulative impact across the system or its contribution to other issues on the same component.

Content Type: Color

GUT-100 - Selected block outline has insufficient color contrast

- **Severity:** Medium
- **Affected Populations:** Low-Vision, Cognitively Impaired
- **Platform(s):** All/ Universal
- **Components affected:** Block Editing

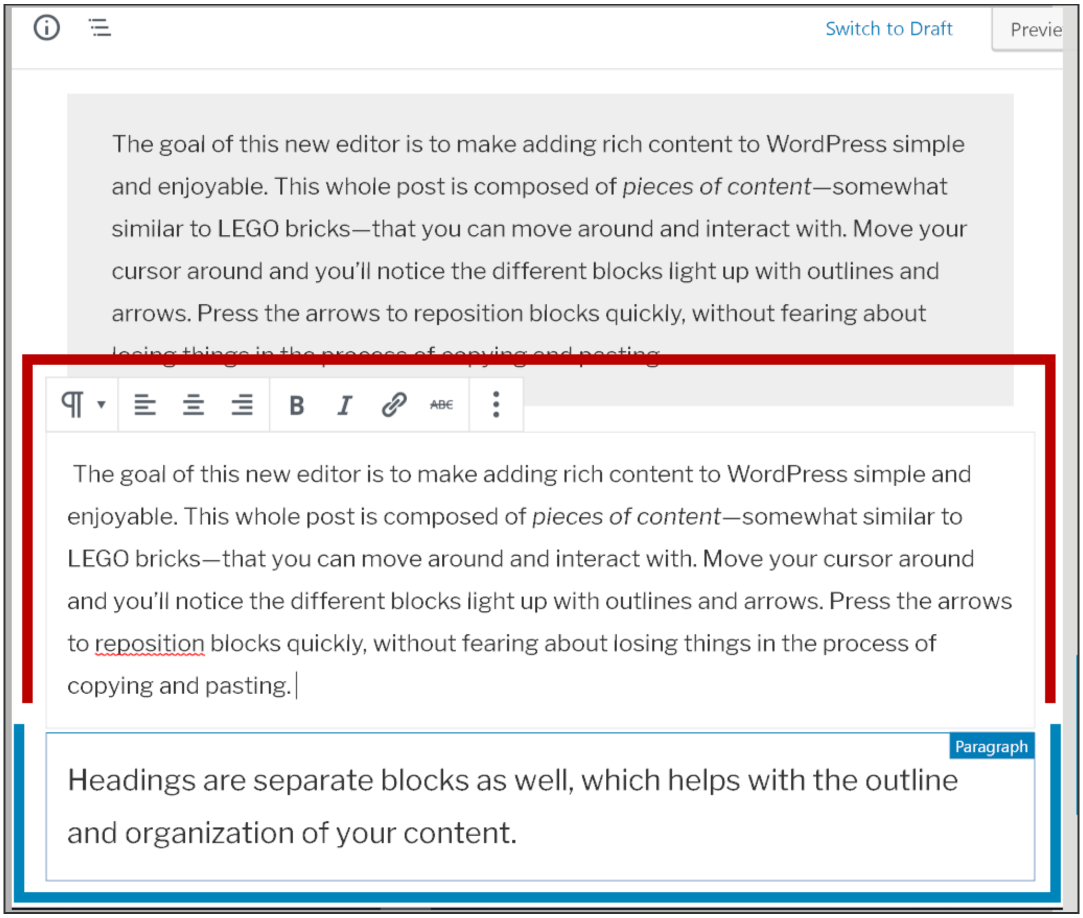
Issue description

When the mouse moves over a block, a bright blue outline appears. However when keyboard focus is in a block or any of that block's editing controls, the outline is a very light gray, the contrast of which is below the minimum threshold of 3:1 for interface controls:

- Rendered block border: #e3e5e7 (light gray) on #fff (white): **1.262:1**

This makes it more difficult for low-vision users to see which block they've moved into. If they're using the Arrow keys to move the cursor, the block toolbar does not immediately appear, and therefore the only way for users to identify which block they're in is this pale gray border, and the text cursor.

Sufficient color contrast is important for users who have low-vision or are color-blind, because content with a low contrast ratio may be difficult or impossible for such users to see.



The goal of this new editor is to make adding rich content to WordPress simple and enjoyable. This whole post is composed of *pieces of content*—somewhat similar to LEGO bricks—that you can move around and interact with. Move your cursor around and you'll notice the different blocks light up with outlines and arrows. Press the arrows to reposition blocks quickly, without fearing about losing things in the process of copying and pasting.

The goal of this new editor is to make adding rich content to WordPress simple and enjoyable. This whole post is composed of *pieces of content*—somewhat similar to LEGO bricks—that you can move around and interact with. Move your cursor around and you'll notice the different blocks light up with outlines and arrows. Press the arrows to reposition blocks quickly, without fearing about losing things in the process of copying and pasting.

Headings are separate blocks as well, which helps with the outline and organization of your content.

Issue Code

```
.editor-block-list__layout .editor-block-list__block.is-selected>.editor-block-list__block-edit::before {
    outline: 1px solid rgba(145,151,162,.25);
}
```

Remediation Guidance

Darken the `outline` color to achieve a minimum contrast ratio of 3:1 relative to the white background.

Recommended Code

```
.editor-block-list__layout .editor-block-list__block.is-selected>.editor-block-list__block-edit::before {
    outline: 1px solid #949494;
}
```

Relevant standards

- 1.4.11 Non-text Contrast (Level AA)

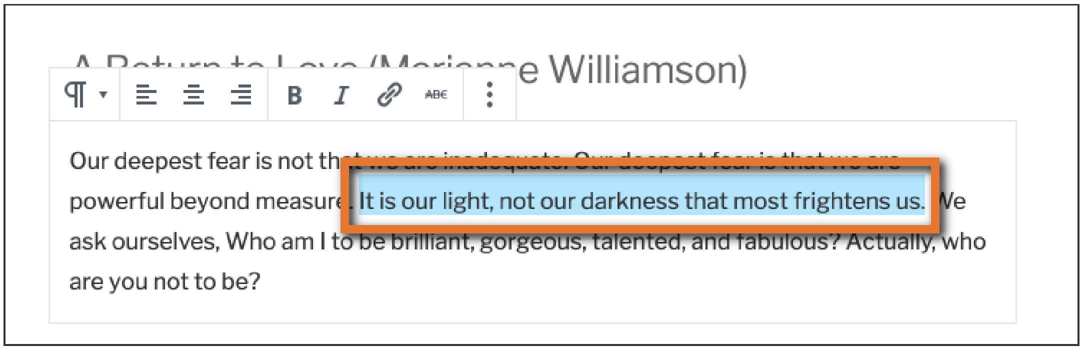
GUT-97 - Blocks use custom selection color

- **Severity:** Low
- **Affected Populations:** Low-Vision
- **Platform(s):** All/ Universal
- **Components affected:** Block Editing, Block Options

Issue description

Blocks use a custom CSS color for text selection.

For Firefox users who have low-vision and use a Windows high-contrast theme, custom selection colors are not honoured, but are not replaced by defaults either. Consequently, these users will not be able to see what text is selected.



A Return to Love (Marianne Williamson)

Our deepest fear is not that we are inadequate. Our deepest fear is that we are powerful beyond measure. It is our light, not our darkness that most frightens us. We ask ourselves, Who am I to be brilliant, gorgeous, talented, and fabulous? Actually, who are you not to be?

Issue Code

```
.editor-block-list__layout .editor-block-list__block ::selection {
    background-color: #b3e7fe;
}
```

Remediation Guidance

Remove custom selection rules, and allow the browser/OS native color to be rendered.

Relevant standards

- 1.3.3 Sensory Characteristics (Level A)

GUT-89 - Insufficient color contrast on block type buttons

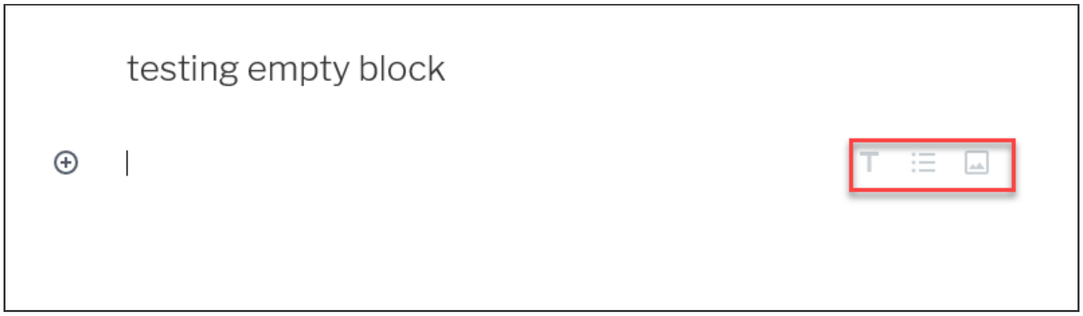
- **Severity:** Medium
- **Affected Populations:** Low-Vision, Cognitively Impaired
- **Platform(s):** All/ Universal
- **Components affected:** Empty Block

Issue description

Three Block type buttons are implemented with very light gray SVG elements, the contrast ratio of which is below the minimum threshold of 3:1 for interface components:

- #c9d0d5 (ash grey) on #fff (white): **1.6:1**

Sufficient color contrast is important for users who have low-vision or are color-blind, because text with a low contrast ratio may be difficult or impossible for such users to see.



testing empty block

Issue Code

```
.editor-inserter-with-shortcuts__block {
    ...
    color: rgba(102,120,134,.35);
}
```

Remediation Guidance

The SVG elements inherit their `fill` color from parent `<span>` elements, via `currentColor`.

Therefore the parent color needs to be darkened to meet the 3:1 minimum contrast ratio. The button border remaining light grey before user interaction, combined with the even darker hover/focus color plus the visible tooltip, will ensure that the difference between default and interacted elements is still clearly visible.

Recommended Code

```
.editor-inserter-with-shortcuts__block {
    ...
    color: rgba(102,120,134,.8);
}
```

Relevant standards

- 1.4.3 Contrast (Minimum) (Level AA)

GUT-76 - Field boundaries have insufficient color contrast

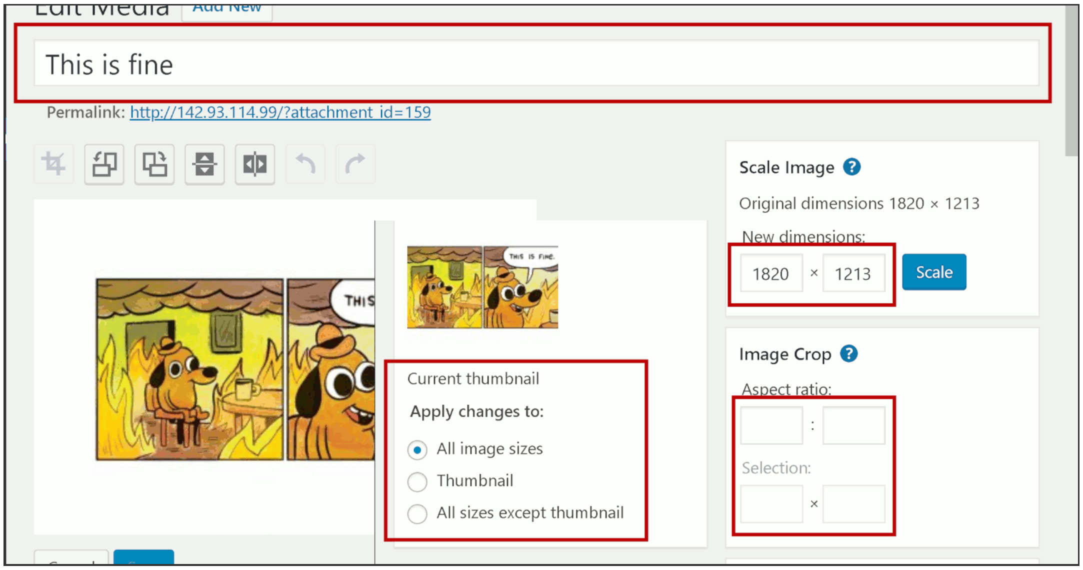
- **Severity:** Low
- **Affected Populations:** Low-Vision, Cognitively Impaired
- **Platform(s):** All/ Universal
- **Components affected:** Edit Media

Issue description

On the Edit Media page, several interactive components have insufficient color contrast for their borders or backgrounds, below the minimum 3:1 for interface components:

- Text input borders: #ddd (light-grey) against #fff (white): **1.35**.
- Checkbox and radio borders: #b4b9be (light grey) against #fff (white): **1.97**.
- Focus shadow for fields and buttons: #5b9dd9 (light blue) on #fff (white): **2.88**.

Sufficient color contrast is important for users who have low-vision or are color-blind, because content with a low contrast ratio may be difficult or impossible for such users to see.



This is fine

Permalink: http://142.93.114.99/?attachment_id=159

Scale Image

Original dimensions 1820 × 1213

New dimensions:

1820 × 1213

Image Crop

Aspect ratio:

:

Selection:

×

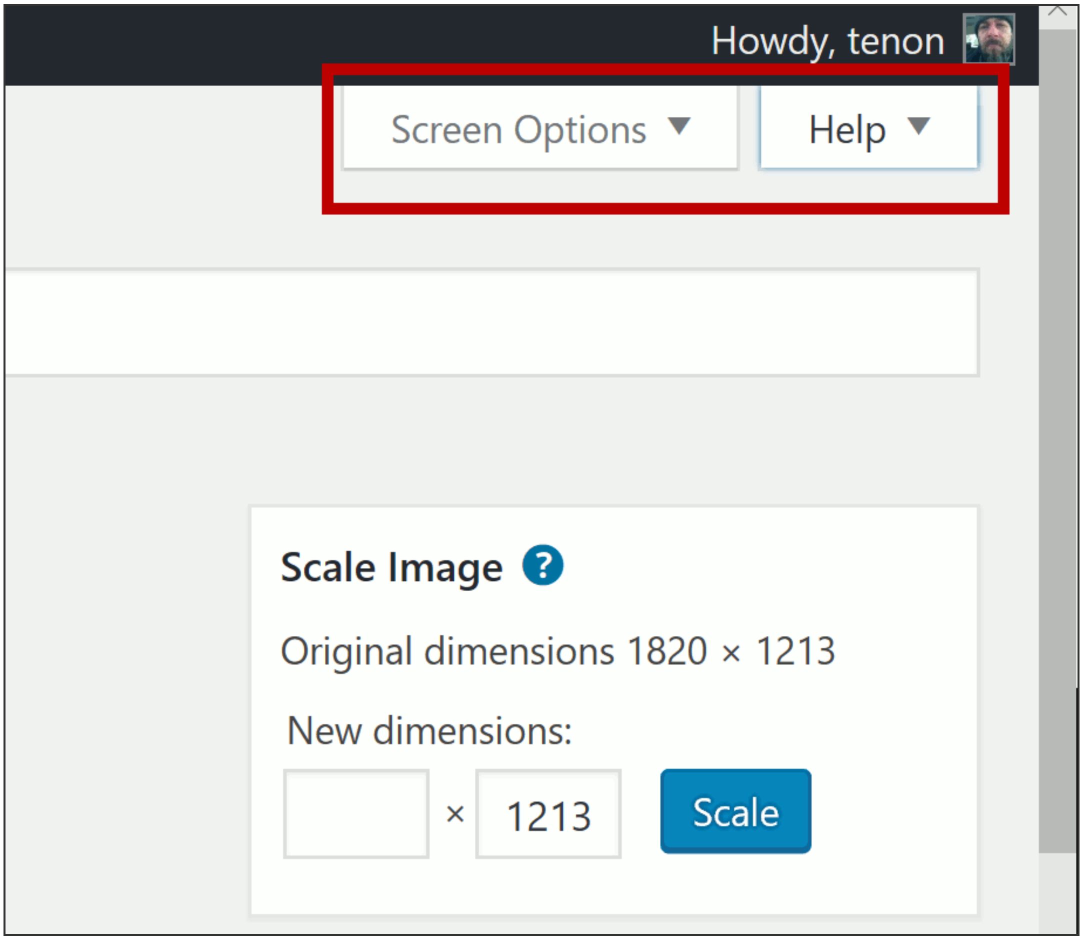
Current thumbnail

Apply changes to:

● All image sizes

○ Thumbnail

○ All sizes except thumbnail



Scale Image ?

Original dimensions 1820 × 1213

New dimensions:

□ × 1213 Scale

Issue Code

```
input[type=checkbox], input[type=color], input[type=date], input[type=datetime-local], input[type=datetime],
input[type=email], input[type=month], input[type=number], input[type=password], input[type=radio],
input[type=search], input[type=tel], input[type=text], input[type=time], input[type=url], input[type=week],
select, textarea {

    border: 1px solid #ddd;

    box-shadow: inset 0 1px 2px rgba(0,0,0,.07);

    background-color: #fff;

    ...

}


/* radio buttons, checkmarks */

input[type=checkbox], input[type=radio] {

    border: 1px solid #b4b9be;

    ...

}


/* focus state */

input[type=checkbox]:focus, input[type=color]:focus, input[type=date]:focus, input[type=datetime-local]:focus,
input[type=datetime]:focus, input[type=email]:focus, input[type=month]:focus, input[type=number]:focus,
input[type=password]:focus, input[type=radio]:focus, input[type=search]:focus, input[type=tel]:focus,
input[type=text]:focus, input[type=time]:focus, input[type=url]:focus, input[type=week]:focus, select:focus,
textarea:focus {

    border-color: #5b9dd9;

    box-shadow: 0 0 2px rgba(30,140,190,.8);

}


/* buttons at top */

.wp-core-ui .button-secondary:focus, .wp-core-ui .button.focus, .wp-core-ui .button:focus {

    box-shadow: 0 0 3px rgba(0,115,170,.8);

}

#screen-meta-links .show-settings {

    border: 0;

    ...

}
```

Remediation Guidance

Darken the gray borders and the focus shadow color to provide a minimum contrast ratio of 3:1 relative to the white background.

Since `box-shadow` is not shown in High Contrast themes, this should also be supplemented with a transparent `outline` (or replaced with a visible outline).

Recommended Code

```
input[type=checkbox], input[type=color], input[type=date], input[type=datetime-local], input[type=datetime],
input[type=email], input[type=month], input[type=number], input[type=password], input[type=radio],
input[type=search], input[type=tel], input[type=text], input[type=time], input[type=url], input[type=week],
select, textarea {

    border: 1px solid #949494;

    box-shadow: inset 0 1px 2px rgba(0,0,0,.07);

    background-color: #fff;

    ...

}


/* radio buttons, checkmarks */

input[type=checkbox], input[type=radio] {

    border: 1px solid #949494;

    ...

}


/* focus state */

input[type=checkbox]:focus, input[type=color]:focus, input[type=date]:focus, input[type=datetime-local]:focus,
input[type=datetime]:focus, input[type=email]:focus, input[type=month]:focus, input[type=number]:focus,
input[type=password]:focus, input[type=radio]:focus, input[type=search]:focus, input[type=tel]:focus,
input[type=text]:focus, input[type=time]:focus, input[type=url]:focus, input[type=week]:focus, select:focus,
textarea:focus {

    border-color: #5999d4;

    box-shadow: 0 0 2px rgba(30,140,190,.8);

    outline: 1px solid transparent;

}


/* buttons at top */

#screen-meta-links .show-settings:focus {

    border: 1px solid #0073aa;

    ...

}
```

Relevant standards

- 1.4.11 Non-text Contrast (Level AA)

GUT-75 - Button has insufficient color contrast

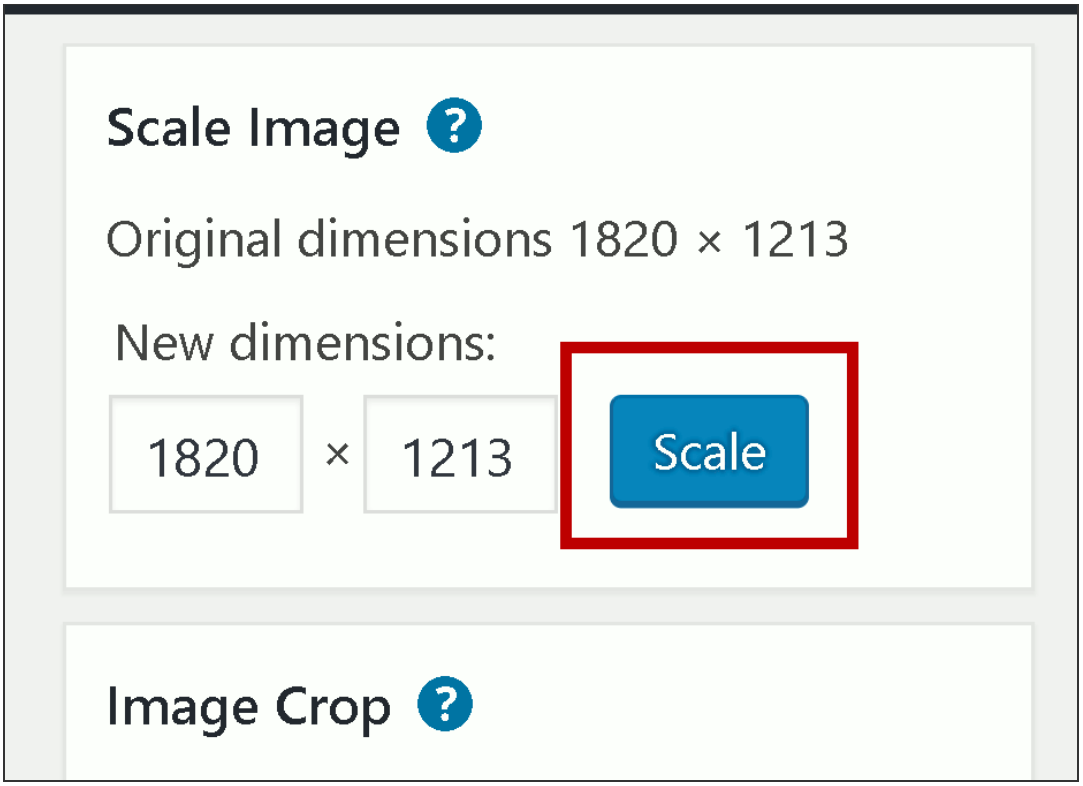
- **Severity:** Low
- **Affected Populations:** Low-Vision
- **Platform(s):** All/ Universal
- **Components affected:** Edit Media

Issue description

On the Edit Media page, the "Scale" button has insufficient color contrast, below the minimum threshold of 4.5:1 for text:

- Scale button: #fff (white) on #0085ba (light blue): **4.14**.

Sufficient color contrast is important for users who have low-vision or are color-blind, because text with a low contrast ratio may be difficult or impossible for such users to see.



Scale Image ?

Original dimensions 1820 × 1213

New dimensions:

1820 × 1213

Image Crop ?

Issue Code

```
.wp-core-ui .button-primary {
    ...
    background: #0085ba;
    color: #fff;
    ...
}
```

Remediation Guidance

Darken the blue background to match the other blues on the page, which have sufficient contrast relative to their white text.

Recommended Code

```
.wp-core-ui .button-primary {
    ...
    background: #0074a2;
    color: #fff;
    ...
}
```

Relevant standards

- 1.4.3 Contrast (Minimum) (Level AA)

GUT-55 - Insufficient color contrast in the options menu

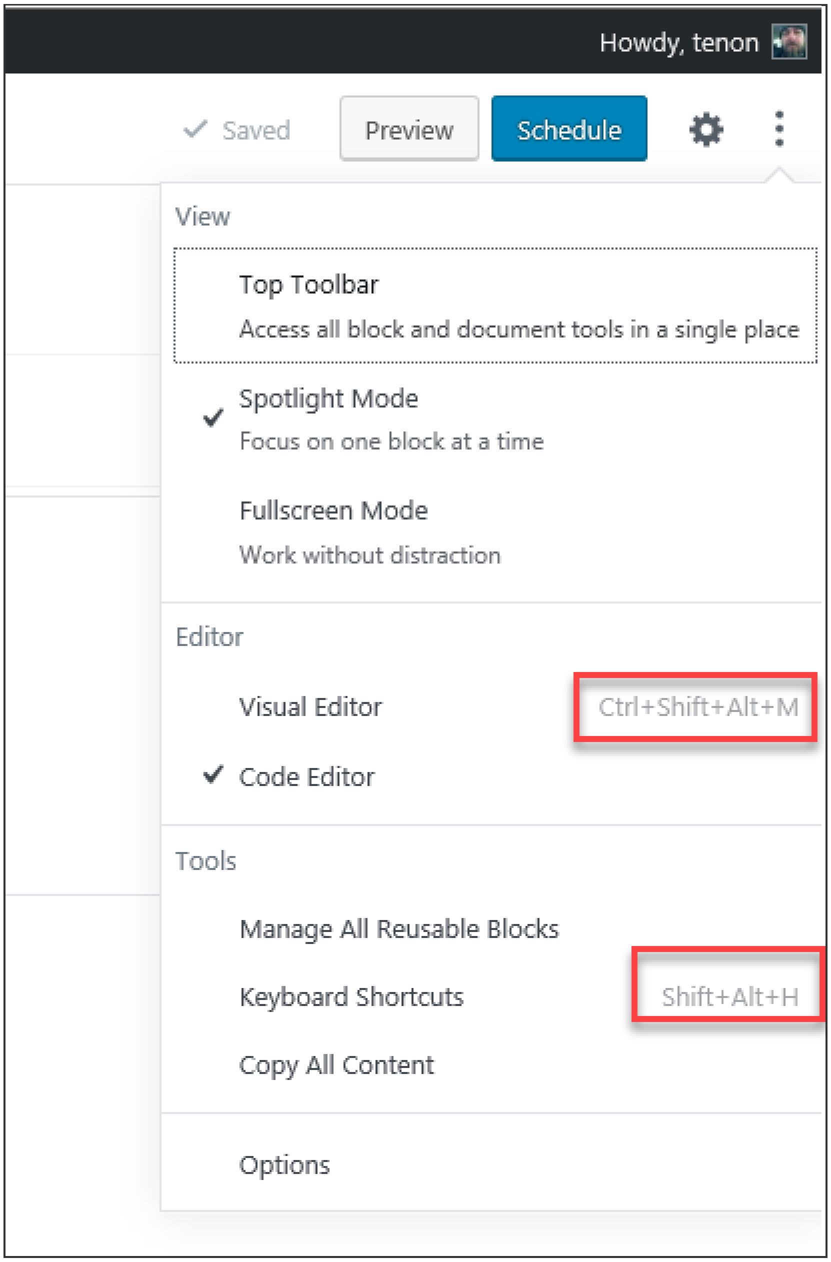
- **Severity:** Medium
- **Affected Populations:** Low-Vision, Cognitively Impaired
- **Platform(s):** All/ Universal
- **Components affected:** Tools and Options

Issue description

Shortcut key instructions in the "Visual Editor" and "Keyboard Shortcuts" items of the Options menu have insufficient color contrast, below the minimum threshold of 4.5:1 for text:

- "Ctrl+Shift+Alt+M" and "Shift+Alt+H" rendered text: #a0a3a6 (light gray) on #fff (white): **2.5:1**

Sufficient color contrast is important for users who have low-vision or are color-blind, because text with a low contrast ratio may be difficult or impossible for such users to see.


Howdy, tenon
Saved
Preview
Schedule
View
Top Toolbar
Access all block and document tools in a single place
✓ Spotlight Mode
Focus on one block at a time
Fullscreen Mode
Work without distraction
Editor
Visual Editor
Ctrl+Shift+Alt+M
✓ Code Editor
Tools
Manage All Reusable Blocks
Keyboard Shortcuts
Shift+Alt+H
Copy All Content
Options

Issue Code

```
/* media all */
.components-menu-item__shortcut {
    ...
    opacity: 0.5;
    ...
}
```

```
/* media all */
:not(:disabled):not([aria-disabled=true]).components-menu-item__button.components-icon-button:focus,
:not(:disabled):not([aria-disabled=true]).components-menu-item__button:focus {
    color: #191e23;
    ...
}
```

Remediation Guidance

Increase the opacity, or remove the opacity and use a mid-gray color, so that the text has a minimum contrast ratio of 4.5:1.

Recommended Code

```
/* media all */
.components-menu-item__shortcut {
    ...
    opacity: 1;
    ...
}
```

```
/* media all */
:not(:disabled):not([aria-disabled=true]).components-menu-item__button.components-icon-button:focus,
:not(:disabled):not([aria-disabled=true]).components-menu-item__button:focus {
    color: #767676;
    ...
}
```

Relevant standards

- 1.4.3 Contrast (Minimum) (Level AA)

GUT-44 - Insufficient contrast on empty field borders

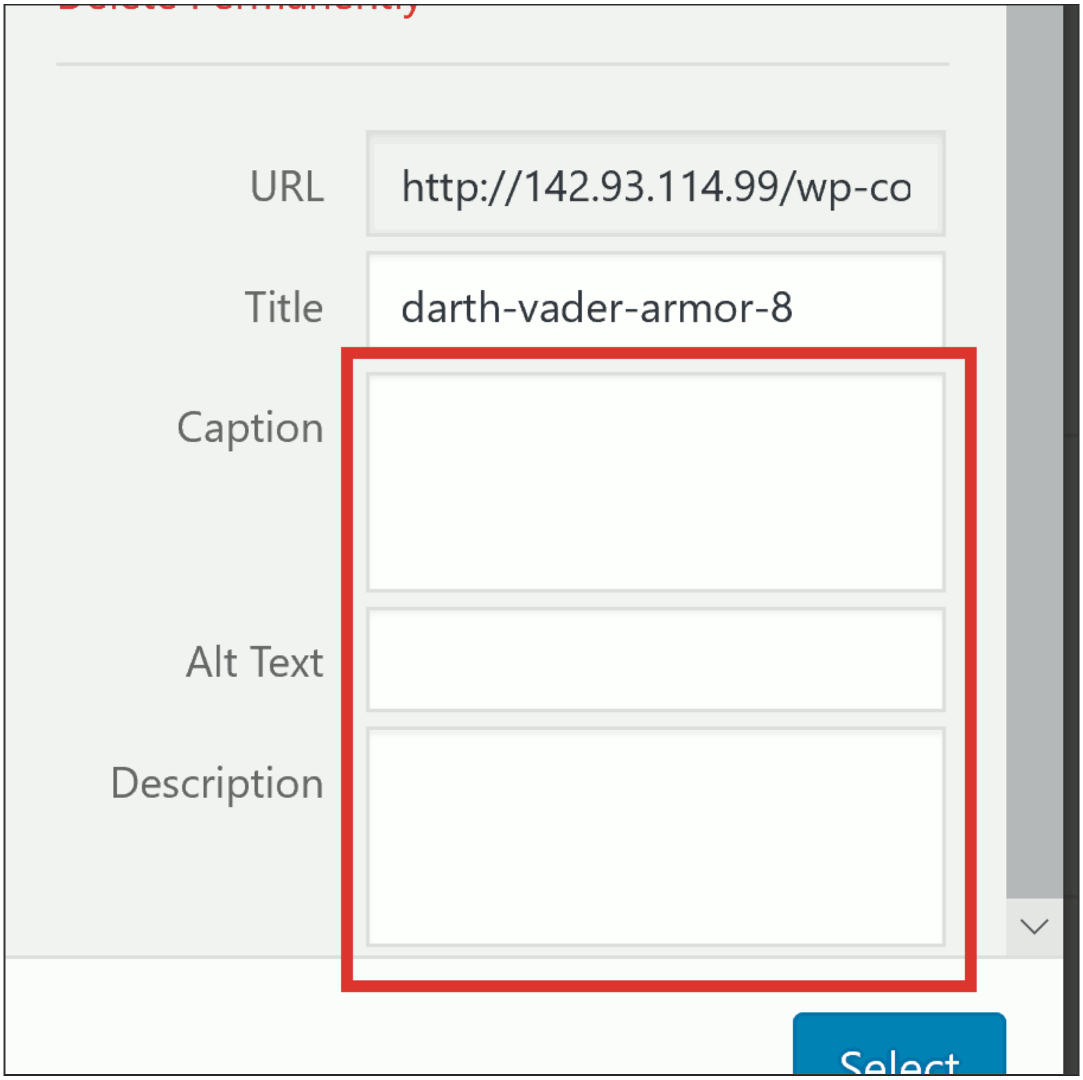
- **Severity:** Low
- **Affected Populations:** Low-Vision, Cognitively Impaired
- **Platform(s):** All/ Universal
- **Components affected:** Media Dialog

Issue description

Some interface components in the "Featured Image" modal have insufficient color contrast for their borders when empty, below the minimum threshold of 3:1 for interface components:

- Caption, Alt Text and Description field borders: light-grey (#ddd) on lighter grey (#f3f3f3): **1.35:1**

Sufficient color contrast is important for users who have low-vision or are color-blind, because content with a low contrast ratio may be difficult or impossible for such users to see.



URL http://142.93.114.99/wp-co

Title darth-vader-armor-8

Caption

Alt Text

Description

Issue Code

```
.media-frame input[type=email], .media-frame input[type=number], .media-frame input[type=password], .media-frame
input[type=search], .media-frame input[type=text], .media-frame input[type=url], .media-frame select, .media-frame
textarea {

    ...

    border-width: 1px;

    border-style: solid;

    border-color: #ddd;

}
```

Remediation Guidance

Darken the `border-color` to create a minimum contrast ratio of 3:1.

Recommended Code

```
.media-frame input[type=email], .media-frame input[type=number], .media-frame input[type=password], .media-frame
input[type=search], .media-frame input[type=text], .media-frame input[type=url], .media-frame select, .media-frame
textarea {

    ...

    border-width: 1px;

    border-style: solid;

    border-color: #8f8f8f;

}
```

Relevant standards

- 1.4.11 Non-text Contrast (Level AA)

GUT-43 - Insufficient contrast on link text

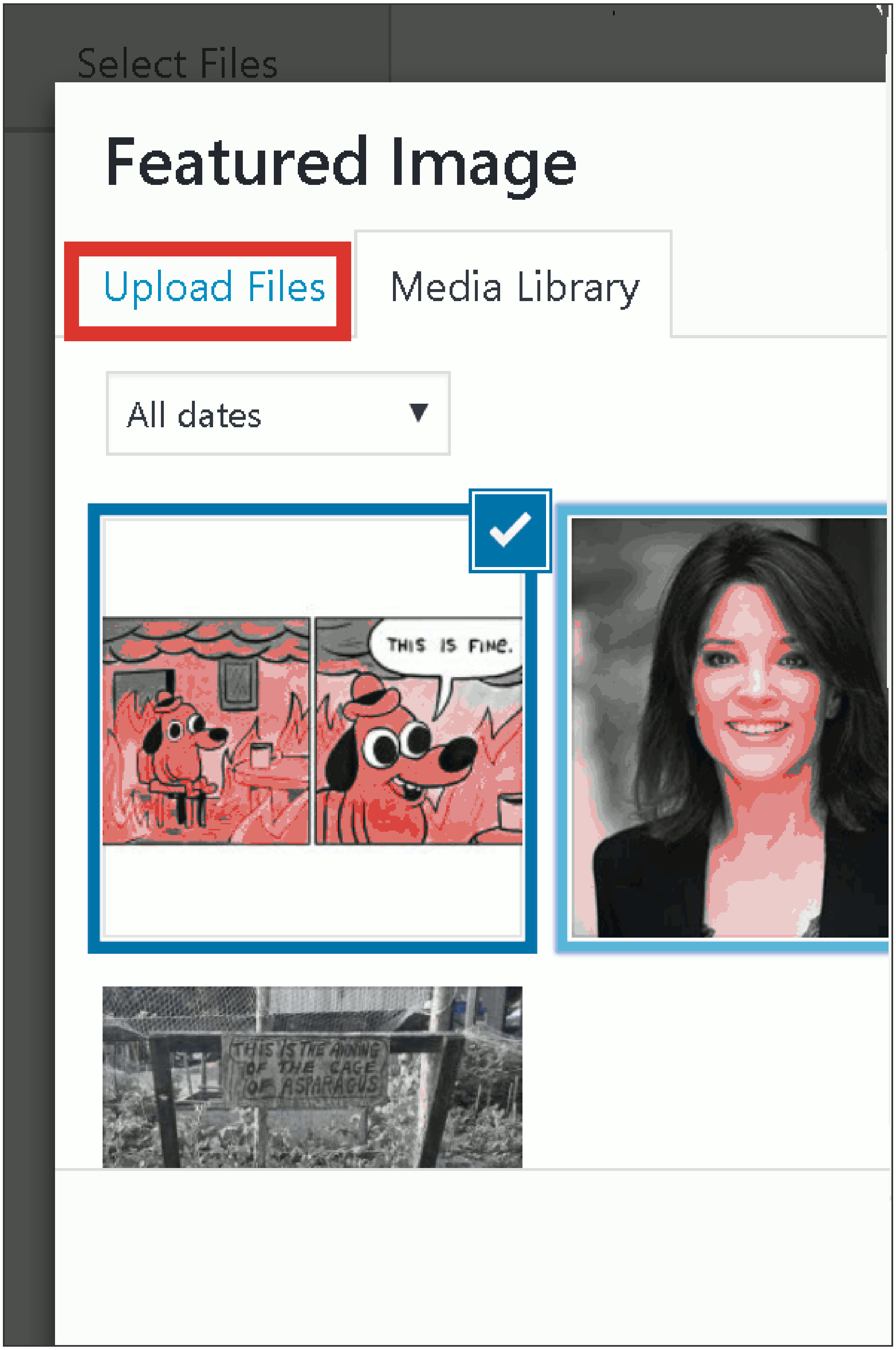
- **Severity:** Medium
- **Affected Populations:** Low-Vision, Cognitively Impaired
- **Platform(s):** All/ Universal
- **Components affected:** Block Types, Media Dialog

Issue description

When the "Featured Image" modal is open, there are two controls coded as links which have insufficient color contrast, below the minimum threshold of 4.5:1 for text:

- "Upload Files" and "Media Library" blue hover color: blue (#00a0d2) on white (#fff): **3.01:1**

Sufficient color contrast is important for users who have low-vision or are color-blind, because text with a low contrast ratio may be difficult or impossible for such users to see.


Select Files
Featured Image
Upload Files
Media Library
All dates ▼
THIS IS FINE.
THIS IS THE ANNING OF THE CAGE OF ASPARAGUS

Issue Code

```
.media-frame a:active, .media-frame a:hover {
    color: #00a0d2;
}
```

Remediation Guidance

Darken the hover color to match the focus color, or otherwise to provide a contrast ratio of 4.5:1 or greater.

Recommended Code

```
.media-frame a:active, .media-frame a:hover {
    color: #124964;
}
```

Relevant standards

- 1.4.3 Contrast (Minimum) (Level AA)

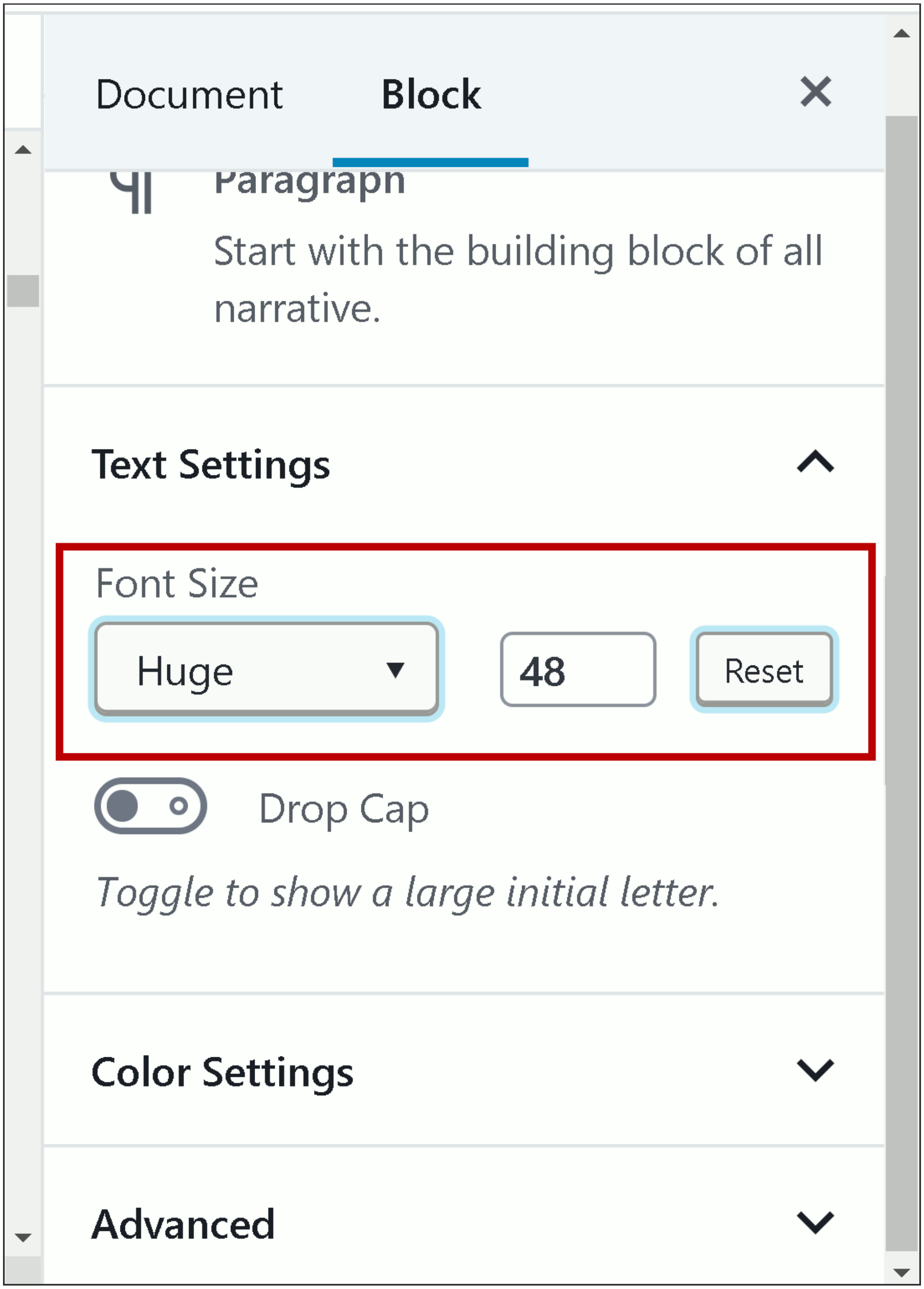
GUT-31 - Focus state has low contrast

- **Severity:** Medium
- **Affected Populations:** Low-Vision, Motor Impaired, Cognitively Impaired
- **Platform(s):** All/ Universal
- **Components affected:** Block Panel

Issue description

Some of the controls in the Block panel have very low contrast in their focussed state.

All interactive elements should have a visible and obvious focus indicator, so that sighted keyboard users have a clear indication of which element has focus. The visual presentation of user interface component states must have a contrast ratio of at least 3:1 against adjacent color(s).



Start with the building block of all narrative.

Font Size: Huge, 48

Drop Cap

Toggle to show a large initial letter.

Issue Code

```
.components-button.is-default:focus:enabled {
    background: #fafafa;
    color: #23282d;
    border-color: #999;
    box-shadow: inset 0 -1px 0 #999, 0 0 0 2px #bfe7f3;
    ...
}
```

Remediation Guidance

Darken the `box-shadow` color on focus.

Recommended Code

```
.components-button.is-default:focus:enabled {
    background: #fafafa;
    color: #23282d;
    border-color: #999;
    box-shadow: 0 0 0 1px #00a0d2;
    ...
}
```

Relevant standards

- 1.4.11 Non-text Contrast (Level AA)
- 2.4.7 Focus Visible (Level AA)

GUT-19 - Insufficient color contrast for calendar selected date and Publish button

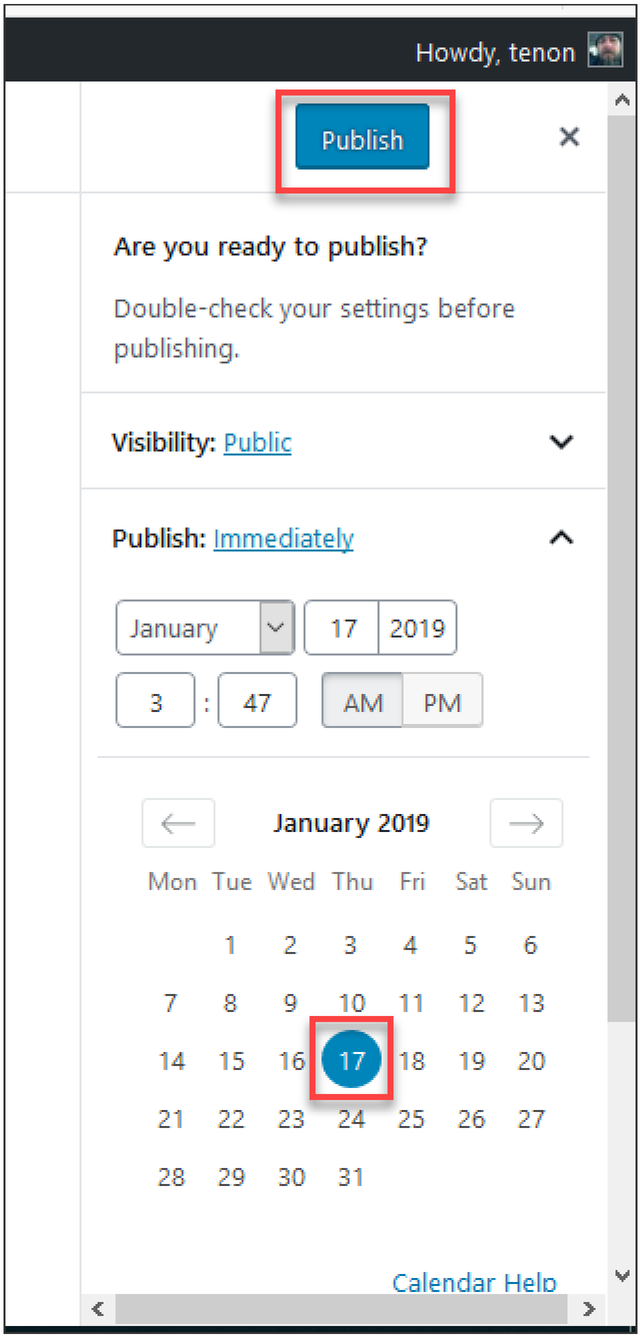
- **Severity:** Medium
- **Affected Populations:** Low-Vision, Cognitively Impaired
- **Platform(s):** All/ Universal
- **Components affected:** Document Panel, Publish and Unpublish

Issue description

The selected date within the calendar, and the "Publish" button, have insufficient color contrast, below the minimum threshold of 4.5:1 for text:

- "Publish" button: #fff (white) on #0085ba (blue): **4.15:1**
- Selected date within the calendar: #fff (white) on #0085ba (blue): **4.15:1**

Sufficient color contrast is important for users who have low-vision or are color-blind, because text with a low contrast ratio may be difficult or impossible for such users to see.



Are you ready to publish?

Double-check your settings before publishing.

Visibility: Public

Publish: Immediately

January 17 2019

3 : 47 AM PM

Mon	Tue	Wed	Thu	Fri	Sat	Sun
	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	31			

Issue Code

```
/*   l     d   l d   d     */
/* selected calendar date */
.components-datetime__date .CalendarDay__selected {
    background: #0085ba;
}
.CalendarDay__selected,
.CalendarDay__selected:active,
.CalendarDay__selected:hover {
    ...
    color: #fff;
}

/* Publish button */
.components-button.is-primary {
    background: #0085ba;
     ...
    color: #fff;
}
```

Remediation Guidance

Darken the background color to achieve a minimum contrast of 4.5:1.

Recommended Code

```
/* selected calendar date */
.components-datetime__date .CalendarDay__selected {
    background: #006a95;
}
.CalendarDay__selected,
.CalendarDay__selected:active,
.CalendarDay__selected:hover {
    ...
    color: #fff;
}

/* Publish button */
.components-button.is-primary {
    background: #006a95;
     ...
    color: #fff;
}
```

Relevant standards

- 1.4.3 Contrast (Minimum) (Level AA)

Related Issues

- Related to "Insufficient color contrast on Editor Top Bar"

GUT-7 - Focus not visible on Preview button, and low contrast on various buttons

- **Severity:** High
- **Affected Populations:** Low-Vision, Motor Impaired, Cognitively Impaired
- **Platform(s):** All/ Universal
- **Components affected:** Editor Top Bar

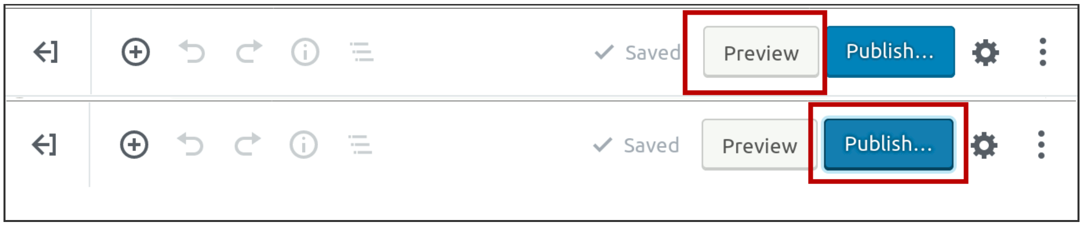
Issue description

In the Editor Top Bar, the "Preview" button's focus state is not visible. The focus state of the "Publish" button is visible but very low contrast; other buttons (such as the default buttons in the Image Block) use the same style:

- #bfe7f3 (light blue) against #fff (white): **1.31:1**.

Since the "Preview" button's focus styles aren't shown in most browsers, this also means that the transparent outline meant for Windows High Contrast users does not appear.

All interactive elements should have a visible and obvious focus indicator, so that sighted keyboard users have a clear indication of which element has focus. Sufficient color contrast showing states is important for users who have low-vision or are color-blind, because states with a low contrast ratio may be difficult or impossible for such users to see.



Issue Code

```
/* Publish */
.components-button.is-primary:focus:enabled {
    box-shadow: inset 0 -1px 0 #00435d,0 0 0 2px #bfe7f3;
}
...
/* Preview */
.components-button:focus:enabled {
    box-shadow: inset 0 0 0 1px #6c7781, inset 0 0 0 2px #fff;
    outline: 2px solid transparent;
    outline-offset: -2px;
}
```

Remediation Guidance

The combination of `:focus` with `:enabled` seems unsupported for elements which do not have a `disabled` state in most browsers. Use another method to style focus states until browser support has improved.

Disabled elements shouldn't be able to receive `:focus` states in any case. Anchors used as buttons may be better off using `tabindex="-1"` in the HTML so that code does not have to specify `:enabled`.

Use a darker blue outline for focus indication to supplement the `box-shadow`.

Recommended Code

```
/* Publish */
.components-button.is-primary:focus {
    outline: 2px solid #00435d;
    box-shadow: inset 0 -1px 0 #00435d,0 0 0 2px #bfe7f3;
}
...
/* Preview */
.components-button:focus {
    box-shadow: inset 0 0 0 1px #6c7781, inset 0 0 0 2px #fff;
    outline: 2px solid transparent;
    outline-offset: -2px;
}
```

Relevant standards

- 1.4.1 Use of Color (Level A)
- 1.4.11 Non-text Contrast (Level AA)
- 2.4.7 Focus Visible (Level AA)

Related Issues

- Related to "Buttons are low contrast and focus not visible"

GUT-1 - Insufficient color contrast on Editor Top Bar

- **Severity:** Medium
- **Affected Populations:** Low-Vision, Cognitively Impaired
- **Platform(s):** All/ Universal
- **Components affected:** Editor Top Bar

Issue description

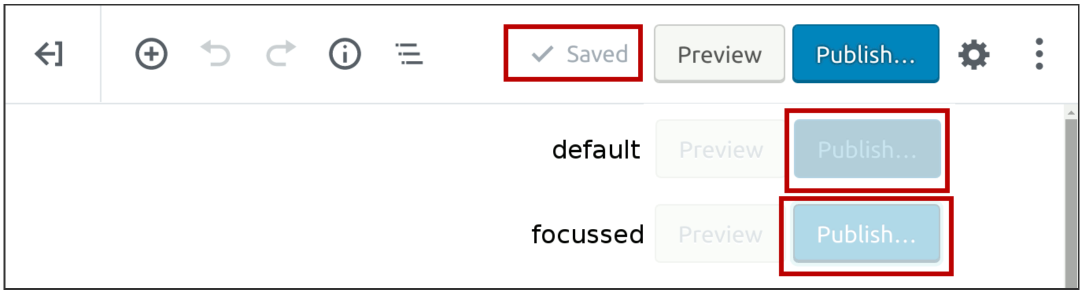
Some controls within the Editor Top Bar have contrast ratios below the 4.5:1 minimum:

- "Saved" text and its icon: #a2aab2 (light grey) on #fff (white): **2.35:1**

The "Publish" button remains focusable and clickable but with low contrast:

- When unfocussed: #4daacf (light blue) on #005d82 (darker blue): **2.75:1**
- When focussed: #fff (white) on #b2d8e7 (this is dark blue #007eb1 but with .3 of the normal opacity): **1.51:1**
- After being made visibly enabled: #fff (white) on #0085ba (blue): **4.15:1**

Sufficient color contrast is important for users who have low-vision or are color-blind, because text with a low contrast ratio may be difficult or impossible for such users to see.



Issue Code

```
/* "saved" text and icon on a #fff background */
.editor-post-saved-state {
    ...
    color: #a2aab2;
}

/* "Publish" unfocussed: */
.components-button.is-primary:disabled,
.components-button.is-primary[aria-disabled=true] {
    ...
    color: #4daacf;
    background: #005d82;
    text-shadow: 0 -1px 0 rgba(0,0,0,.1);
}
/* "Publish" focussed: */
.components-button.is-primary:focus:enabled,
.components-button.is-primary:hover {
    background: #007eb1;
    ...
    color: #fff;
}
.components-button:disabled,
.components-button[aria-disabled=true] {
    ...
    opacity: .3;
}
```

Remediation Guidance

Darken the "Saved" text until it meets the 4.5:1 minimum ratio.

Do not allow focus to reach the "Publish" button when it's attempting to display a disabled state. Either increase the contrast at all times *or* make the button truly disabled until the "Publish" button can be activated.

Other disabled controls use disabled colors and are focusable, however their visible tooltips have sufficient contrast, allowing a user with low-vision to read what the disabled control's name and keyboard shortcut is.

Recommended Code

```
/* "saved" text and icon */
.editor-post-saved-state {
    ...
    color: #767676;
}

/* Publish button */
.components-button.is-primary {
    background: #006a95;
    ...
    color: #fff;
}
```

Relevant standards

- 1.4.3 Contrast (Minimum) (Level AA)

Related Issues

- Related to "Insufficient color contrast for calendar selected date and Publish button"

Content Type: Document Structure

GUT-113 - Users cannot add video metadata in the visual editor

- **Severity:** Medium
- **Affected Populations:** Blind, Low-Vision, Cognitively Impaired
- **Platform(s):** All/ Universal
- **Components affected:** Block Types, Output Markup

Issue description

Authors cannot add captions (such as .srt or .vtt files) to video media using the visual editor.

Users who do not have HTML knowledge may not know how to do these things using HTML, resulting in them publishing content which is less accessible to readers.

Remediation Guidance

The ability to add captions should be part of the Video block's visual editor, similar to the way that image media provides the option to add a caption.

Relevant standards

- 1.2.2 Captions (Prerecorded) (Level A)

Related Issues

- Related to "Users cannot add table metadata in the visual editor"

GUT-112 - Users can duplicate post titles

- **Severity:** Medium
- **Affected Populations:** Blind, Low-Vision, Cognitively Impaired
- **Platform(s):** All/ Universal
- **Components affected:** Output Markup

Issue description

It's possible for authors to create multiple posts with the same page title.

Assistive technology users may rely on the title to understand the primary purpose of each page. When titles are not unique, it may be more difficult and laborious for such users to determine what the page is about.

Remediation Guidance

Having something warn authors that another post has already been published (not drafted) with the same title would help authors ensure post titles are unique.

This need not literally prevent users from creating multiple posts with the same title, merely warn them why this is not a good idea.

Relevant standards

- 2.4.2 Page Titled (Level A)

GUT-103 - Users cannot add table metadata in the visual editor

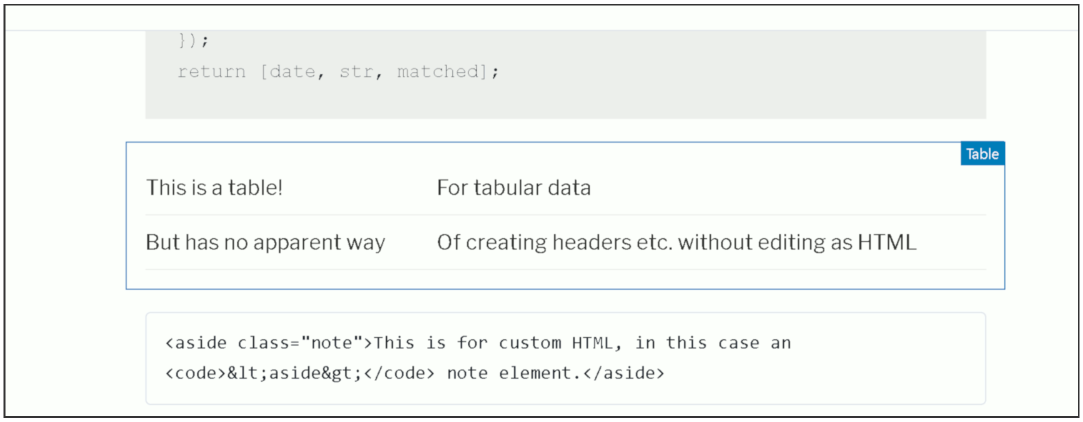
- **Severity:** Medium
- **Affected Populations:** Blind, Low-Vision, Cognitively Impaired
- **Platform(s):** All/ Universal
- **Components affected:** Block Editing, Output Markup

Issue description

Authors using the visual editor can create a table, however the table which results is only a series of rows and columns with no column-header information, which may cause assistive technologies to assume these are layout tables rather than data tables.

Users appear to be only able to add table headers by switching to the HTML view. However not all authors have HTML knowledge and so may be producing inaccessible tables.

Authors attempting to add table semantics such as a `<caption>` element are shown an error, which may cause people using the editor to think these are real errors.



```
});
return [date, str, matched];
```

This is a table!	For tabular data
But has no apparent way	Of creating headers etc. without editing as HTML

```
<aside class="note">This is for custom HTML, in this case an
<code>&lt;aside&gt;</code> note element.</aside>
```

Issue Code

```
<!-- default resultant code in HTML view -->
<table class="wp-block-table is-style-regular">
    <tbody>
        <tr>
            <td>This is a table!</td>
            <td>For tabular data</td>
        </tr>
        <tr>
            <td>But has no apparent way</td>
            <td>Of creating headers etc. without editing as HTML</td>
        </tr>
    </tbody>
</table>
```

Remediation Guidance

Make the default Table block type assume that the first row is a header row. This assumption is similar to what CSV files do, and will help content authors create semantic tables despite any lack of HTML knowledge.

Allow users to use the HTML editor to add in other semantics, such as `<caption>` elements or `scope` attributes, without showing these as errors to users.

Recommended Code

```
<table class="wp-block-table is-style-regular">
    <thead>
        <tr>
            <th>This is a table!</th>
            <th>For tabular data</th>
        </tr>
    </thead>
    <tbody>
        <tr>
            <td>But has no apparent way</td>
            <td>Of creating headers etc. without editing as HTML</td>
        </tr>
    </tbody>
</table>
```

Related Issues

- Related to "Users cannot add video metadata in the visual editor"

GUT-99 - Keyboard focus order doesn’t match visual order at high browser zoom

- **Severity:** Medium
- **Affected Populations:** Low-Vision, Motor Impaired, Cognitively Impaired
- **Platform(s):** All/ Universal
- **Components affected:** Block Editing, Block Options

Issue description

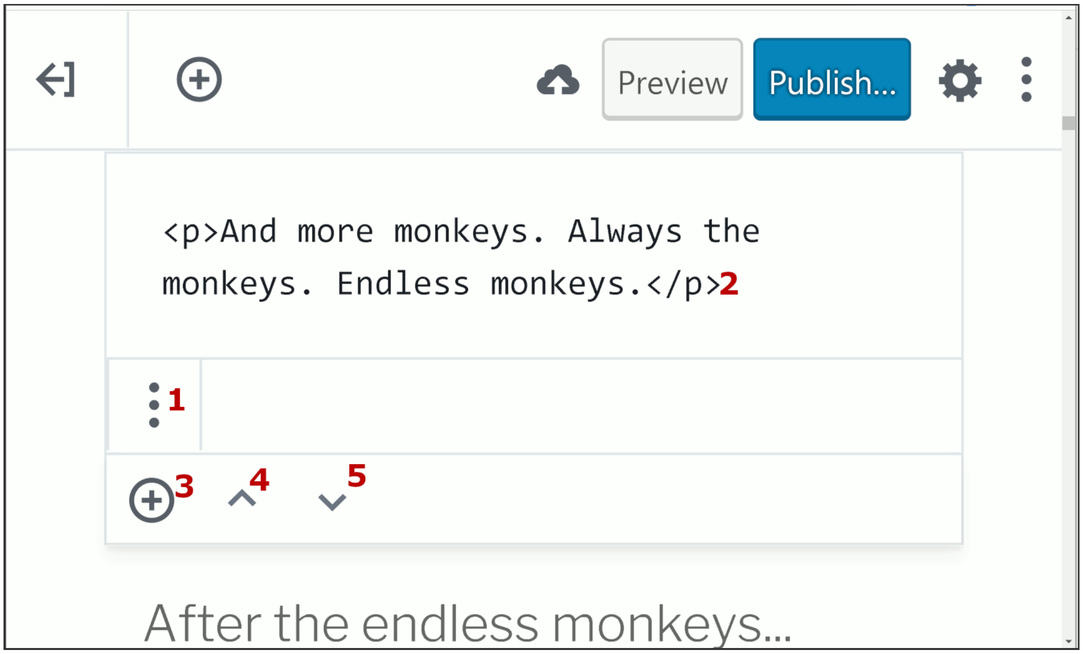
When users are zoomed-in up to 400%, content is re-arranged such that the focus order differs from the visual order, and focus is able to move to items which are covered by other content.

Specifically:

- Users who choose to open the Block panel from the Options menu do not have their focus moved there, however at higher zoom levels the Block panel is styled to completely cover the viewport like a modal dialog.
- Users who, after making changes in the Block panel and following the skip link back to the selected block, have their focus moved back to the block in question, but again the Block panel remains open and covers the block. Users who instead close the Block panel with its "Close" button then have no means to quickly jump back to the Block they're editing.
- The editing toolbar for a block (when there is no Unified Toolbar) is positioned below the block's text. Focus moving forward goes from the invisible "Add Block" button, to the toolbar, to the block's text or main area, then finally the second "Add Block" and move buttons. The focus jumps back and forth between items whose visual order is different, and this is exacerbated when blocks are very tall, because it also causes the screen to jump around.

Keyboard users benefit from a logical focus order which is consistent with the reading order. Users who have a cognitive disability and have difficulty reading may also be confused by an unexpected or illogical focus order.

Obscuring the focus position is potentially confusing for sighted keyboard users, since it may appear as though no item has focus, or make it unclear as to where the focus position is.



```
<p>And more monkeys. Always the
monkeys. Endless monkeys.</p>
```

After the endless monkeys...

Issue Code

```
di id "   " l   "   " d        "   /      h"  hi d  " "  i l h l " l k        h"
```

```
<div id="..." class="..." data-type="core/paragraph" tabindex="0" aria-label="Block: Paragraph">
    <div class="...">
        <div tabindex="-1" class="...">
            <div class="...">
                <button type="button" aria-label="Add block" ... class="...">...</button>
            </div>
        </div>
    </div>
    <div class="..."></div>
    <div class="...">
        <button type="button" aria-label="Move up" ...>...</button>
        <div ...aria-hidden="true" draggable="true"><svg...>...</svg></div>
        <button type="button" aria-label="Move down" ...>...</button>
        <span id="..." class="...">...</span>
    </div>
    <div class="editor-block-list__block-edit">
        <div role="toolbar" aria-orientation="horizontal" class="..." aria-label="Block tools">
            <div class="editor-block-toolbar">...</div>
        </div>
        ...
        <div class="components-autocomplete">
            <p aria-label="Paragraph block" aria-autocomplete="list" role="textbox" aria-multiline="true"
class="..." contenteditable="true"...> What you are reading now is a <strong>text block</strong> the most basic
block of all. The text block has its own controls to be moved freely around the post... </p>
        </div>
        ...
    </div>
</div>
```

Remediation Guidance

It would be most ideal if the focus order matched the source order, however since controls are not visibly available until after focus has moved to a block, the visual order should match the focus order when controls are visible.

Additionally, a strong focus style when the block content has focus (such as the dark blue outline shown on mouseover) would assist users in clearly seeing that it is the block content with focus; currently only the user's cursor (and sometimes the page jumping) shows the user that focus has moved to the block content.

Relevant standards

- 1.4.10 Reflow (Level AA)
- 2.1.1 Keyboard (Level A)
- 2.4.3 Focus Order (Level A)

GUT-83 - Page content is an invisible Tab stop

- **Severity:** Medium
- **Affected Populations:** Blind, Low-Vision, Motor Impaired, Cognitively Impaired
- **Platform(s):** All/ Universal
- **Components affected:** Edit Media

Issue description

On the Edit Media page, the main content element is an invisible Tab stop, because it's in the Tab order but does not show focus.

This is potentially confusing for sighted keyboard users, since it will make it appear as though no item has focus, or make it unclear as to where the focus position is. Screen reader users will know they are on the main area due to the `aria-label`, but it may be unclear to them why the whole area is focusable.

Issue Code

```
<div id="wpbody-content" aria-label="Main content" tabindex="0" style="overflow: hidden;">...</div>


a, div {
    outline: 0;
}
#wpbody-content {
    ...
    overflow: visible !important;
}
```

Remediation Guidance

If the `tabindex` is being used to allow the main content area to be a working destination for skip links, then use `tabindex="-1"` instead. This will allow the element to be focusable without placing it in the Tab order.

This change will not affect keyboard users' ability to scroll the page when the element has focus.

Recommended Code

```
<div id="wpbody-content" aria-label="Main content" tabindex="-1" style="overflow: hidden;">...</div>
```

Relevant standards

- 2.1.1 Keyboard (Level A)
- 2.4.7 Focus Visible (Level AA)

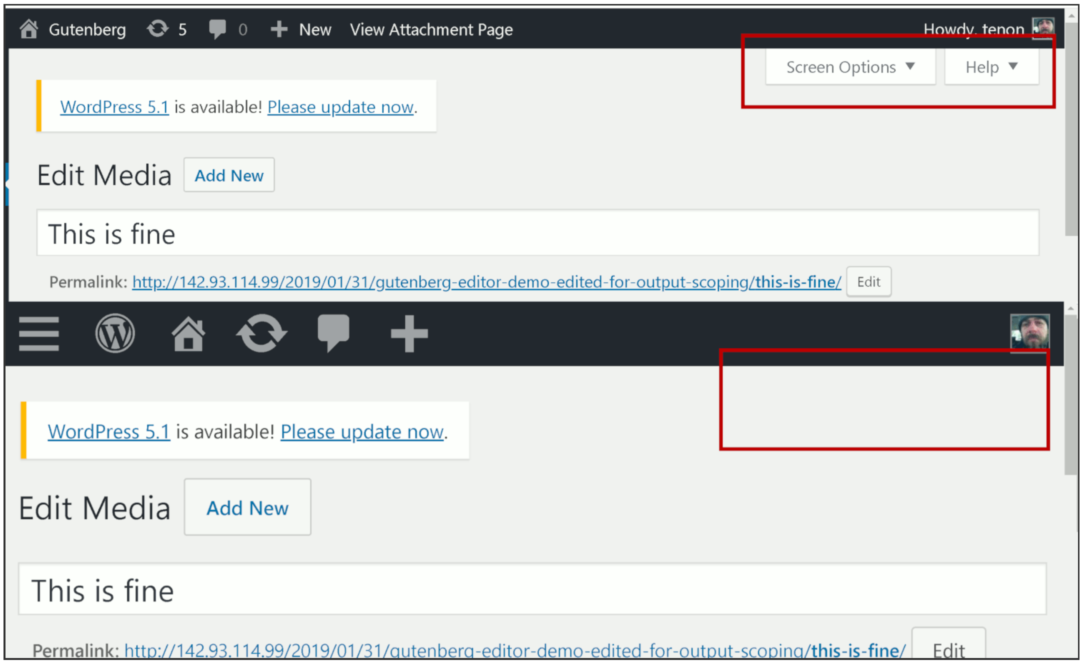
GUT-79 - Options vanish at higher zoom levels

- **Severity:** Medium
- **Affected Populations:** Low-Vision, Cognitively Impaired
- **Platform(s):** All/ Universal
- **Components affected:** Edit Media

Issue description

On the Edit Media page there are two buttons ("Screen Options" and "Help") which offer additional settings and information. These buttons are missing when the page is enlarged with browser zoom above 150%.

Users who use browser zoom to enlarge text and clickable areas may not realise there are options available to add media metadata such as captions and descriptions, nor that there's an explanatory overview.



WordPress 5.1 is available! Please update now.

Edit Media

This is fine

Permalink: http://142.93.114.99/2019/01/31/gutenberg-editor-demo-edited-for-output-scoping/this-is-fine/

WordPress 5.1 is available! Please update now.

Edit Media

This is fine

Permalink: http://142.93.114.99/2019/01/31/gutenberg-editor-demo-edited-for-output-scoping/this-is-fine/

Issue Code

```
@media screen and (max-width: 782px) {
    #collapse-menu,
    #screen-meta,
    #screen-meta-links,
    .post-format-select {
        display: none!important;
    }
}
```

Remediation Guidance

Using the same media query, find another suitable design to retain these options and keep them visible, or offer an alternative method of reaching these options.

Relevant standards

- 1.4.10 Reflow (Level AA)

Related Issues

- Related to "Cursor suggests functionality even when functionality is not present"

GUT-58 - Tab structure does not use tab semantics

- **Severity:** Medium
- **Affected Populations:** Blind, Low-Vision, Motor Impaired, Cognitively Impaired
- **Platform(s):** Windows - Screen Reader, Windows - ZoomText, Windows - Dragon, Mac - VoiceOver, Android - TalkBack, iOS - VoiceOver
- **Components affected:** Media Dialog

Issue description

In the "Featured Image" modal dialog, two links are used to switch views (between the "Media Library" and the "Upload Files" panels). These are coded as links with redundant `href` attributes ("#"), which is not semantically valid, and does not convey the currently-active link to assistive technologies.

The choice of button vs link has a significant effect on user expectations for assistive technology users: links convey the expectation that the browser will navigate to a different location or page, which standard buttons do not. The difference also affects what keys can be used within assistive technologies to navigate between the elements, and which page summaries the element appears in (for example, it won't show up in a list of all form elements on the page).

Semantic markup is vital for assistive technology users, to convey the structure and relationships between information. When appropriate semantic markup is not used, it may be more difficult for such users to understand what the content means, and to comprehend the structure that the visual presentation conveys.

Issue Code

```
<div class="media-router">
    <a href="#" class="media-menu-item active">Upload Files</a>
    <a href="#" class="media-menu-item">Media Library</a>
</div>
```

Remediation Guidance

The visual presentation of the content within the "Featured Image" dialog is a tab panel. Using the tab panel pattern https://www.w3.org/TR/wai-aria-practices-1.1/#tabpanel will give the elements proper roles, and using `aria-selected` will allow users of assistive technologies to know which tab is currently selected.

Recommended Code

```
<div role="tablist" class="media-router">
    <button role="tab" aria-selected="true" class="media-menu-item active">Upload Files</button>
    <button role="tab" class="media-menu-item">Media Library</button>
</div>
```

Relevant standards

- 1.3.1 Info and Relationships (Level A)
- 4.1.2 Name, Role, Value (Level A)

GUT-49 - Insufficient labelling and inappropriate nesting of form controls

- **Severity:** Low
- **Affected Populations:** Blind, Low-Vision, Cognitively Impaired
- **Platform(s):** Windows - Screen Reader, Windows - ZoomText, Mac - VoiceOver, Android - TalkBack, iOS - VoiceOver
- **Components affected:** Media Dialog

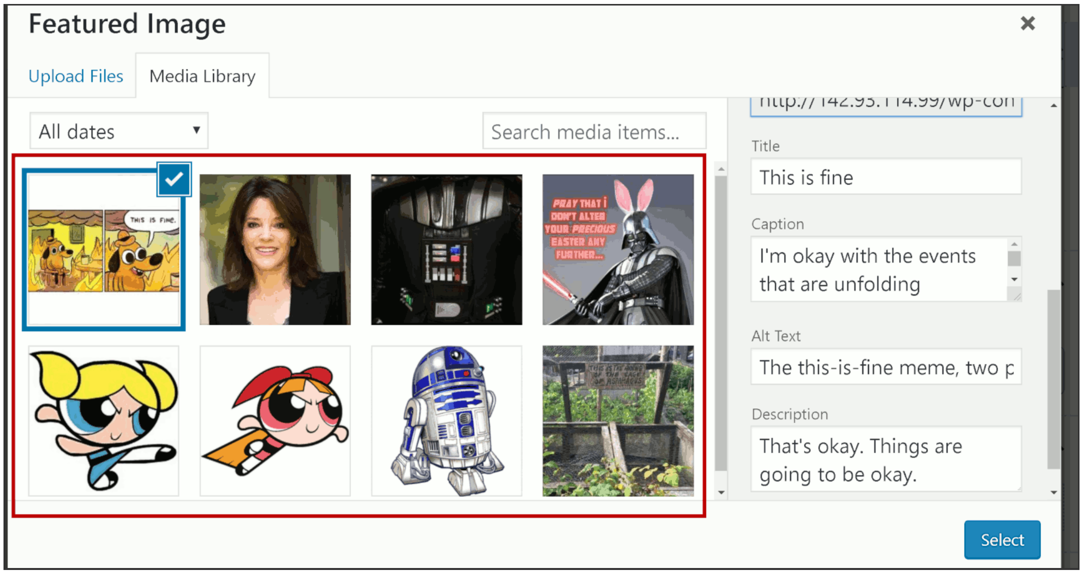
Issue description

In the "Featured Image" modal dialog's "Media Library" tab, users can choose an image to be the featured image; this is presented as a series of checkbox elements which, once checked, show a button to uncheck inside the custom checkbox element.

This button is in the Tab order, however its only name is "Deselect", it doesn't tell the user what is being deselected. Users must infer what this button does from context, and may wonder why it's next to seemingly-ordinary checkboxes which themselves are natively unselectable.

The `button` role of these buttons does not convey that the user is checking or unchecking an item, and their placement inside the `checkbox` elements is semantically confusing, and may be functionally confusing since one focusable element is nested inside another.

Screen reader users hear that a list is present, yet find that they cannot navigate by list-items.



http://142.93.114.99/wp-con

Title
This is fine

Caption
I'm okay with the events that are unfolding

Alt Text
The this-is-fine meme, two p

Description
That's okay. Things are going to be okay.

Issue Code

```
<ul tabindex="-1" class="attachments ui-sortable ui-sortable-disabled" id="__attachments-view-53">
    <li tabindex="0" role="checkbox" aria-label="Marianne_Williamson" aria-checked="true" ... class="...">
        <div class="...">
            <div class="thumbnail">
                <div class="centered">
                    <img src="...this-is-fine-300x200.jpg" draggable="false" alt="">
                </div>
            </div>
        </div>
        <button type="button" class="check" tabindex="0">
            <span class="..."></span>
            <span class="screen-reader-text">Deselect</span>
        </button>
    </li>
    ...
</ul>
```

Remediation Guidance

Allow the list semantics to remain available by not overriding the `<li>` elements' native list-item roles.

Remove the `<button>` inside the checkbox and allow the original checkbox itself to be hidden and shown (perhaps styled with the custom checkmarks and "-" symbols, using the `:before` pseudo-element, as used in other checkboxes); hiding elements with `opacity` still allows speech recognition and screen readers to perceive the checkbox as though it were visible. The `alt` attribute inside the image will be able to label the checkbox if it's inside the `<label>` element.

The checkbox now can offer both the image name and its state together in one unit, allowing users to always know what they are deselecting.

Recommended Code

```
<fieldset>
    <legend class="screen-reader">Media options result</legend>
    <ul tabindex="-1" class="attachments ui-sortable ui-sortable-disabled" id="__attachments-view-53">
        <li ... class="...">
            <input type="checkbox" id="data_159">
            <label for="data_159">
                <img src="...this-is-fine-300x200.jpg" draggable="false" draggable="false"
alt="Marianne_Williamson">
            </label>
        </li>
        ...
    </ul>
</fieldset>
```

```
li {
    position: relative;
}
li>input {
    position: absolute;
    opacity: 0;
    ...
}
li input+label img {
    ...
}
li input:focus+label img {
    /* styles for focussed input */
    /* include an actual outline in addition to box-shadow for high contrast users */
}
li input:checked {
    opacity: 1;
}
li input:checked:focus,
li input:checked:hover {
    /* change from checkmark to dash */
}
li input:checked+label img {
    ...
}
```

Relevant standards

- 1.3.1 Info and Relationships (Level A)

Related Issues

- Related to "Insufficient form semantics"

GUT-35 - Text styled as headings are not headings

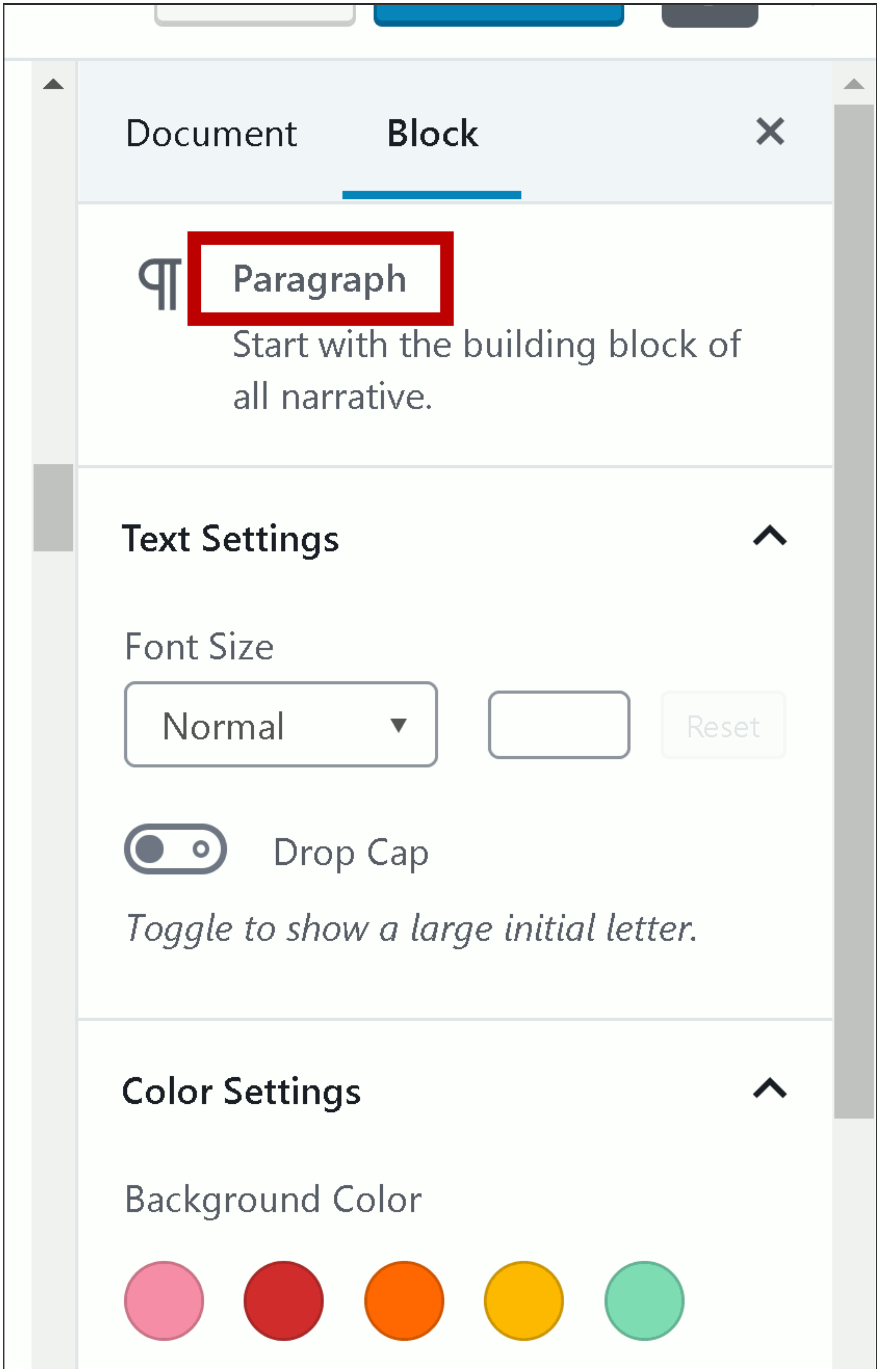
- **Severity:** Low
- **Affected Populations:** Blind, Cognitively Impaired
- **Platform(s):** All/ Universal
- **Components affected:** Block Panel, Title Block

Issue description

At the top of the Block panel, the name of the block type is styled as a heading, and functions as a heading for the content describing the block type. It has a class name suggesting it is meant as a section title.

In some places in the Block panel and in the Title Block, a label-input pair is imitated with a paragraph-result pair. There is no semantic relationship between the paragraph and the content following it.

Headings provide important wayfinding points for assistive technology users, and help to provide context for the content that follows them. When headings are missing or improperly defined, it may be more difficult for such users to understand and navigate the content.



¶ Paragraph

Start with the building block of all narrative.

Text Settings

Font Size

Normal ▾

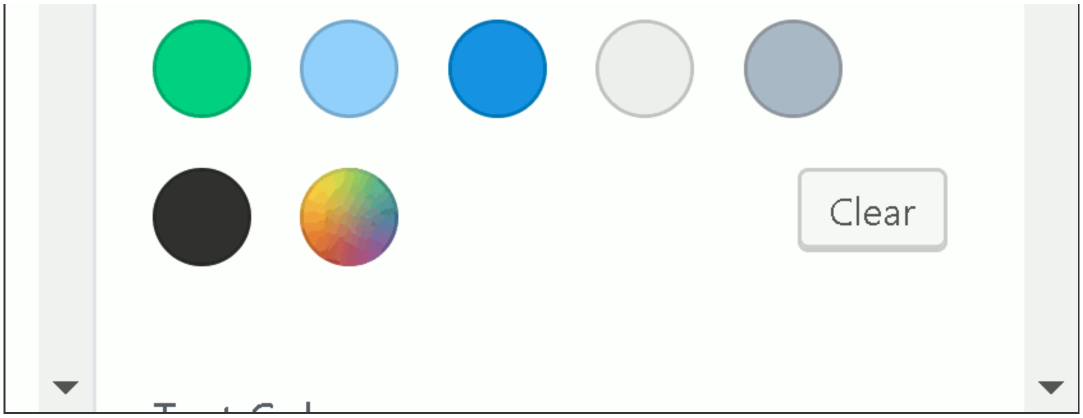
Drop Cap

Toggle to show a large initial letter.

Color Settings

Background Color

● ● ● ● ●



Issue Code

```
<div class="editor-block-inspector__card-title">Paragraph</div>
...
<div class="components-panel__body is-opened">
    ...
    <label...>URL</label>
    <input ...>
    ...
    <p class="edit-post-post-link__preview-label">Preview</p>
    <a target="_blank" class="..." href="...">...</a>
    ...
</div>
```

Remediation Guidance

Ensure elements presented and styled as headings are, in fact, headings.

In places where the format is switching between a label-input pair to a label-result pair, use headings to semantically represent that the label text is giving context to the content which follows it.

Recommended Code

```
<h2 class="editor-block-inspector__card-title">Paragraph</h2>
...
<div class="components-panel__body is-opened">
    ...
    <label...>URL</label>
    <input ...>
    ...
    <h3  class="edit-post-post-link__preview-label">Preview</h3>
    <a target="_blank" class="..." href="...">...</a>
    ...
</div>
```

Relevant standards

- 1.3.1 Info and Relationships (Level A)
- 2.4.6 Headings and Labels (Level AA)
- 4.1.2 Name, Role, Value (Level A)

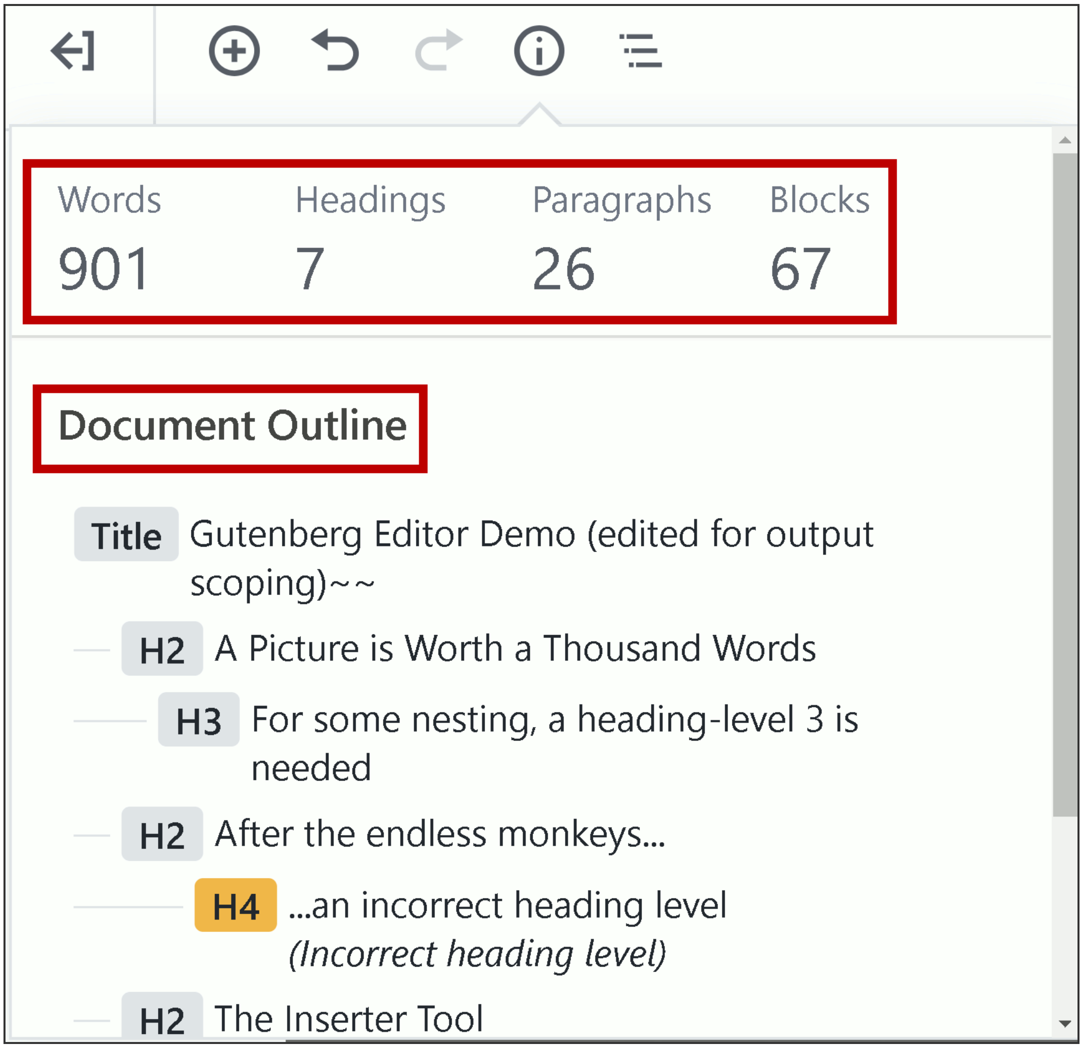
GUT-25 - Content lacks semantic markup

- **Severity:** Low
- **Affected Populations:** Blind, Low-Vision, Cognitively Impaired
- **Platform(s):** All/ Universal
- **Components affected:** Structure and Navigation

Issue description

The Content Structure panel includes a summary of the content by type, and some text that's styled and positioned like a heading ("Document Outline"), however this content does not use appropriate semantic markup.

Semantic markup is vital for assistive technology users, to convey the structure and relationships between information. When appropriate semantic markup is not used, it may be more difficult for such users to understand what the content means, and to comprehend the structure that the visual presentation conveys.



Words	Headings	Paragraphs	Blocks
901	7	26	67

Document Outline

- Title Gutenberg Editor Demo (edited for output scoping)~~
  - H2 A Picture is Worth a Thousand Words
    - H3 For some nesting, a heading-level 3 is needed
  - H2 After the endless monkeys...
      - H4 ...an incorrect heading level *(Incorrect heading level)*
  - H2 The Inserter Tool

Issue Code

```
<div class="table-of-contents__counts" role="note" aria-label="Document Statistics" tabindex="0">
    <div class="table-of-contents__count">Words<span class="word-count">901</span></div>
    <div class="table-of-contents__count">Headings<span class="table-of-contents__number">7</span></div>
    <div class="table-of-contents__count">Paragraphs<span class="table-of-contents__number">27</span></div>
    <div class="table-of-contents__count">Blocks<span class="table-of-contents__number">68</span></div>
</div>

<span class="table-of-contents__title">Document Outline</span>
```

Remediation Guidance

Use a structure such as `<ul>` or `<dl>` to semantically associate the content type information.

Elements which are styled and positioned like headings should be marked-up as headings. The Recommended Code uses an `<h2>` on the assumption that the open panel should be treated as standalone context within the page (i.e. at a level beneath the page's `<h1>`).

Recommended Code

```
<div class="table-of-contents__counts" role="note" aria-label="Document Statistics" tabindex="0">
    <dl>
        <div class="table-of-contents__count">
            <dt>Words</dt>
            <dd class="word-count">901</dd>
        </div>
        <div class="table-of-contents__count">
            <dt>Headings</dt>
            <dd class="table-of-contents__number">7</dd>
        </div>
        <div class="table-of-contents__count">
            <dt>Paragraphs</dt>
            <dd class="table-of-contents__number">27</dd>
        </div>
        <div class="table-of-contents__count">
            <dt>Blocks</dt>
            <dd class="table-of-contents__number">68</dd>
        </div>
    </dl>
</div>

<h2 class="table-of-contents__title">Document Outline</h2>
```

Relevant standards

- 1.3.1 Info and Relationships (Level A)
- 2.4.6 Headings and Labels (Level AA)
- 4.1.2 Name, Role, Value (Level A)

Related Issues

- Related to "Content overlaps at 200% text enlarge"

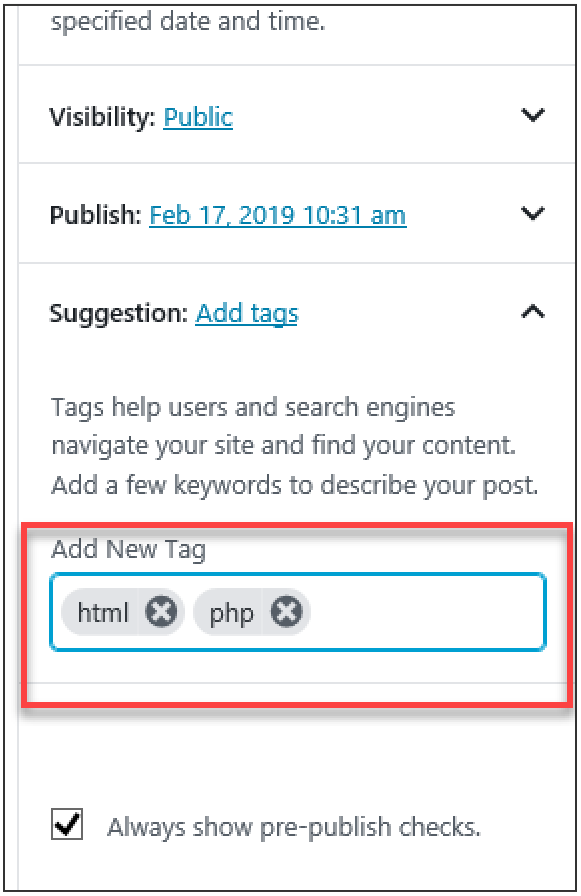
GUT-14 - Suggested tags should be a list

- **Severity:** Low
- **Affected Populations:** Blind, Low-Vision, Cognitively Impaired
- **Platform(s):** Windows - Screen Reader, Mac - VoiceOver
- **Components affected:** Publish and Unpublish

Issue description

Users are provided with an option to add tags when publishing, however the tag list is not marked up as a list. Additional text elements have been used to convey the number of items ("1 out of 3" etc.), however these do not sufficiently compensate for the lack of structural semantics.

Using list markup will aid comprehension for assistive technology users, by conveying the structure of the content and providing an overview of the number of items.



specified date and time.

Visibility: Public

Publish: Feb 17, 2019 10:31 am

Suggestion: Add tags

Tags help users and search engines navigate your site and find your content. Add a few keywords to describe your post.

Add New Tag

html ⊗ php ⊗

☑ Always show pre-publish checks.

Issue Code

```
di    bi d   "  "  l    "   "
```

```
<div tabindex="-1" class="...">
    <span class="components-form-token-field__token">
        <span class="components-form-token-field__token-text" id="components-form-token-field__token-text-3">
            <span class="screen-reader-text">html (1 of 3)</span>
            <span aria-hidden="true">html</span>
        </span>
        <button class="..." aria-describedby="components-form-token-field__token-text-3" aria-label="Remove Tag" type="button">
            ...
        </button>
    </span>
    <span class="components-form-token-field__token">
        <span class="components-form-token-field__token-text" id="components-form-token-field__token-text-4">
            <span class="screen-reader-text">htm (2 of 3)</span>
            <span aria-hidden="true">htm</span>
        </span>
        <button class="components-button components-icon-button components-form-token-field__remove-token" aria-describedby="components-form-token-field__token-text-4" aria-label="Remove Tag" type="button">
            ...
        </button>
    </span>
    ...
    <input class="components-form-token-field__input" id="components-form-token-input-1" role="combobox"...>
    ...
</div>
```

Remediation Guidance

Convert this markup to an unordered-list (`<ul>`) with individual list-items (`<li>`).

The `.screen-reader-text` instructions can then be removed, and the `aria-hidden="true"` should be removed from the remaining visible `<span>`.

Recommended Code

```
<div class="components-form-token-field__input-container" tabindex="-1">
    <ul class="components-form-token-field__token">
        <li>
            <span id="components-form-token-field__token-text-0" class="...">php</span>
            <button type="button" aria-label="Remove Tag" aria-describedby="components-form-token-field__token-text-0" class="...">...</button>
        </li>
        <li>
            <span id="components-form-token-field__token-text-4" class="...">html</span>
            <button type="button" aria-label="Remove Tag" aria-describedby="components-form-token-field__token-text-4" class="...">...</button>
        </li>
        ...
    </ul>
    <input id="components-form-token-input-6" type="text" size="3" class="..." role="combobox"...>
    ...
</div>
```

Relevant standards

- 1.3.1 Info and Relationships (Level A)

Related Issues

- Related to "Useful content is unnecessarily hidden"

Content Type: Dynamic Content

GUT-94 - Changes and updates are not announced to assistive technologies

- **Severity:** Medium
- **Affected Populations:** Blind, Low-Vision, Cognitively Impaired
- **Platform(s):** Windows - Screen Reader, Windows - ZoomText, Mac - VoiceOver, Android - TalkBack, iOS - VoiceOver
- **Components affected:** Block Editing, Block Options

Issue description

Many of the Block functions are not announced to assistive technologies. Although some functions are announced (such as creating a new block), many other functions are not (such as hiding or showing the Block panel, removing/deleting blocks, duplicating blocks, and actions such as Undo and Redo).

Some actions are quite easy to perform by mistake. For example, when users hit the Enter key while their cursor is at the end of any text or content in a block, this will trigger the creation of a new block. Although this is announced, it's so easy to trigger unintentionally that users may come to rely on the Undo and Redo states to maintain the output they want, and these states are not announced.

Users who are blind or who have low-vision must determine what's changed by navigating around the various blocks, and this may be particularly difficult for users with a cognitive disability. Users attempting to removing several duplicates or blocks which happen to have a lot of similar text must inspect most of the block text to ascertain their current state, and requires users with print disabilities to rely on short-term memory more than sighted users.

Remediation Guidance

Use a short `aria-live` region announcement for actions which makes a visible change to the document or interface and which aren't already announced, such as block duplication or removal, opening and closing the Block panel, and Redo and Undo actions.

An example of the success state being announced is when a link is removed: assistive technologies announce "Link removed".

Also consider whether an "earcon" (a short noise signalling success or failure) might be appropriate for some actions.

Additionally, it would be helpful to provide the ability to silence these announcements, via a user profile setting, for those who find the announcements unnecessary (such as a sighted or partially-sighted screen reader user).

Recommended Code

```
<!-- example current existing success status message -->

<div id="wp-a11y-speak-assertive" aria-live="assertive" aria-relevant="additions text" aria-atomic="true"
class="screen-reader-text wp-a11y-speak-region">Link removed.</div>


<!-- example additional status message -->

<div id="wp-a11y-speak-assertive" aria-live="assertive" aria-relevant="additions text" aria-atomic="true"
class="screen-reader-text wp-a11y-speak-region">Block duplicated.</div>
```

Relevant standards

- 3.2.5 Change on Request (Level AAA)
-

Related Issues

- Related to "Undo and Redo functions do not communicate changes"

GUT-73 - Status message not exposed to assistive technologies

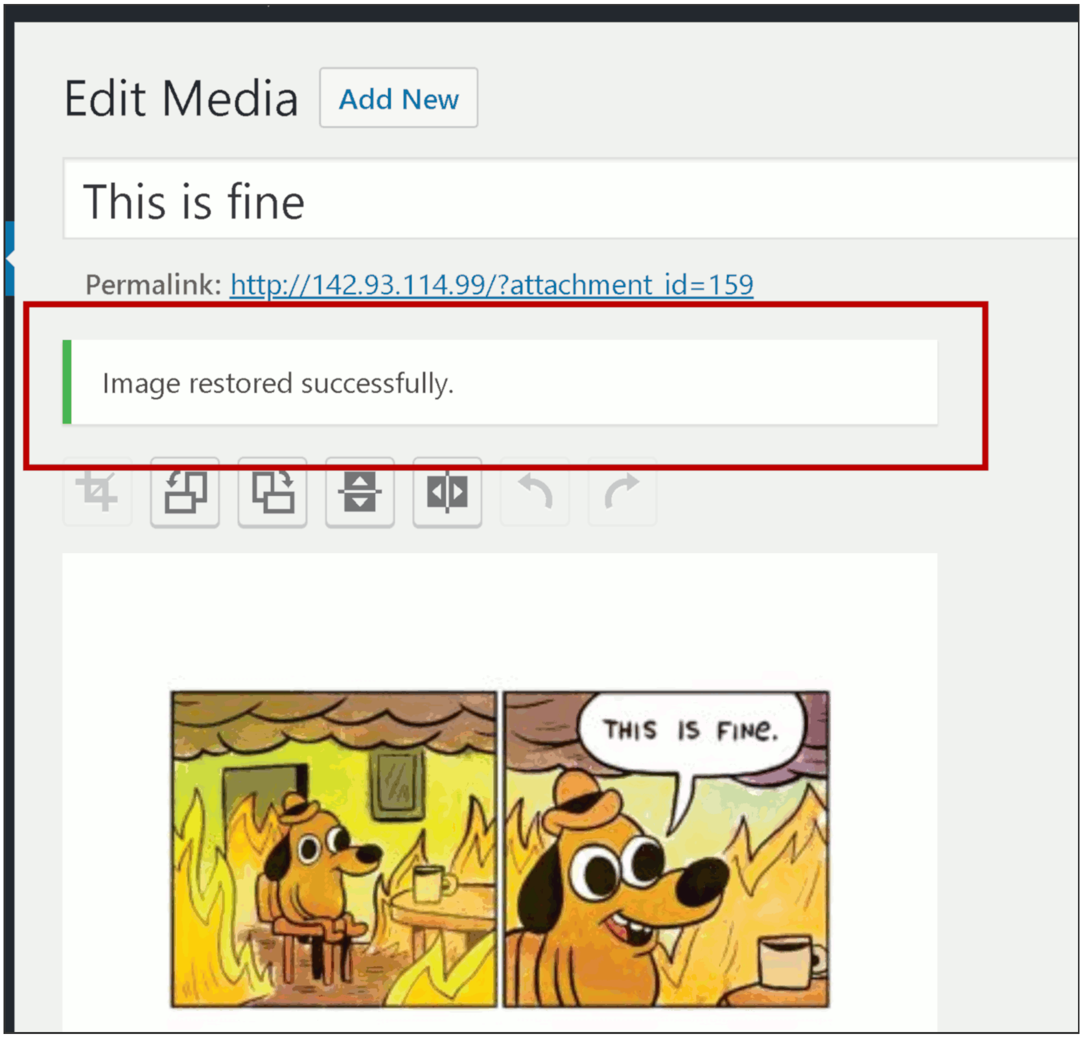
- **Severity:** Medium
- **Affected Populations:** Blind, Low-Vision, Cognitively Impaired
- **Platform(s):** Windows - Screen Reader, Mac - VoiceOver, Android - TalkBack, iOS - VoiceOver
- **Components affected:** Edit Media

Issue description

Within the Edit Media page, when activating the "Restore Image" button, a message is shown above the image while the Restore button itself disappears.

Since the button would have been focused at the time when activated by keyboard, this causes the keyboard focus position to be lost and reset to the top of the page.

The message itself is also not announced by screen readers, and may not be visible to screen magnification users if it appears outside their current view.



Edit Media

Add New

This is fine

Permalink: http://142.93.114.99/?attachment_id=159

Image restored successfully.

Issue Code

```
di   l    "i   di      l             l   fi "
```

```
<div class="imgedit-panel-content wp-clearfix">
    <div class="error">
        <p>Cannot save image metadata.</p>
    </div>
    <div class="imgedit-menu wp-clearfix">
        ...
    </div>
</div>

...

<div class="imgedit-panel-content wp-clearfix">
    <div class="updated">
        <p>Image restored successfully.</p>
    </div>
    <div class="imgedit-menu wp-clearfix">
        ...
    </div>
</div>
```

Remediation Guidance

Add `tabindex="-1"` to the message, and then programatically focus it when it appears. This will maintain a logical focus position, and ensure that all visual users will see it, while assistive technologies announce it.

Adding the `alert` role will also help to convey to screen readers that this is an alert-type message.

Recommended Code

```
<div class="imgedit-panel-content wp-clearfix">
    <div class="error" tabindex="-1" role="alert">
        <p>Cannot save image metadata.</p>
    </div>
    <div class="imgedit-menu wp-clearfix">
        ...
    </div>
</div>

...

<div class="imgedit-panel-content wp-clearfix">
    <div class="updated" tabindex="-1" role="alert">
        <p>Image restored successfully.</p>
    </div>
    <div class="imgedit-menu wp-clearfix">
        ...
    </div>
</div>
```

Relevant standards

- 2.1.1 Keyboard (Level A)
-

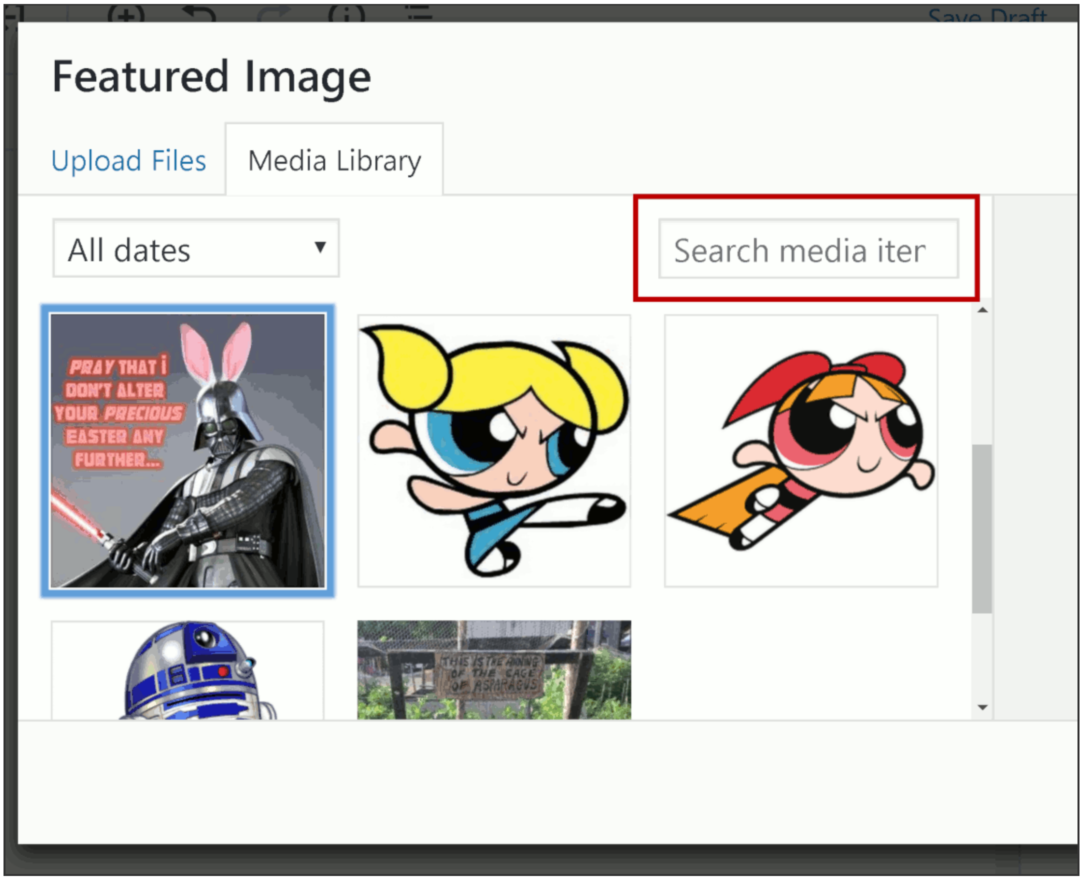
GUT-63 - Live search does not announce results to assistive technologies

- **Severity:** Medium
- **Affected Populations:** Blind, Low-Vision, Cognitively Impaired
- **Platform(s):** Windows - Screen Reader, Mac - VoiceOver, Android - TalkBack, iOS - VoiceOver
- **Components affected:** Media Dialog

Issue description

When users in the "Featured Image" modal's "Media Library" do a search using the search/filter control, the results change dynamically, however these changes are not announced to screen readers, so blind and low-vision users cannot tell what, if anything, matches the search without manually reading through the image list.

Assistive technologies rely on state information to understand the behaviour of dynamic content, and if this information is incomplete or inaccurate it may create a confusing experience for users.



Issue Code

```
<div class="media-toolbar-primary search-form">
    <label for="media-search-input" class="screen-reader-text">Search Media</label>
    <input type="search" placeholder="Search media items..." id="media-search-input" class="search">
</div>
```

Remediation Guidance

Use a live region to announce the number of matched results, updating as users type and delete characters (ideally with a short timeout or throttling so that the announcements don't interfere with user typing).

Relevant standards

GUT-42 - Feature Image dialog does not follow the dialog pattern

- **Severity:** High
- **Platform(s):** Windows - Screen Reader, Windows - ZoomText, Mac - VoiceOver, Android - TalkBack, iOS - VoiceOver
- **Components affected:** Media Dialog

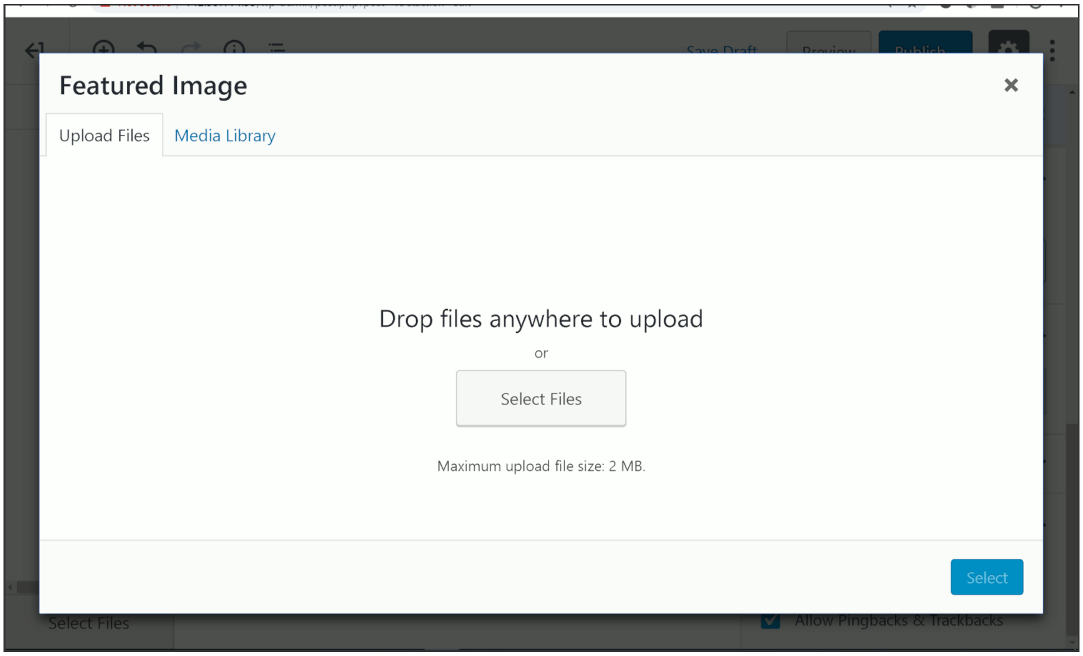
Issue description

Users opening the "Feature Image" dialog are not told they are entering a dialog. Keyboard focus is trapped, however screen reader users are able to go beyond the last items of the dialog without realising they have left it.

Keyboard users who find their focus cycling around the dialog get a Tab stop with no visible focus. This is the dialog itself, which can receive keyboard focus while having no visible focus state.

The triggering "Set Featured Image" button does not express an expanded or collapsed state, which would not matter if users could not reach the button when the modal was open; however, they can.

Lack of an explicit role for modal dialogs may be confusing for assistive technology users, since they may not realise they're inside a dialog, and that consequently the keyboard interactions may be different from the rest of the page. Lack of an explicit label for dialogs may be confusing for assistive technology users, since they won't have an immediate sense of what the dialog is for.


Featured Image
×
Upload Files
Media Library
Drop files anywhere to upload
or
Select Files
Maximum upload file size: 2 MB.
Select
Select Files
Allow Pingbacks & Trackbacks

Issue Code

```
<button type="button" class="components-button editor-post-featured-image__toggle">Set featured image</button>
...
<div tabindex="0" class="media-modal wp-core-ui">
    <button type="button" class="...">...Close media panel...</button>
    <div class="media-modal-content">
        ...
        <div class="media-frame-title">
            <h1>Featured Image<span class="dashicons dashicons-arrow-down"></span></h1>
        </div>
        ...
    </div>
</div>
```

Remediation Guidance

Give the modal a `role` of `dialog` and set `"aria-modal=true"`. Additionally, set `aria-hidden="true"` on the element which wraps the rest of the page, to prevent older browsers or assistive technologies that don't support `aria-modal` from accessing the rest of the page while the modal is open.

Use `aria-labelledby` and an `id` on the modal heading to name the modal.

Change the `tabindex` to `-1` so that it can be focused via JavaScript without being in the Tab order. The exception to this advice is if having the modal in the Tab order is necessary so that keyboard users can scroll content inside it; however in existing cases, scrollable areas are separate parts of the modal. But if a container must receive focus in order to allow keyboard users to scroll the content, then the container must visibly show focus.

Recommended Code

```
<div id="wpwrap" aria-hidden="true">...</div>

<div role="dialog" aria-modal="true" aria-labelledby="media-modal-title" tabindex="-1" class="media-modal wp-core-ui">
    <button type="button" class="...">...Close media panel...</button>
    <div class="media-modal-content">
        ...
        <div class="media-frame-title">
            <h1 id="media-modal-title">Featured Image<span class="dashicons dashicons-arrow-down"></span></h1>
        </div>
        ...
    </div>
</div>
```

Relevant standards

- 1.3.1 Info and Relationships (Level A)
- 4.1.2 Name, Role, Value (Level A)

Related Issues

- Related to "Insufficient form semantics"

GUT-41 - Text inadvertently rendered by assistive technologies

- **Severity:** Medium
- **Affected Populations:** Blind, Low-Vision, Cognitively Impaired
- **Platform(s):** Windows - Screen Reader, Mac - VoiceOver, Android - TalkBack, iOS - VoiceOver
- **Components affected:** Block Panel, Document Panel, Media Dialog

Issue description

Users of assistive technologies such as screen readers who navigate to the bottom of the Settings panels will find a button which they cannot activate (nor see if sighted) called "Select files".

Additionally, if users change the color modes in the popup custom color picker, the current mode is announced in a live region. However, long after users are done with choosing a color, even after choosing to edit another block on the page, when they reach the bottom of the Block panel they'll still hear "hex color mode active". At this point, users may not remember what this was for and have no idea what this is referencing, as it no longer has any context.

Issue Code

```
<!-- color announcement -->

<div id="wp-a11y-speak-polite" aria-live="polite" aria-relevant="additions text" aria-atomic="true" class="screen-reader-text wp-a11y-speak-region">Hex color mode active</div>


<!-- select files -->

<div class="wp-uploader-browser" id="wp-uploader-browser-o_1d2saqvj4l3skgruvl1ejo1ejp3" style="position: fixed; top: -1000px; left: -1000px; height: 0px; width: 0px;">

    <button type="button" class="browser button button-hero" id="__wp-uploader-id-1" style="display: inline-block; position: relative; z-index: 1;">Select Files</button>

</div>
```

Remediation Guidance

When users have performed an action, such as clicking another block (or whatever action causes "No block selected" to appear in the Block panel), clear the live region so that users who encounter it while manually reading do not hear it. Refill the live region when users change color modes.

The hidden file selection button should be hidden from all users with `display: none` whenever it is not visible nor meant to be used.

Recommended Code

```
<!-- color announcement -->

<div id="wp-a11y-speak-polite" aria-live="polite" aria-relevant="additions text" aria-atomic="true" class="screen-
reader-text wp-a11y-speak-region"> </div>


<!-- select files -->

<div class="wp-uploader-browser" id="wp-uploader-browser-o_1d2saqvj4l3skgruvl1ejo1ejp3" style="display:none;">

    <button type="button" class="browser button button-hero" id="__wp-uploader-id-1" style="display: inline-block;
position: relative; z-index: 1;">Select Files</button>

</div>
```

GUT-10 - Undo and Redo functions do not communicate changes

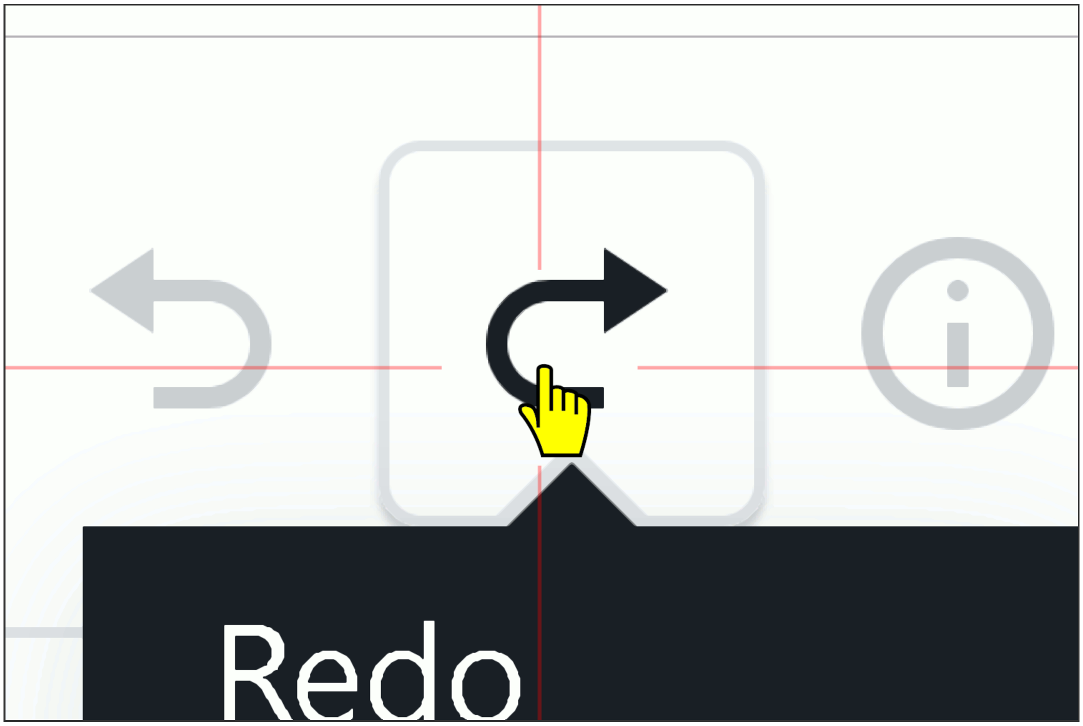
- **Severity:** Medium
- **Affected Populations:** Blind, Low-Vision, Cognitively Impaired
- **Platform(s):** Windows - Screen Reader, Windows - ZoomText, Mac - VoiceOver, Android - TalkBack, iOS - VoiceOver
- **Components affected:** Editor Top Bar

Issue description

When users navigate to the Editor Top Bar after writing some content, and use the "Undo" or "Redo" buttons, the only way to know what changed or whether it was successful is to be able to see the page content while activating it.

For a screen reader user, activating the control with Enter communicates nothing, while using Space merely repeats the Space key. Some screen readers announce when the button's state goes from enabled to disabled, while others don't.

Low-vision users who are either zoomed-in or using screen magnification cannot see changes caused by Undo and Redo unless their viewable area includes the content that has changed; in this case, the only apparent state is when they've reached the end of the Undo or Redo buffer, because the buttons change to the disabled colors.



Remediation Guidance

One solution could be an `aria-live` region which describes the change. For example, if a block creation was undone, the live region text might be "Block deleted."

Another idea, which would assist low-vision and assistive technology users, would be to add a small number next to the button (similar to a notifications-icon number) showing the number of Undo or Redo actions which are left in the buffer; this number could then be removed when the button is disabled. A screen reader user can then have the live region state how many actions are remaining, and changes to this value would then communicate that an Undo or Redo has occurred (to confirm, for example, that the action has only happened once, rather than multiple times due to holding the Space bar down for too long).

Relevant standards

- 1.3.1 Info and Relationships (Level A)
- 1.3.3 Sensory Characteristics (Level A)

Related Issues

- Related to "Changes and updates are not announced to assistive technologies"

Content Type: Forms

GUT-87 - Permalink and its Edit button could be clearer

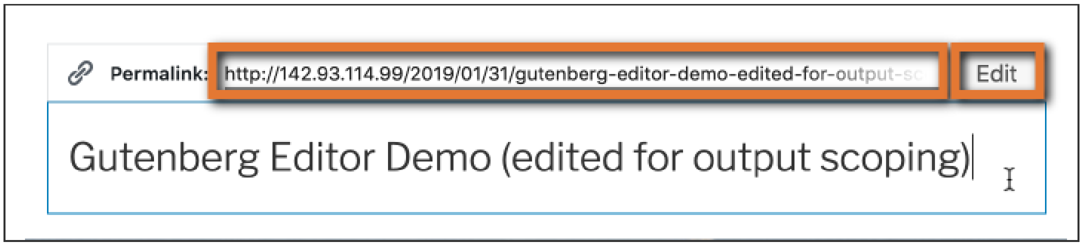
- **Severity:** Low
- **Affected Populations:** Blind, Cognitively Impaired
- **Platform(s):** All/ Universal
- **Components affected:** Title Block

Issue description

The "Permalink" section of the Title Block has an "Edit" button which is not semantically associated with the permalink except by context. Similarly, the permalink itself has the word "Permalink:" before it, but this is not explicitly associated either.

Although association by context is generally sufficient for content like this, it would nevertheless be helpful to add a more descriptive label to the button, and to describe the fact that the link is a permalink within its screen reader text.

These changes would help to make the meaning clearer to screen reader users, especially those with a cognitive disability.



http://142.93.114.99/2019/01/31/gutenberg-editor-demo-edited-for-output-sc

Gutenberg Editor Demo (edited for output scoping)

Issue Code

```
<span class="editor-post-permalink__label">Permalink:</span>
<a target="_blank" class="..." href="http://142.93.114.99/?p=19" rel="external noreferrer noopener">
    http://142.93.114.99/2019/01/31/gutenberg-editor-demo-edited-for-output-scoping/
    <span class="screen-reader-text">(opens in a new tab)</span>
    ...
</a>
<button type="button" class="...">Edit</button>
```

Remediation Guidance

Add an `aria-label` to the "Edit" button which explains what it does more clearly, and amend the `.screen-reader-text` of the permalink to do the same.

Recommended Code

```
<span class="editor-post-permalink__label">Permalink:</span>
<a target="_blank" class="..." href="http://142.93.114.99/?p=19" rel="external noreferrer noopener">
    http://142.93.114.99/2019/01/31/gutenberg-editor-demo-edited-for-output-scoping/
    <span class="screen-reader-text">(permalink opens in a new tab)</span>
    ...
</a>
<button aria-label="Edit permalink" type="button" class="...">Edit</button>
```

Relevant standards

- 1.3.1 Info and Relationships (Level A)
- 3.3.2 Labels or Instructions (Level A)

GUT-81 - Disabled fieldset does not convey the disabled state of controls

- **Severity:** Medium
- **Affected Populations:** Blind, Low-Vision, Motor Impaired, Cognitively Impaired
- **Platform(s):** All/ Universal
- **Components affected:** Edit Media

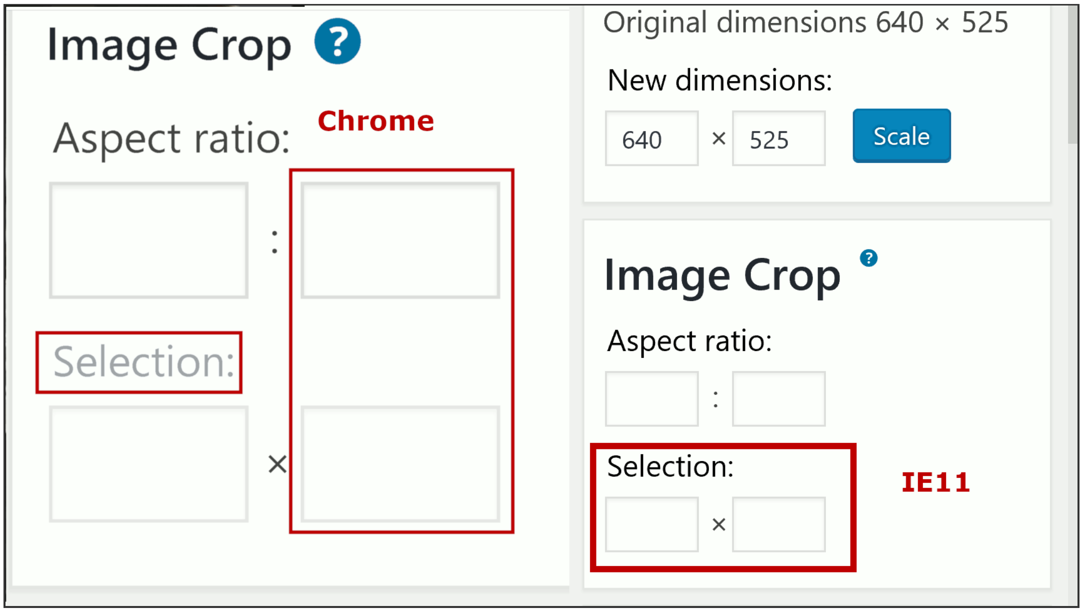
Issue description

On the Edit Media page, when users haven't made any selections for cropping the image, the "Selection" legend is grayed-out to indicate that this section is disabled, and the inputs are meant to be disabled by use of the `disabled` attribute on the fieldset itself.

The only visible clue that the fields are meant to be disabled is that the legend is a slightly lighter color; the fields themselves appear identical to enabled fields. The state is only revealed to users by a mouse user's inability to click inside, and a keyboard user's inability to focus them. Speech recognition software may indicate that the disabled fields are enabled when users call out fields by type.

Additionally, the change of style is not even conveyed in IE11, in which the "Selection" legend looks identical to enabled groups.

This can be confusing to users with cognitive disabilities and speech recognition users.



Issue Code

```
<fieldset id="imgedit-crop-sel-159" class="imgedit-crop-sel disabled" disabled="">
    <legend>Selection:</legend>
    <div class="nowrap">
        <label><span class="screen-reader-text">selection width</span>
            <input type="text" id="imgedit-sel-width-159" ...>
        </label>
        <span class="imgedit-separator">×</span>
        <label><span class="screen-reader-text">selection height</span>
            <input type="text" id="imgedit-sel-height-159" ...>
        </label>
    </div>
</fieldset>
...
.imgedit-settings .disabled, .media-disabled {
    color: #a0a5aa;
}
```

Remediation Guidance

While the `disabled` attribute is allowed on `<fieldset>` elements, it should be avoided as the sole method of conveying disabled states, because it does not universally apply the `disabled` state to the fields in all assistive technologies.

Instead, place the `disabled` attribute on the fields directly.

Explicitly set the `<legend>` style changes with a `legend` selector so that it applies in all browsers, and style disabled controls to be more obviously disabled; the current slight difference in `box-shadow` is difficult to discern even for users without visual disabilities. Ideally, the styling would use `opacity` rather than `color` properties so that the difference is still visible in Windows High Contrast themes.

Also ensure that the relationship between the `<label>` and `<input>` elements is explicitly defined using `for` and `id` attributes, for better compatibility with assistive technologies.

Recommended Code

```
<fieldset id="imgedit-crop-sel-159" class="imgedit-crop-sel disabled">
    <legend>Selection:</legend>
    <div class="nowrap">
        <label for="imgedit-sel-width-159"><span class="screen-reader-text">selection width</span></label>
        <input disabled type="text" id="imgedit-sel-width-159" ...>
        <span class="imgedit-separator">×</span>
        <labelfor="imgedit-sel-height-159"><span class="screen-reader-text">selection height</span></label>
        <input disabled type="text" id="imgedit-sel-height-159" ...>
    </div>
</fieldset>
...
.imgedit-settings .disabled,
.media-disabled,
.disabled legend {
    color: #444;
    opacity: 0.5;
}
input.disabled,
input:disabled,
select.disabled,
select:disabled,
textarea.disabled,
textarea:disabled {
    opacity: 0.5;
}
```

References

https://cccaccessibility.org/web/web-developer-tutorials/using-the-fieldset-and-legend-elements

https://caniuse.com/#feat=fieldset-disabled

Relevant standards

- 1.3.3 Sensory Characteristics (Level A)

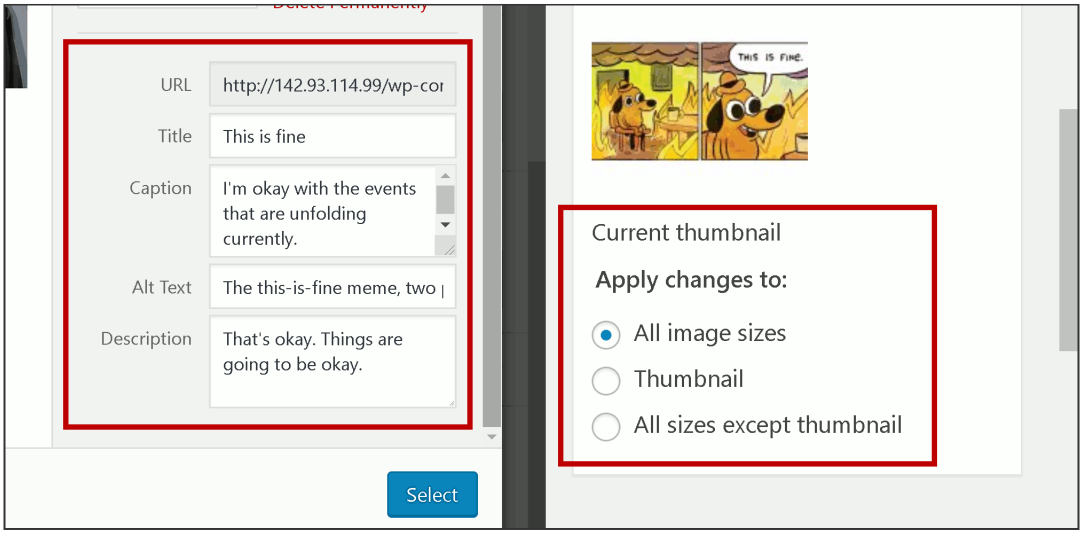
GUT-74 - Radio and checkbox labels rely on implicit association

- **Severity:** Low
- **Affected Populations:** Motor Impaired, Cognitively Impaired
- **Platform(s):** Windows - Dragon
- **Components affected:** Edit Media, Media Dialog

Issue description

In both the "Featured Image" dialog and the Edit Media page, form inputs are wrapped in label elements without explicit association.

Some combinations of browser and assistive technologies do not reliably support implicit association (for example, browsers used with Dragon Naturally Speaking), and so for affected users, the label text may not be announced with the field it relates to.



URL	http://142.93.114.99/wp-cor
Title	This is fine
Caption	I'm okay with the events that are unfolding currently.
Alt Text	The this-is-fine meme, two	
Description	That's okay. Things are going to be okay.

Current thumbnail

Apply changes to:

● All image sizes

○ Thumbnail

○ All sizes except thumbnail

Issue Code

```
 !    di    di
```

```
<!-- Edit Media -->
<div id="imgedit-save-target-159" class="imgedit-save-target">
    <fieldset>
        <legend><strong>Apply changes to:</strong></legend>
        <label class="imgedit-label">
        <input type="radio" name="imgedit-target-159" value="all" checked="checked"> All image sizes</label>
        ...
    </fieldset>
</div>

<!-- Media Dialog -->
<div tabindex="0" data-id="159" class="attachment-details save-ready needs-refresh">
    <h2>Attachment Details ... Saved ...</h2>
    <div class="attachment-info">...</div>
    <label class="setting" data-setting="url">
        <span class="name">URL</span>
        <input type="text" value="http://142.93.114.99/wp-content/uploads/2019/01/this-is-fine.jpg" readonly="">
    </label>
     ...
</div>
```

Remediation Guidance

Use `for` and `id` attributes to explicitly associate each `<label>` with its field.

Recommended Code

```
<!-- Edit Media -->
<div id="imgedit-save-target-159" class="imgedit-save-target">
    <fieldset>
        <legend><strong>Apply changes to:</strong></legend>
        <label for="imgedit-radio-1"  class="imgedit-label">
            <input id="imgedit-radio-1" type="radio" name="imgedit-target-159" value="all" checked="checked">
            All image sizes
        </label>
        ...
    </fieldset>
</div>

<!-- Media Dialog -->
<div tabindex="0" data-id="159" class="attachment-details save-ready needs-refresh">
    <h2>Attachment Details ... Saved ...</h2>
    <div class="attachment-info">...</div>
    <label for="imgedit-url" class="setting" data-setting="url">
        <span class="name">URL</span>
        <input id="imgedit-url" type="text" value="http://142.93.114.99/wp-content/uploads/2019/01/this-is-fine.jpg" readonly="">
    </label>
     ...
</div>
```

References

https://webaim.org/discussion/mail_thread?thread=7458

GUT-64 - Visible labelling only done via placeholder attribute

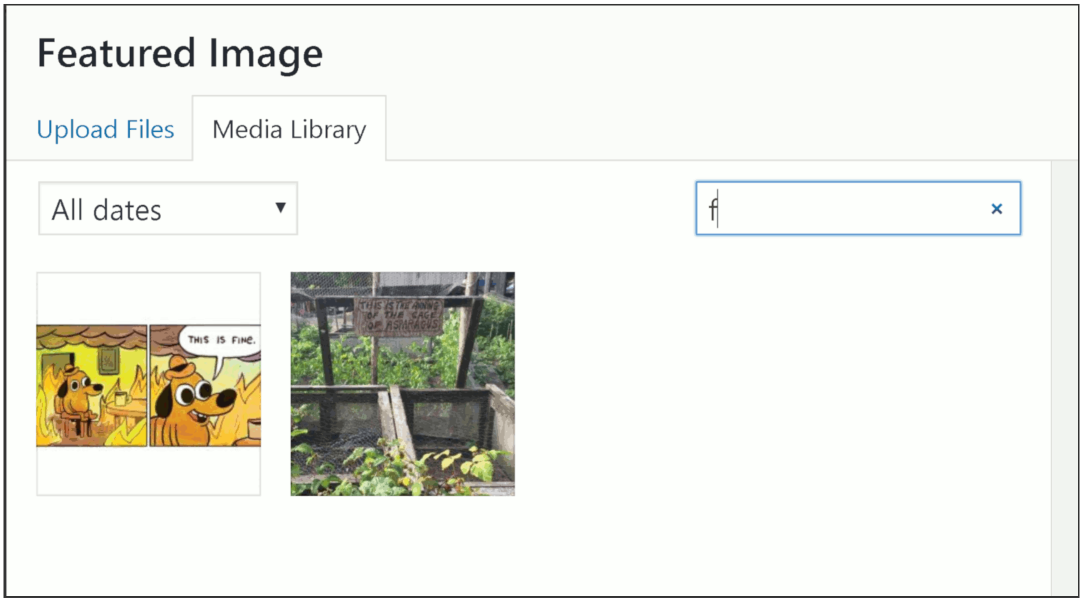
- **Severity:** Low
- **Affected Populations:** Low-Vision, Motor Impaired, Cognitively Impaired
- **Platform(s):** All/ Universal
- **Components affected:** Media Dialog

Issue description

In the "Media Library" panel of the "Featured Image" modal dialog, a live search field is available to filter uploaded images and narrow results.

While there is a label element present, it is rendered visually-hidden, while a placeholder with similar text is visually present in the input.

Users who've entered any values must empty the field to see the placeholder text, or otherwise memorise it. Users with cognitive disabilities may have trouble remembering what the filled field does, while speech recognition users cannot see the name they can speak to set focus on the field. Low-vision users with high text-size may not be able to see the whole placeholder even when it's visible, if its value is clipped by the edge of the input.



Featured Image

Issue Code

```
<div class="media-toolbar-primary search-form">
    <label for="media-search-input" class="screen-reader-text">Search Media</label>
    <input type="search" placeholder="Search media items..." id="media-search-input" class="search">
</div>
...
.screen-reader-text  {
    border: 0;
    clip: rect(1px,1px,1px,1px);
    -webkit-clip-path: inset(50%);
    clip-path: inset(50%);
    height: 1px;
    margin: -1px;
    overflow: hidden;
    padding: 0;
    position: absolute;
    width: 1px;
    word-wrap: normal!important;
}
```

Remediation Guidance

Make labels visible instead of using placeholders, as much as possible.

GUT-60 - Errors and field information are not associated with their control

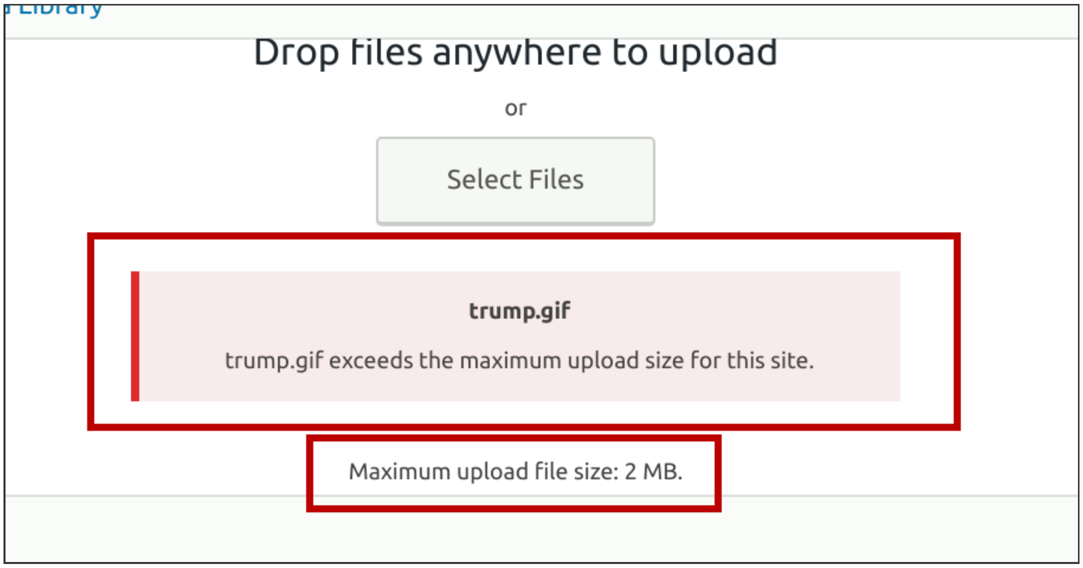
- **Severity:** High
- **Affected Populations:** Blind, Low-Vision, Cognitively Impaired
- **Platform(s):** Windows - Screen Reader, Windows - ZoomText, Mac - VoiceOver, Android - TalkBack, iOS - VoiceOver
- **Components affected:** Media Dialog

Issue description

When users attempt to upload an image in the "Featured Image" modal dialog, and the image is too large, an error message appears visibly. However blind and low-vision users are not alerted with the announcement, and focus remains on the upload button (except in Edge browser, where focus goes back to the top of the page after a failed upload).

The 2MB limit is also not programmatically associated with the "Select Files" button, so blind and low-vision users may miss that limit unless they happen to navigate past the button in exploration of the whole modal.

Users who are zoomed-in or using magnification software may not see the error as it may be outside their viewports. Screen reader users remain on the "Select Files" button and need to navigate around to see if there are any errors or if the upload was successful.



trump.gif

trump.gif exceeds the maximum upload size for this site.

Maximum upload file size: 2 MB.

Issue Code

```
<button type="button" class="browser button button-hero" id="__wp-uploader-id-4" style="display: inline-block;
position: relative; z-index: 1;">Select Files</button>

...

<div class="upload-errors">

    <div class="upload-error">

        <span class="upload-error-filename">trump.gif</span>

        <span class="upload-error-message">trump.gif exceeds the maximum upload size for this site.</span>

    </div>

</div>

...

<div class="post-upload-ui">

    <p class="max-upload-size">Maximum upload file size: 2 MB.</p>

</div>
```

Remediation Guidance

Put `role=alert` on the error, and on any additional error or success messages that are added as users upload media. This will allow screen reader users to know immediately that there is an error.

Consider moving focus to either the error container or the error container close button when the error appears. This will ensure that Braille-only and screen magnification users are alerted that an error has occurred.

Normally, moving focus to errors like this is not advised, however currently the error and the informative text are visually far away from the "Select Files" button, and neither Braille nor screen magnification software announce status messages. A better solution is to change the design to bring the error visually directly under the "Select Files" button (while still also using a live region on the error message).

Programmatically associate the informative text ("max 2MB size") with the "Select Files" button using `aria-labelledby`. Do this by giving the informative text container an `id` value and set this in order in an `aria-labelledby` attribute on the "Select Files" button. Include the button's own `id` token in the `aria-labellebdy` so that the button's text isn't overridden.

Recommended Code

```
<button type="button" class="..." aria-labelledby="__wp-uploader-id-4 post-upload-info" id="__wp-uploader-id-4"
style="...">Select Files</button>

...

<button type="button" class="...">

    <span class="screen-reader-text">Dismiss Errors</span>

</button>

...

<div class="upload-errors">

    <div class="upload-error" role="alert">

        <h3 class="...">trump.gif</h3>

        <p class="...">trump.gif exceeds the maximum upload size for this site.</p>

    </div>

</div>

...

<div class="post-upload-ui" id="post-upload-info">

    <p class="max-upload-size">Maximum upload file size: 2 MB.</p>

</div>
```

Relevant standards

- 1.3.1 Info and Relationships (Level A)
- 1.3.3 Sensory Characteristics (Level A)
- 3.3.1 Error Identification (Level A)
- 3.3.2 Labels or Instructions (Level A)

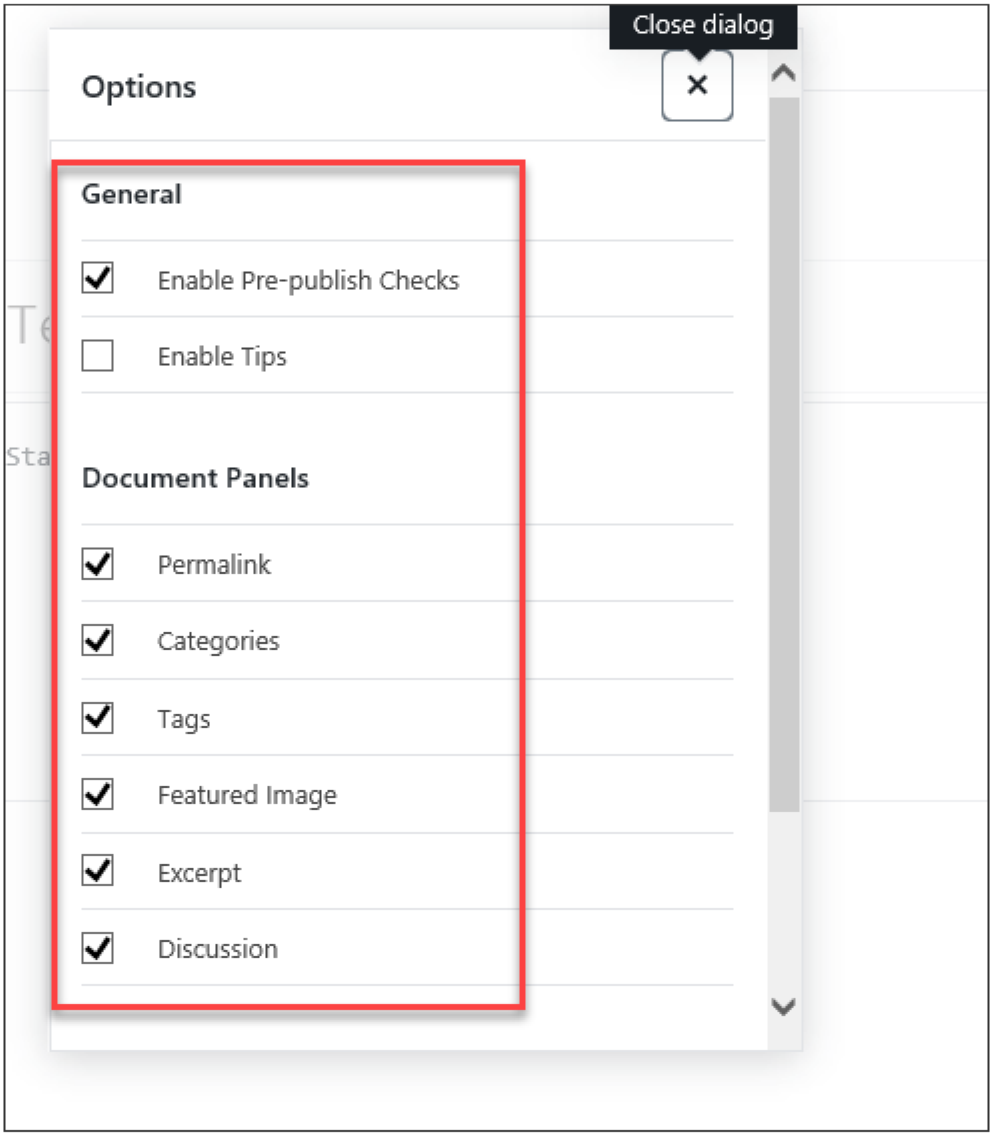
GUT-54 - Form controls are not grouped

- **Severity:** Low
- **Affected Populations:** Blind, Low-Vision, Cognitively Impaired
- **Platform(s):** All/ Universal
- **Components affected:** Tools and Options

Issue description

Within the Options dialog, individual controls are split into sections with preceding heading text, however this grouping is not described using form semantics and therefore not properly conveyed to assistive technologies.

Splitting complex forms into fieldsets helps assistive technology users to understand the grouping and relationships between similar fields.



Options

General

☑ Enable Pre-publish Checks

□ Enable Tips

Document Panels

☑ Permalink

☑ Categories

☑ Tags

☑ Featured Image

☑ Excerpt

☑ Discussion

Issue Code

! 1

```
<!-- General -->
<section class="edit-post-options-modal__section">
   <h2 class="edit-post-options-modal__section-title">General</h2>
   <div class="components-base-control edit-post-options-modal__option">
      <div class="components-base-control__field">
          <input class="components-checkbox-control__input" id="inspector-checkbox-control-0" type="checkbox"
checked="" value="1">
          <label class="components-checkbox-control__label" for="inspector-checkbox-control-0">Enable Pre-publish
Checks</label>
       </div>
   </div>
    ...
</section>

<!-- Document Panel -->
<section class="edit-post-options-modal__section">
   <h2 class="edit-post-options-modal__section-title">Document Panels</h2>
   <div class="components-base-control edit-post-options-modal__option">
      <div class="components-base-control__field">
         <input class="components-checkbox-control__input" id="inspector-checkbox-control-2" type="checkbox"
checked="" value="1">
         <label class="components-checkbox-control__label" for="inspector-checkbox-control-2">Permalink</label>
      </div>
   </div>
</section>

<!-- Advanced Panels -->
<section class="edit-post-options-modal__section">
   <h2 class="edit-post-options-modal__section-title">Advanced Panels</h2>
   <div class="components-base-control edit-post-options-modal__option">
      <div class="components-base-control__field">
         <input class="components-checkbox-control__input" id="inspector-checkbox-control-6" type="checkbox"
value="1">
         <label class="components-checkbox-control__label" for="inspector-checkbox-control-6">Custom
Fields</label>
      </div>
   </div>
</section>
```

Remediation Guidance

Turn the wrapping elements into `<fieldset>` and the group name elements into `<legend>`. The `<legend>` elements themselves can also contain inner `<h2>` elements, in order to continue providing the benefits of a headings structure while still providing a form group structure.

Recommended Code

```
<!-- General -->
<fieldset class="edit-post-options-modal__section">
   <legend><h2 class="edit-post-options-modal__section-title">General</h2></legend>
   <div class="components-base-control edit-post-options-modal__option">
      <div class="components-base-control__field">
         <input class="components-checkbox-control__input" id="inspector-checkbox-control-0" type="checkbox" 
checked="" value="1">
         <label class="components-checkbox-control__label" for="inspector-checkbox-control-0">Enable Pre-publish 
Checks</label>
       </div>
   </div>
   ...
</fieldset>


<!-- Document Panel -->
<fieldset class="edit-post-options-modal__section">
   <legend><h2 class="edit-post-options-modal__section-title">Document Panels</h2></legend>
   <div class="components-base-control edit-post-options-modal__option">
      <div class="components-base-control__field">
         <input class="components-checkbox-control__input" id="inspector-checkbox-control-2" type="checkbox" 
checked="" value="1">
         <label class="components-checkbox-control__label" for="inspector-checkbox-control-2">Permalink</label>
      </div>
   </div>
</fieldset>


<!-- Advanced Panels -->
<fieldset class="edit-post-options-modal__section">
   <legend><h2 class="edit-post-options-modal__section-title">Advanced Panels</h2></legend>
   <div class="components-base-control edit-post-options-modal__option">
      <div class="components-base-control__field">
         <input class="components-checkbox-control__input" id="inspector-checkbox-control-6" type="checkbox" 
value="1">
         <label class="components-checkbox-control__label" for="inspector-checkbox-control-6">Custom 
Fields</label>
      </div>
   </div>
</fieldset>
```

Relevant standards

- 3.3.2 Labels or Instructions (Level A)

GUT-50 - Insufficient form semantics

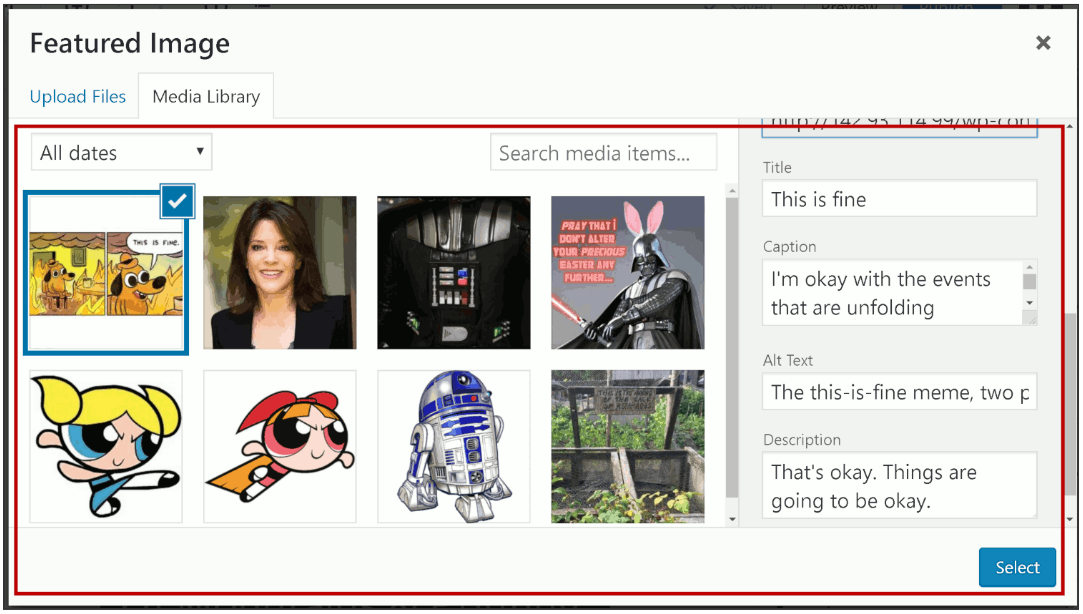
- **Severity:** Low
- **Affected Populations:** Blind, Cognitively Impaired
- **Platform(s):** All/ Universal
- **Components affected:** Media Dialog

Issue description

The "Media Library" panel of the "Featured Image" modal dialog lists some filtering of options, a group of checkboxes, a sub-form of relevant meta-data, and a "Select" button. However there are no semantics which associate these controls together, nor give the orphan "Select" button any meaning relative to the selected checkbox/image.

Additionally, a link taking users to a new window/tab with an editing form does not inform users that it opens a new window or tab.

Semantic markup is vital for assistive technology users, to convey the structure and relationships between information. When appropriate semantic markup is not used, it may be more difficult for such users to understand what the content means, and to comprehend the structure that the visual presentation conveys.



http://142.93.114.99/wp-con

Title: This is fine

Caption: I'm okay with the events that are unfolding

Alt Text: The this-is-fine meme, two p

Description: That's okay. Things are going to be okay.

Issue Code

```
di   l    "     h      b      "
```

```
<div class="attachments-browser">...
    <div class="media-toolbar">
        <div class="media-toolbar-secondary">
            <label for="media-attachment-date-filters" class="screen-reader-text">Filter by date</label>
            <select id="media-attachment-date-filters" class="attachment-filters">...</select>
        </div>
        <div class="media-toolbar-primary search-form">
            <label for="media-search-input" class="screen-reader-text">Search Media</label>
            <input type="search" placeholder="Search media items..." id="media-search-input" class="search">
        </div>
    </div>
    <ul tabindex="-1" class="..."><li ...>...</li></ul>
     ...
    <div tabindex="0" data-id="159" class="attachment-details save-ready">
        <h2>Attachment Details  <span>...</span></h2>
            <div class="attachment-info">
                <div class="thumbnail thumbnail-image">...</div>
                    ...
                    <div class="dimensions">1820 × 1213</div>
                    <a class="edit-attachment" href="...&amp;action=edit..." target="_blank">Edit Image</a>
                    <button type="button" class="...">Delete Permanently</button>
                </div>
            </div>
            <label class="...">
                <span...>URL</span>
                <input type="text" value="...this-is-fine.jpg" readonly="">
            </label>
            ...
        </div>
        <form class="compat-item"></form>
        </div>
    </div>
    <div class="media-toolbar-primary search-form">
        <button type="button" class="...">Select</button>
    </div>
</div>
```

Remediation Guidance

Turn the form into an actual `<form>` element.

Wrap `<fieldset>` tags (with accompanying `<legend>` elements) around groups of controls explaining what they are.

Use the `id` on the `<form>` element with the `form` attribute on the "Select" button to designate this as the form's `submit` button.

Add text to the "Edit" link that opens in a new window or tab, informing users that this will happen. When the form in the new tab/window is submitted, users should be brought back to the original form.

Recommended Code

```
<div class="attachments-browser">...
    <form id="media_form">
    <fieldset class="media-toolbar">
    <legend class="screen-reader">Filter media options</legend>
        <div class="media-toolbar-secondary">
            <label for="media-attachment-date-filters" class="screen-reader-text">Filter by date</label>
            <select id="media-attachment-date-filters" class="attachment-filters">...</select>
        </div>
        <div class="media-toolbar-primary search-form">
            <label for="media-search-input" class="screen-reader-text">Search Media</label>
            <input type="search" placeholder="Search media items..." id="media-search-input" class="search">
        </div>
    </fieldset>
    <fieldset>
        <legend class="screen-reader">Media options result</legend>
        <ul tabindex="-1" class="..."><li ...>...</li></ul>
    </fieldset>
     ...
        <fieldset tabindex="0" data-id="159" class="attachment-details save-ready">
            <legend>Attachment Details  <span>...</span></legend>
            <div class="attachment-info">
                <div class="thumbnail thumbnail-image">...</div>
                    ...
                    <div class="dimensions">1820 × 1213</div>
                    <a class="edit-attachment" href="...&amp;action=edit..." target="_blank">Edit Image <span
class="screen-reader"> Opens in a new window</span></a>
                    <button type="button" class="...">Delete Permanently</button>
                    </div>
                </div>
                <label class="...">
                    <span...>URL</span>
                    <inputtype="text" value="...this-is-fine.jpg" readonly="">
                </label>
                ...
            </div>
        </fieldset>
    </form>
    <div class="media-toolbar-primary search-form">
        <button form="media_form" type="button" class="...">Select</button>
    </div>
</div>
```

Relevant standards

- 1.3.1 Info and Relationships (Level A)
- 1.3.3 Sensory Characteristics (Level A)
- 3.3.2 Labels or Instructions (Level A)
- 4.1.2 Name, Role, Value (Level A)

Related Issues

- Related to "Feature Image dialog does not follow the dialog pattern"
- Related to "Insufficient labelling and inappropriate nesting of form controls"

GUT-39 - Instructions are unclear

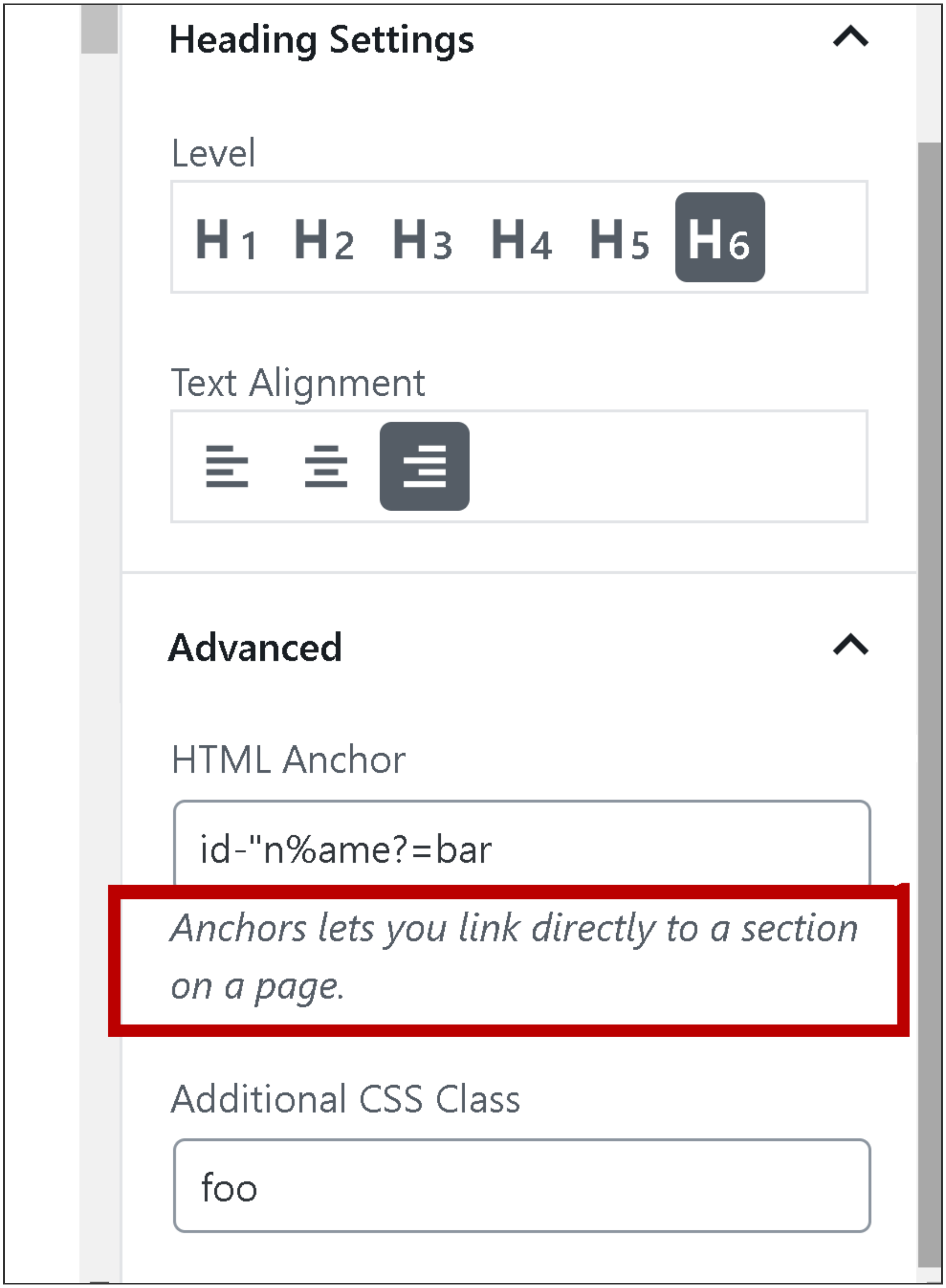
- **Severity:** Low
- **Affected Populations:** Cognitively Impaired
- **Platform(s):** All/ Universal
- **Components affected:** Block Panel

Issue description

Users going through the Block panel after selecting a Heading type block, will come to a field asking for an "HTML anchor", with the explanation: "Anchors lets you link directly to a section on a page."

While it may be too technical to explain that this field adds an `id` attribute to the heading, the brief explanation leads users to be unsure if the field wants an actual anchor tag or a string, and what the rules are for its value.

Users with cognitive, reading or writing disabilities may also have trouble with the input's automatic changing of space and other characters as they type, without explanation.



Heading Settings

Level

Text Alignment

Advanced

HTML Anchor

id-"n%ame?=bar

Anchors lets you link directly to a section on a page.

Additional CSS Class

foo

Issue Code

```
<div class="components-base-control__field">
    <label class="..." for="inspector-text-control-4">HTML Anchor</label>
    <input class="..." type="text" id="inspector-text-control-4" aria-describedby="inspector-text-control-4__help" value="...">
</div>
<p id="inspector-text-control-4__help" class="...">Anchors lets you link directly to a section on a page.</p>
```

Remediation Guidance

Change "anchor" to something that doesn't suggest the user should actually add an anchor. This also avoids web-developer jargon: more users know what a "link" is than an "anchor".

Have the explanatory text mention the sole restriction of id-token strings (i.e. they cannot contain spaces) to help explain why spaces are being automatically changed to other characters.

Consider showing an example of resulting HTML code so that users can see what changes as they change their string. Users who don't understand HTML code are still free to ignore this.

Recommended Code

```
<div class="components-base-control__field">
    <label class="..." for="inspector-text-control-4">Add an-page link</label>
    <input class="..." type="text" id="inspector-text-control-4" aria-describedby="inspector-text-control-4__help" value="...">
</div>
<p id="inspector-text-control-4__help" class="...">Add a unique name (without spaces) to your heading to allow linking directly to that section on the page.</p>

<div class="...">
    <label class="..." for="inspector-text-4__help">Example code result</label>
    <input class="..." type="text" id="inspector-text-4__help" readonly="" value="[heading with id here]">
</div>
```

Relevant standards

- 3.3.2 Labels or Instructions (Level A)

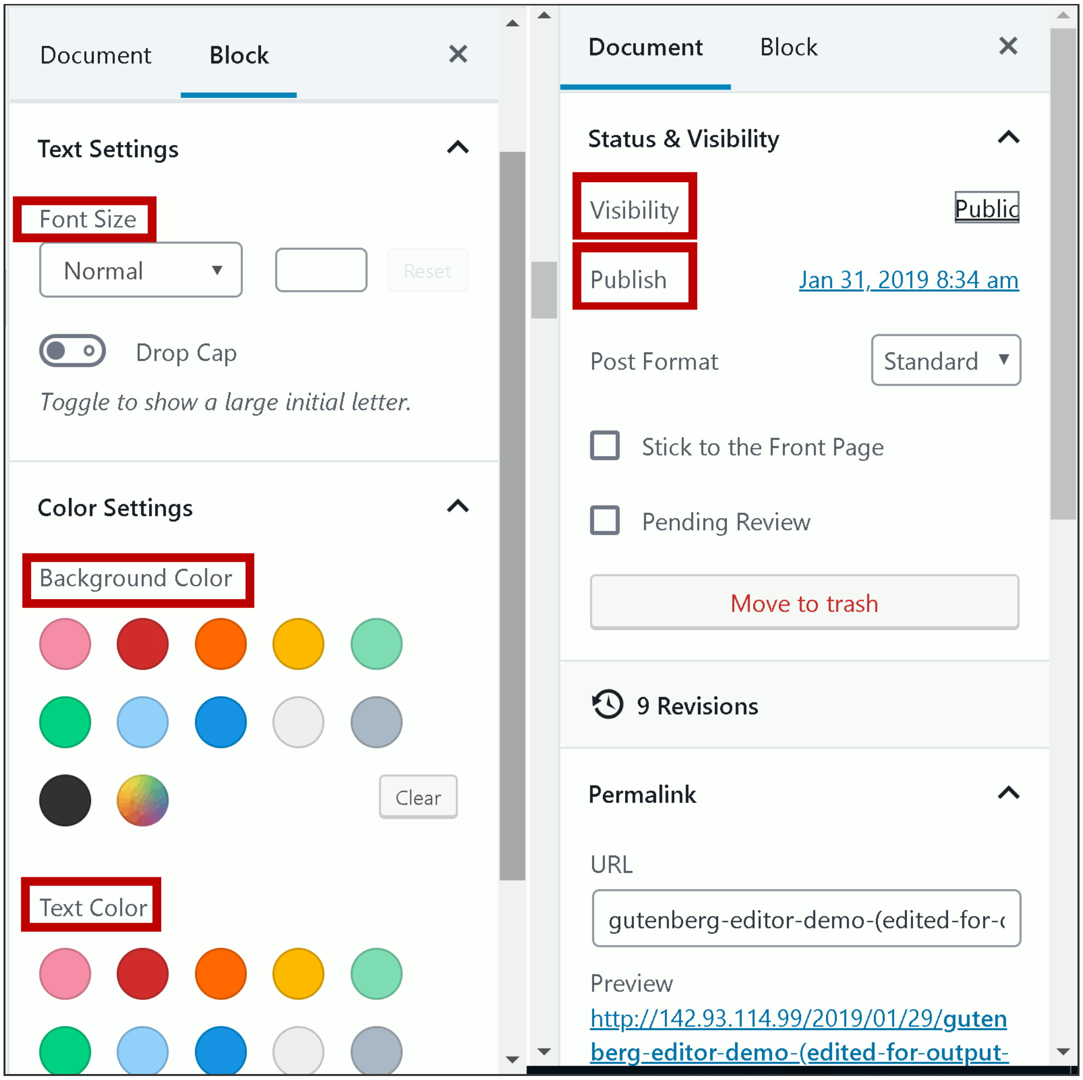
GUT-36 - Form controls are neither grouped nor named

- **Severity:** Medium
- **Affected Populations:** Blind, Cognitively Impaired
- **Platform(s):** All/ Universal
- **Components affected:** Block Panel, Document Panel

Issue description

In the Settings dropdown (both Document and Block panels), loose text is visually placed next to groups of controls, making the controls' context visually clear, but not programmatically clear.

Additionally, one specific button (the "Public" button which is styled as a link) has two `<label>` elements attached to it, the second overriding the first and giving non-visual users less context.



Font Size: Normal ▾

○ Drop Cap

Toggle to show a large initial letter.

Background Color

Text Color

Visibility: Public

Publish: Jan 31, 2019 8:34 am

Post Format: Standard ▾

□ Stick to the Front Page

□ Pending Review

URL: gutenberg-editor-demo-(edited-for-(

Preview: http://142.93.114.99/2019/01/29/gutenberg-editor-demo-(edited-for-output-

Issue Code

```
/* Publish options */

<div class="components-panel__row edit-post-post-visibility">

   <span>Visibility</span>

<div>

<button type="button" aria-expanded="false" class="components-button edit-post-post-visibility__toggle is-
link">Public</button>

...

<div class="components-panel__row edit-post-post-schedule">

    <label for="edit-post-post-schedule__toggle-13" id="edit-post-post-schedule__heading-13">Publish</label>

    <div>

        <label class="edit-post-post-schedule__label" for="edit-post-post-schedule__toggle-13">Jan 31, 2019 8:34
am Click to change</label>

        <button type="button" id="edit-post-post-schedule__toggle-13" aria-expanded="false" aria-live="polite"
class="components-button edit-post-post-schedule__toggle is-link">Jan 31, 2019 8:34 am</button>

    </div>

</div>

...

/* example section areas */

<div class="components-base-control editor-color-palette-control">

    <div class="components-base-control__field">

        <span class="components-base-control__label">Background Color</span>

        ...

        <button type="button" aria-label="Color: Pale pink" ...></button>

        ...

        <button type="button" class="...">Clear</button>

    </div>

    ...

</div>

...

<div class="block-library-image__dimensions">

    <p class="block-library-image__dimensions__row">Image Dimensions</p>

    <div class="...">

        <label ...>Width</label>

        <input type="number" ...>

        ...

        <label ...>Height</label>

        <input type="number" ...>

        ...

        <div aria-label="Image Size" class="components-button-group" role="group">

            <button type="button" ...">25%</button>

            <button type="button" ...>50%</button>

            <button type="button" ...>75%</button>

            <button type="button" ...>100%</button>

        </div>

        <button type="button" class="...">Reset</button>
    </div>
</div>
```

Remediation Guidance

For most of the selection areas, turning the wrapping element into a `fieldset` and the element with the group name text to a `legend` is sufficient.

For the "Visibility" and "Publish" sections, the Recommended Code shows turning the visible text elements near the buttons into headings (level `h3`) and adding additional context to the publish date button as hidden screen reader text.

Another option available is to add`id` attributes to the `<span>` and `<button>` elements, then referencing both of them to label the button using `aria-labelledby`.

Recommended Code

```
/* Publish options */

<div class="components-panel__row edit-post-post-visibility">

   <h3>Visibility</h3>

<div>

<button type="button" aria-expanded="false" class="components-button edit-post-post-visibility__toggle is-link">Public</button>

...

<div class="components-panel__row edit-post-post-schedule">

    <h3 id="edit-post-post-schedule__heading-13">Publish</h3>

    <div>

        <button type="button" id="edit-post-post-schedule__toggle-13" aria-expanded="false" ...">Jan 31, 2019 8:34 am

            <span class="screen-reader-text">Change date</span>

        </button>

    </div>

</div>

...

/* example section areas */

<div class="components-base-control editor-color-palette-control">

    <fieldset class="components-base-control__field">

        <legend class="components-base-control__label">Background Color</legend>

        ...<button type="button" aria-label="Color: Pale pink" ...></button>

        ...

        <button type="button" class="...">Clear</button>

    </fieldset>

    ...

</div>

...

<fieldset class="block-library-image__dimensions">

    <legend class="block-library-image__dimensions__row">Image Dimensions</legend>

    <div class="...">

        <label ...>Width</label>

        <input type="number" ...>

        ...

        <label ...>Height</label>

        <input type="number" ...>

        ...

        <div aria-label="Image Size" class="components-button-group" role="group">

            <button type="button" ...">25%</button>

            <button type="button" ...>50%</button>

            <button type="button" ...>75%</button>

            <button type="button" ...>100%</button>

        </div>

        <button type="button" class="...">Reset</button>
    </div>
</fieldset>
```

Relevant standards

- 1.3.1 Info and Relationships (Level A)
- 3.3.2 Labels or Instructions (Level A)
- 4.1.2 Name, Role, Value (Level A)

Content Type: Frames

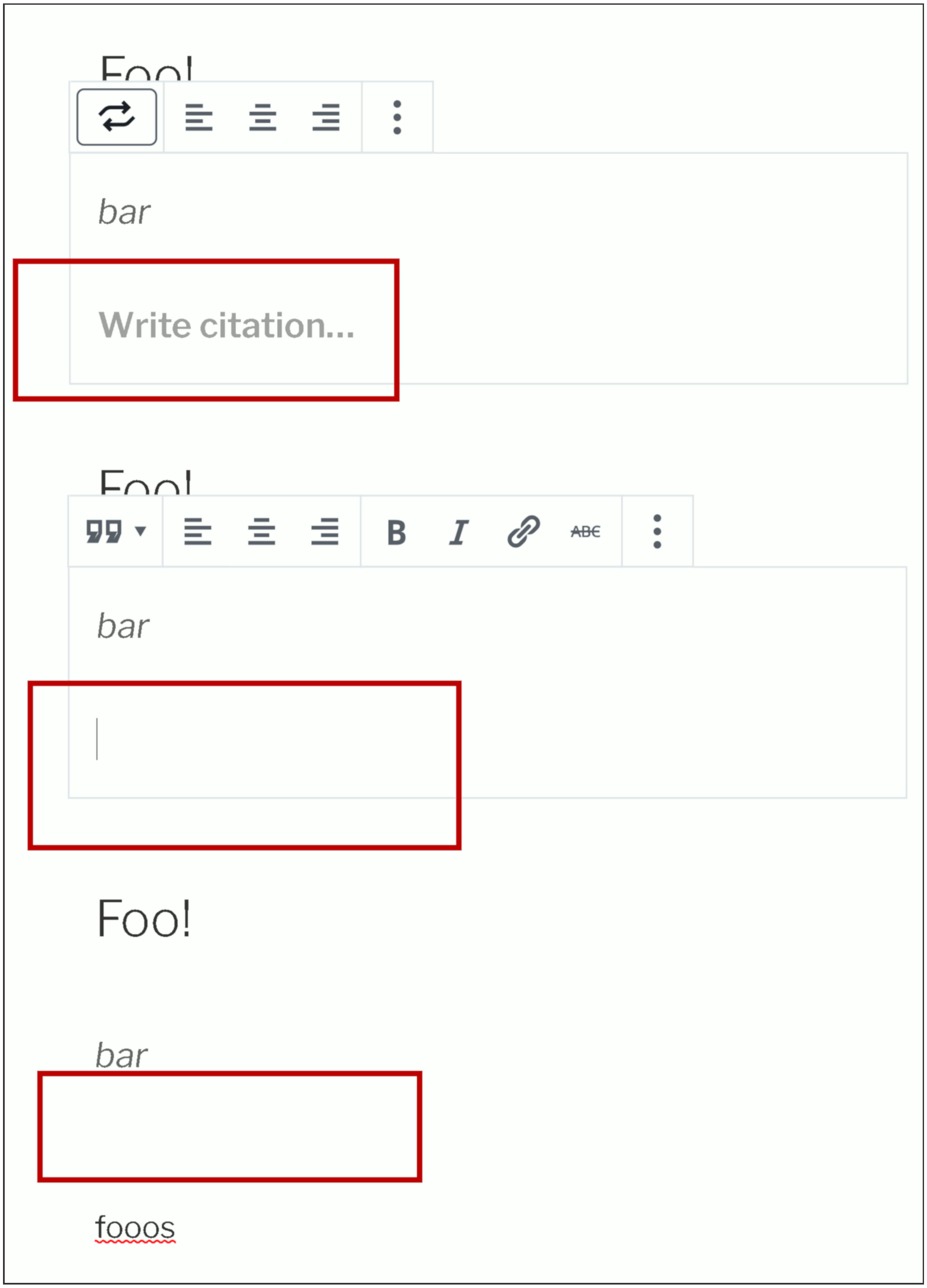
GUT-104 - Citation field is only indicated by placeholder text

- **Severity:** Low
- **Affected Populations:** Low-Vision, Cognitively Impaired
- **Platform(s):** All/ Universal
- **Components affected:** Block Editing

Issue description

The Quote block type includes an editable area to define a citation. However this area is only visually indicated with placeholder text, it has no visible boundary. The placeholder text also vanishes when focus leaves the block, and when focus enters the control. There is no other visual cue that a field exists there.

Active user interface components should be easily distinguishable by people with moderately low-vision, and consistently available so long as the control can be activated, for the benefit of users with short-term memory loss and other cognitive disabilities.



Foo!

bar

Write citation…

Foo!

bar

Foo!

bar

fooos

Remediation Guidance

So long as the citation field can be filled in, use a design that makes it visually obvious that it's an editable field.

Content Type: Images and Non-text Content

GUT-108 - Users are able to directly link images

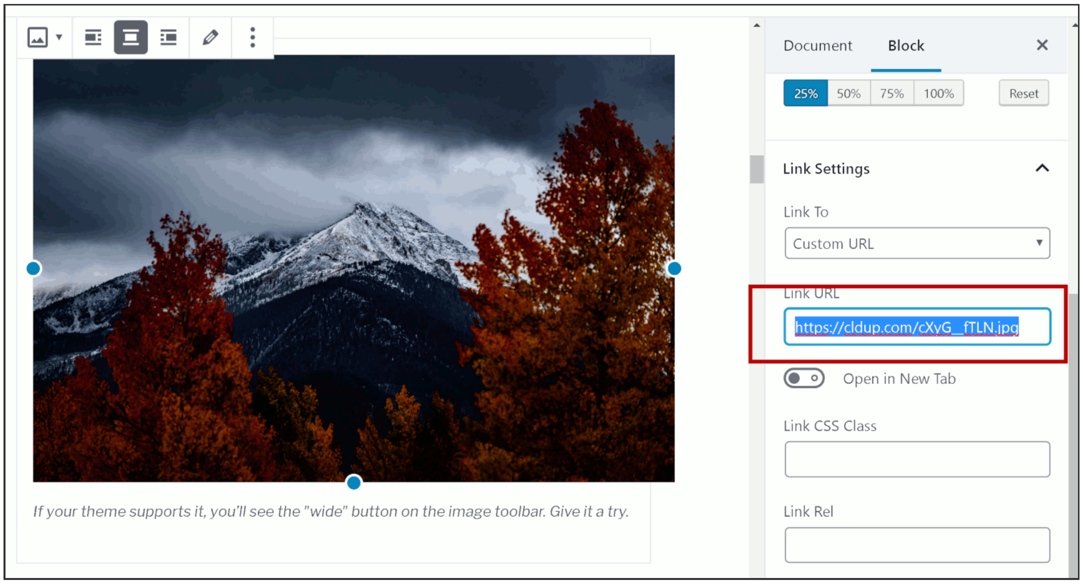
- **Severity:** Medium
- **Affected Populations:** Blind, Low-Vision, Cognitively Impaired
- **Platform(s):** All/ Universal
- **Components affected:** Block Editing

Issue description

Authors adding image media to posts have the option to link to that image directly. There is a section to write alt text for the image, but nothing for link text.

Links whose sole link text is the alternate text of an image do not adequately state where the link will go to, and this can be confusing for screen reader users and users with cognitive disabilities. Linking directly to images is also inadvisable, because the resulting image view in the browser does not include alternate text.

Therefore the editor should warn users about these issue.



If your theme supports it, you'll see the "wide" button on the image toolbar. Give it a try.

Link Settings

Link To: Custom URL

Link URL: https://cldup.com/cXyG__fTLN.jpg

○ Open in New Tab

Link CSS Class:

Link Rel:

Remediation Guidance

The Block panel should include a warning about linking directly to images, similar to the way that poor combinations of text and background contrast are conveyed.

Additionally, images used to link to other resources should offer a field in the Block panel to add dedicated link text separate from the image's own description.

Relevant standards

- 2.4.4 Link Purpose (In Context) (Level A)

GUT-84 - Undo and Redo icons are not consistent throughout the interface

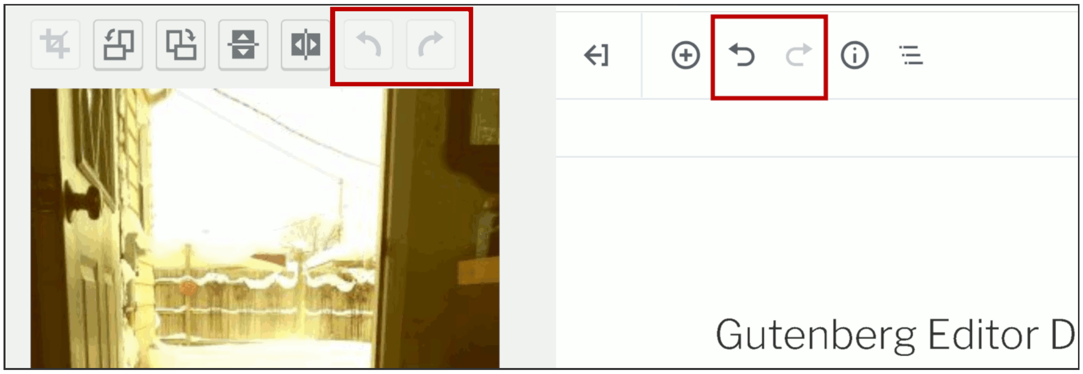
- **Severity:** Low
- **Affected Populations:** Cognitively Impaired
- **Platform(s):** All/ Universal
- **Components affected:** Edit Media

Issue description

On the Edit Media page, there are various icon buttons for performing actions to the selected media image, along with "Undo" and "Redo" buttons.

These two buttons use different icons than the "Undo" and "Redo" buttons in the Editor Top Bar.

Ideally these buttons' icons would be consistent with the ones in the Editor Top Bar, especially since these icons look like Rotate icons (causing the actual Rotate buttons to require more complex glyphs to appear more clearly as Rotate-the-image), while the Editor Top Bar icons do not look like Rotate buttons.



Remediation Guidance

Use the same icons for the "Undo" and "Redo" buttons as are used in the Editor Top Bar.

Relevant standards

- 3.2.4 Consistent Identification (Level AA)

GUT-66 - Icons alone are visible for controls unless they are hovered or focussed

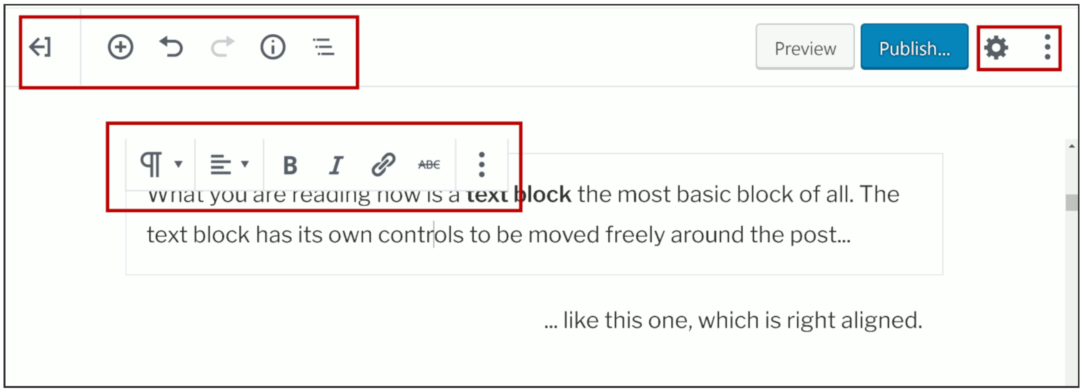
- **Severity:** Low
- **Affected Populations:** Motor Impaired, Cognitively Impaired
- **Platform(s):** All/ Universal
- **Components affected:** Global

Issue description

Throughout the editor, large icons denote the meaning of controls visually. The buttons have programmatically accessible names, and these names (and their keyboard shortcuts if any) can be made visible by hovering with the mouse or moving keyboard focus to them.

However hovering is an action that cannot be performed on most touch devices, and moving keyboard focus generally requires a keyboard (or something imitating it such as Switch Control). This leaves speech recognition users (who rely on knowing the name of a control for the easiest activation) in the position of needing to manually get focus to the page and then using "press tab" to move focus to each control to see (and memorise) the name for later use, or using the more cumbersome Mouse Grid to slowly move the mouse to the desired control.

Touchscreen users with cognitive disabilities have no easy way to learn what controls are going to do before they interact with them.



What you are reading now is a **text block** the most basic block of all. The text block has its own controls to be moved freely around the post...

... like this one, which is right aligned.

Remediation Guidance

One solution would be to provide users with a setting, which shows labels in place of (or as well as) icons, similar to Gmail's settings. An example of this can be see at: https://allenpike.com/images/2019/lukew-translate.jpg

This setting would allow users who prefer icons-only to continue to work with icons only, while those who prefer or require text labels would have that option too.

Content Type: Keyboard Accessibility and Focus Control

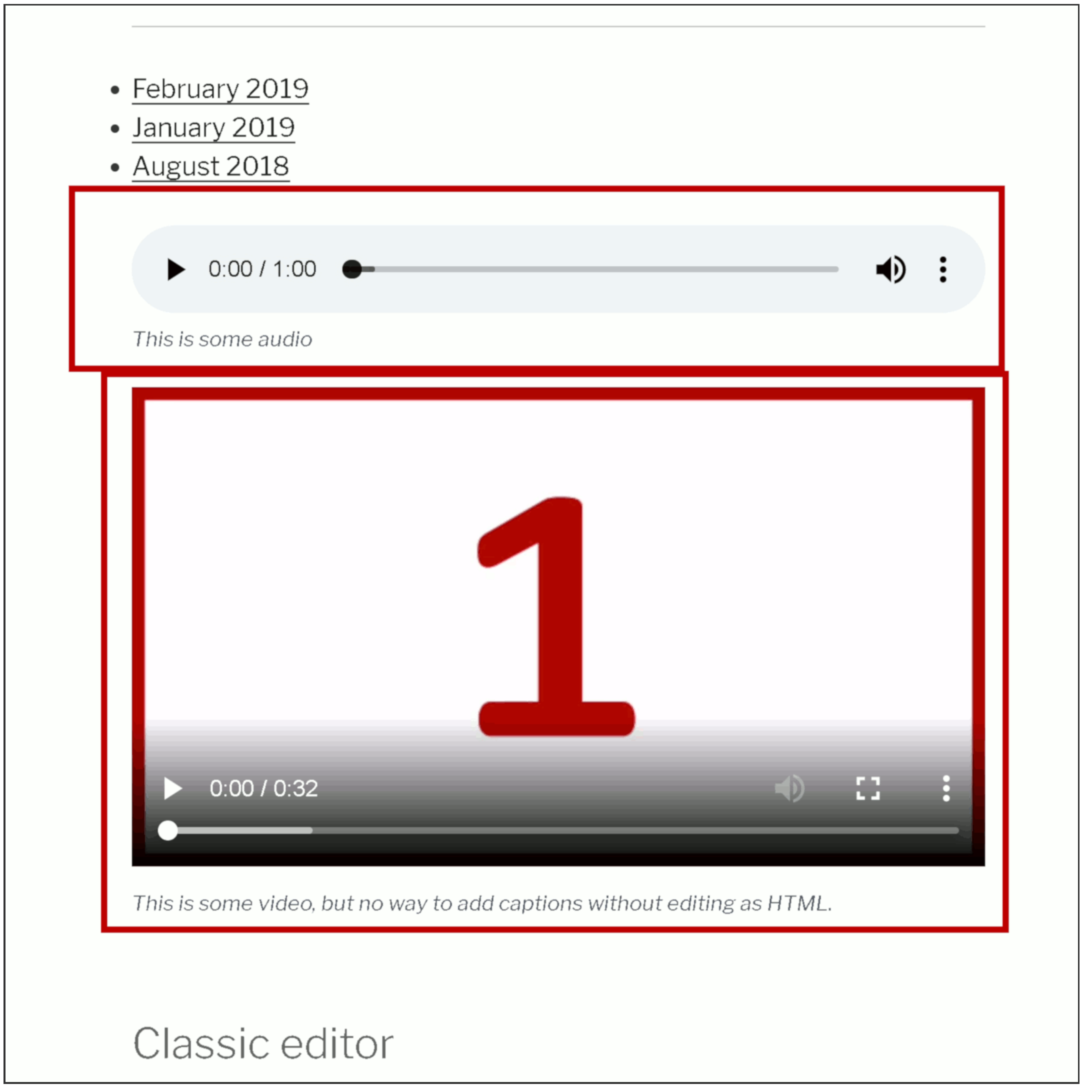
GUT-106 - Media players receive focus but show no visible focus state

- **Severity:** High
- **Affected Populations:** Low-Vision, Motor Impaired, Cognitively Impaired
- **Platform(s):** All/ Universal
- **Components affected:** Output Markup

Issue description

Keyboard users who are navigating through a published (or previewed) page may come across media players implemented with native audio and video elements. These receive keyboard focus in most or all browsers, but the page does not visually show it. This results in invisible Tab stops.

All interactive elements should have a visible and obvious focus indicator, so that sighted keyboard users have a clear indication of which element has focus.



- February 2019
- January 2019
- August 2018

This is some audio

This is some video, but no way to add captions without editing as HTML.

Classic editor

Issue Code

```
:focus {
    outline: none;
}
```

Remediation Guidance

Whenever the native focus `outline` is removed from focusable elements, a replacement visual focus state must be provided, with a minimum contrast ratio of 3:1.

Recommended Code

```
audio:focus, video:focus {
    outline: 1px solid #0085ba;
}
```

Relevant standards

- 2.1.1 Keyboard (Level A)
- 2.4.7 Focus Visible (Level AA)

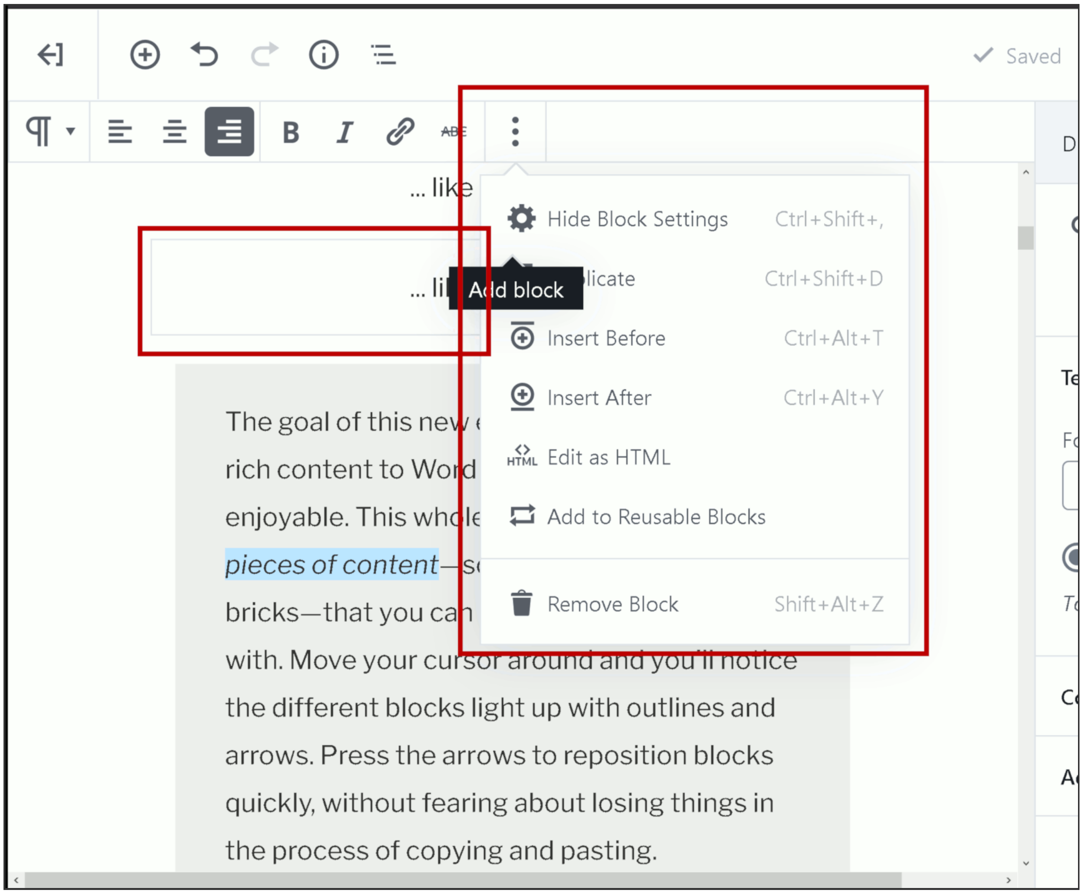
GUT-101 - Menu remains open after actions which moves focus

- **Severity:** Medium
- **Affected Populations:** Low-Vision, Motor Impaired, Cognitively Impaired
- **Platform(s):** All/ Universal
- **Components affected:** Block Options

Issue description

When the Unified Toolbar is being used, the Options menu remains open after triggering an action that causes the keyboard focus to be moved to the edit area. This results in a situation where the focus position might be underneath the menu. The actions of duplicating a block, inserting new content before or after a block, or removing a block, all have this behaviour.

Obscuring the focus position is potentially confusing for sighted keyboard users, since it may appear as though no item has focus, or make it unclear as to where the focus position is. Additionally it is simply difficult to view the content and moving keyboard focus to and from the toolbar is cumbersome.



The goal of this new e
rich content to Word
enjoyable. This whole
pieces of content—s
bricks—that you can
with. Move your cursor around and you'll notice
the different blocks light up with outlines and
arrows. Press the arrows to reposition blocks
quickly, without fearing about losing things in
the process of copying and pasting.

Remediation Guidance

Whenever an action in the Options menu causes the focus to move, the menu should close automatically.

Relevant standards

- 2.1.1 Keyboard (Level A)
- 2.4.7 Focus Visible (Level AA)

GUT-95 - Keyboard operation is confusing and inconsistent

- **Severity:** Medium
- **Affected Populations:** Blind, Low-Vision, Motor Impaired, Cognitively Impaired
- **Platform(s):** All/ Universal
- **Components affected:** Block Editing

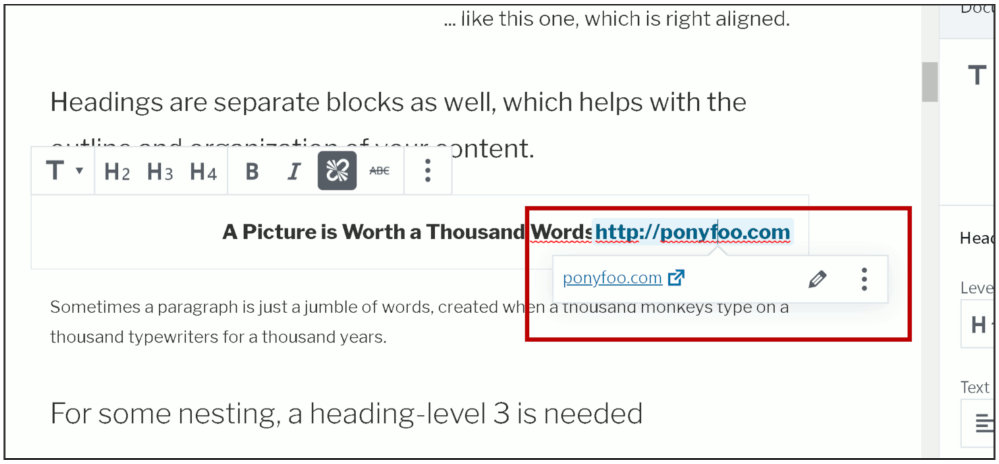
Issue description

Keyboard navigation between Blocks, Block controls, and the Block panel, is inconsistent and difficult to use.

In the default setup (no Unified Toolbar), getting controls to appear generally requires moving keyboard focus with the Tab key. Moving between Blocks is meant to be done by moving the cursor with arrow keys. However:

- Some Blocks seem to trap the cursor, specifically the Cover type, however some Paragraph Blocks from the Gutenberg demo do as well (for unknown reasons; creating Paragraph type duplicates of these Blocks does not reproduce the trapping, so it's not clear why this occurs). In order to escape these Blocks, users must use the Tab key, which then adds the additional Tab stops of the toolbars which appear.
- When attempting to copy text within a Block using Shift and Arrow keys, the last line (or first line if starting from the bottom and going back) can't be selected without also beginning to select multiple Blocks. This might be deliberate (to encourage users who are attempting to select all of a Block's content to instead use the dedicated duplicate function) but users should expect to be able to select any text they wish. Additionally, it's common in a multi-line Block to accidentally move the cursor to a previous Block when merely trying to get the cursor moved to the first line (and accidentally move to a later Block while moving the cursor to the last line), especially if a screen reader is running.
- Users who are not using zoom must Shift+Tab backwards from the text to reach the toolbar, which matches the visual order when the toolbar is above the text. However users who are zoomed-in see the toolbar **under** the text, yet still have to Shift+Tab backwards to reach it (see: GUT-99 Done).

- While interacting with the Block's toolbar, focus sometimes ends up at the top of the document. This seems to be intermittent and unintended, but every time it happens it is frustrating.
- After a link is created, moving focus to it again offers an editing popup, however this is not keyboard accessible. It appears to be the only place where users can set whether the link opens in a new tab/window, and therefore this functionality is not available to keyboard users.



... like this one, which is right aligned.

Headings are separate blocks as well, which helps with the outline and organization of your content.

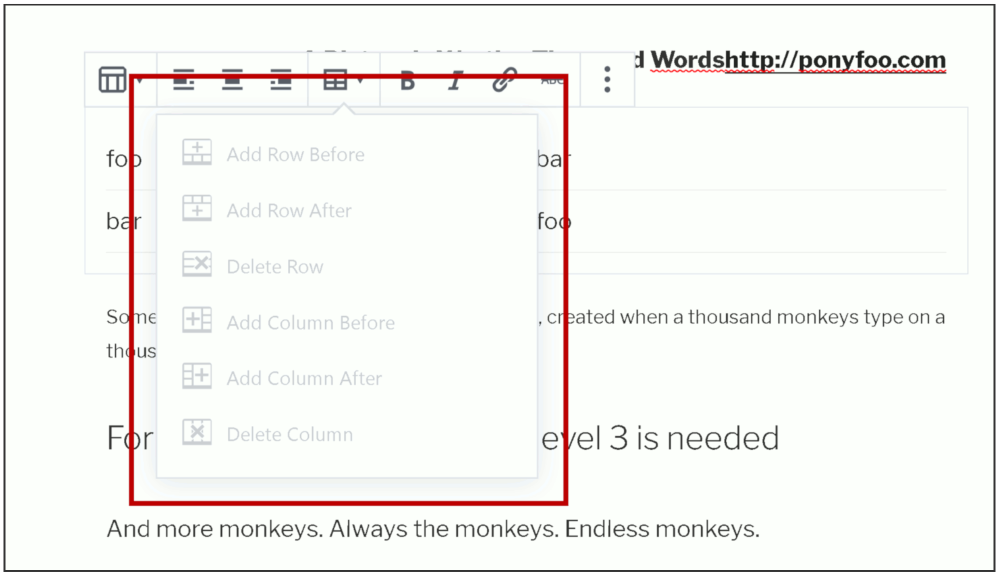
A Picture is Worth a Thousand Words http://ponyfoo.com

ponyfoo.com

Sometimes a paragraph is just a jumble of words, created when a thousand monkeys type on a thousand typewriters for a thousand years.

For some nesting, a heading-level 3 is needed

- If a Table is being created and the "Edit Table" popup is opened, users may attempt to use their Arrow keys to navigate it, however this unexpectedly moves the user focus to the top or bottom of the whole editable area. The only interaction users can safely do is to immediately press Esc to close the dropdown. This happens when the dropdown opens and all of the options are disabled.



In the Unified Toolbar setup, only the "New Block" and move controls are in the Tab order, and moving between Blocks is still done by moving the cursor with Arrow keys. Users can use the keyboard shortcut `Alt+F10` to jump focus to the Unified Toolbar (and can then Tab past the Editor Top Bar buttons to the Block toolbar), however there is no way to jump back from the panel to the Block in question.

In both settings, there is no non-exhaustive way of going from a Block to the Block Settings panel, and if somehow the chosen Block is forgotten (i.e. not seen as selected by the editor), then the Block panel goes blank, and therefore the button that bring users back to the Block in question is also lost.

Remediation Guidance

Add some kind of Bypass Block mechanism for keyboard user to move from a selected block to the Block Panel, since this appears to be the only place where font settings, drop cap and colors can be set.

Add a similar Bypass Block mechanism for users who've jumped to the Unified Toolbar, to get back to the selected block without needing to go through the Block panel's skip link.

If opening a popup cannot allow JavaScript to move focus to the first item in the popup automatically (for example, because the popup has no focusable items inside), then the Arrow keys should either move the cursor up or down from the toolbar button (thereby closing the popup), or keep focus there until users close the popup with the Esc key.

In any case, do not allow Arrow keys to move the cursor to the beginning or end of the document; this is the last thing users would expect. Ideally, if there are no available options in a dropdown, perhaps the activating button should have the dropdown replace the options with a message such as "No options currently available."

Wait for all content inside a block to receive selection before offering multiple-block selection.

Allow cursor movement to always be able to move between blocks regardless of type (unless there was a good reason for this; if so, explaining the reason to users would then be helpful).

Ensure popups opened from within blocks can be reached by keyboard.

Relevant standards

- 2.1.1 Keyboard (Level A)
- 2.4.1 Bypass Blocks (Level A)
- 2.4.3 Focus Order (Level A)

GUT-82 - Reading and focus order do not match visual order

- **Severity:** Medium
- **Affected Populations:** Blind, Low-Vision, Cognitively Impaired
- **Platform(s):** All/ Universal
- **Components affected:** Edit Media

Issue description

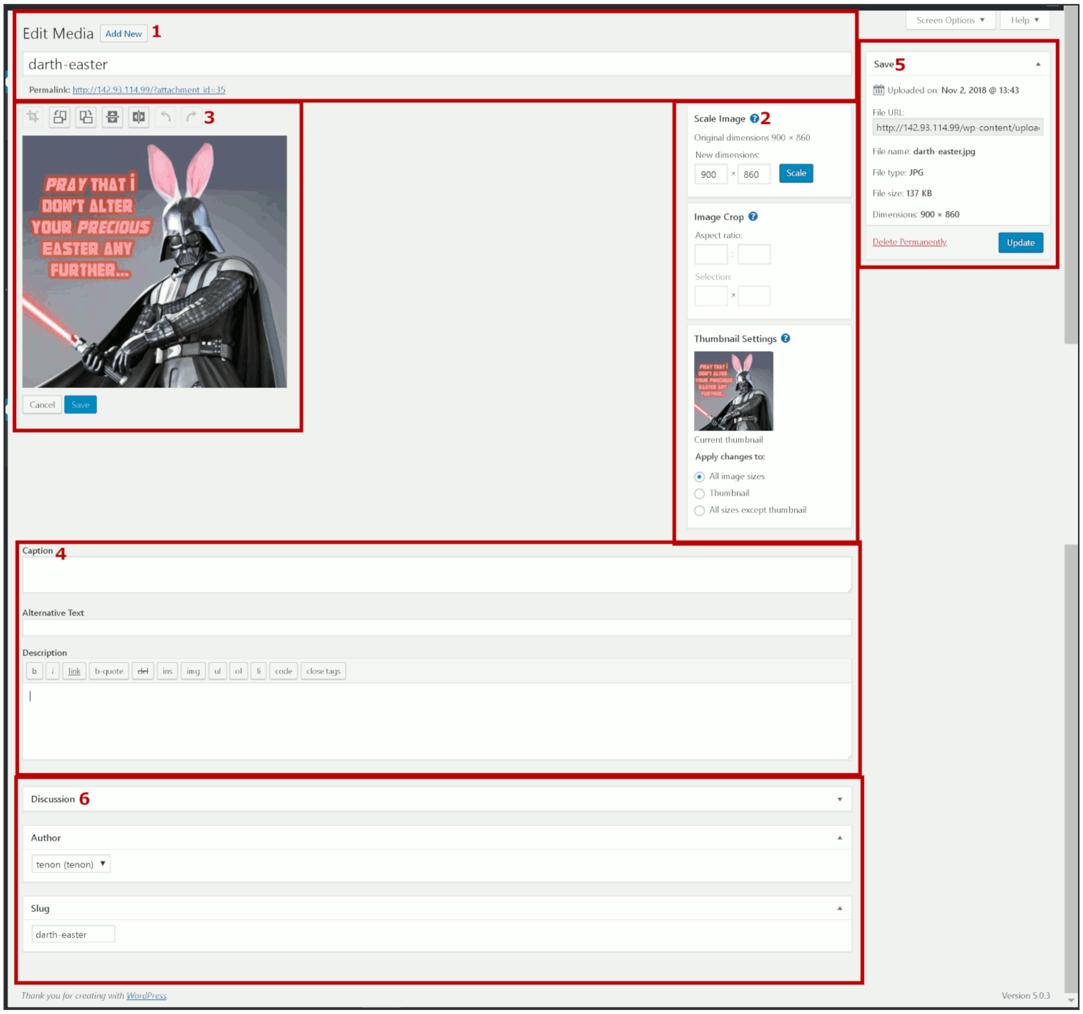
On the Edit Media page, the keyboard focus order and the visual reading order are presented in a zig-zag pattern.

Users who are not using browser zoom can reach an option to show more or fewer columns:

1. If users select content to appear in a single column (which still produces two columns visually), content at the top is first, then comes content on the right (Scale and Crop sections), then content back to the left (the actual image and editing buttons), and finally content underneath both columns (metadata).

2. If users select content to appear in two columns (which produces three columns visually), content at the top is first, then comes a column in the middle ("Scale" and "Crop" sections), then the column in the left (the actual image and editing buttons), and then the content at the bottom (the default metadata). After some of the content at the bottom, the "Save" section is on the far right, and then if the user added more boxes, these are last and visually placed after the default metadata content.

Keyboard users benefit from a logical focus order which is consistent with the reading order. Users who have a cognitive disability and have difficulty reading may also be confused by an unexpected or illogical focus order.



Edit Media

darth-easter

Permalink: http://142.93.114.99/?attachment_id=35

Scale Image

Original dimensions 900 × 860

New dimensions: 900 × 860

Image Crop

Aspect ratio: :

Selection: ×

Thumbnail Settings

Current thumbnail

Apply changes to:

● All image sizes

○ Thumbnail

○ All sizes except thumbnail

Save

Uploaded on: **Nov 2, 2018 @ 13:43**

File URL: http://142.93.114.99/wp-content/uploa

File name: **darth-easter.jpg**

File type: **JPG**

File size: **137 KB**

Dimensions: **900 × 860**

Caption

Alternative Text

Description

Discussion ▾

Author ▴

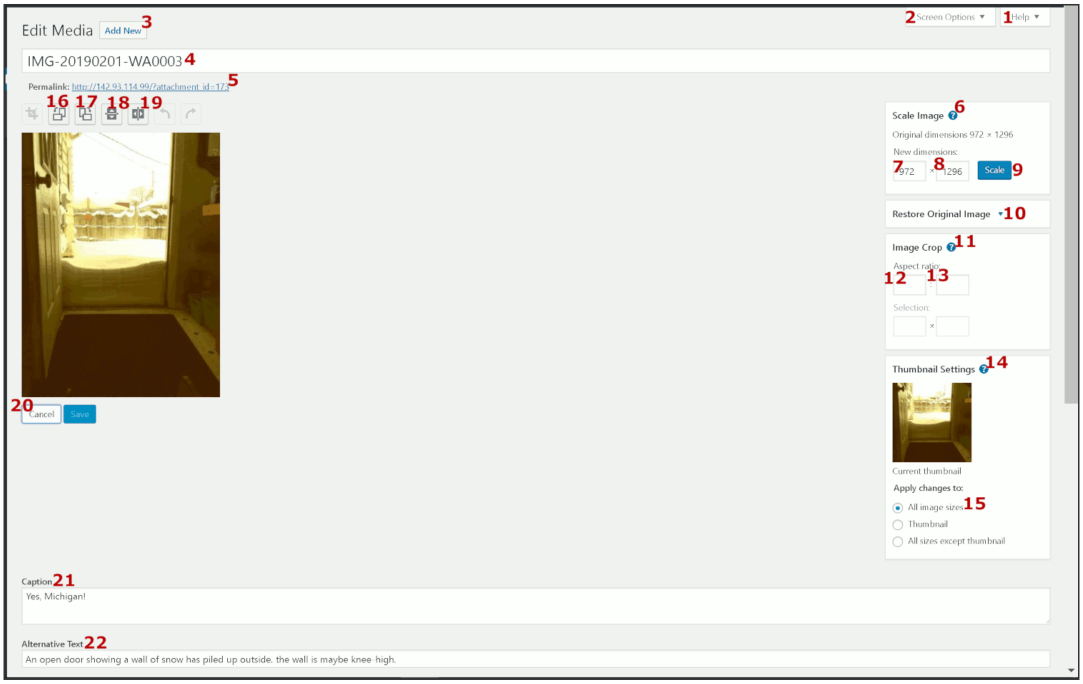
tenon (tenon) ▾

Slug ▴

darth-easter

Thank you for creating with WordPress.

Version 5.0.3



IMG-20190201-WA0003

Permalink: http://142.93.114.99/?attachment_id=173

Original dimensions 972 × 1296

New dimensions: 972 × 1296

Caption: Yes, Michigan!

Alternative Text: An open door showing a wall of snow has piled up outside. the wall is maybe knee-high.

Remediation Guidance

Either change the visible layout with CSS to make the visual order match the content order, or re-order the content in the DOM to match the visual order.

Relevant standards

- 1.3.2 Meaningful Sequence (Level A)
- 2.4.3 Focus Order (Level A)

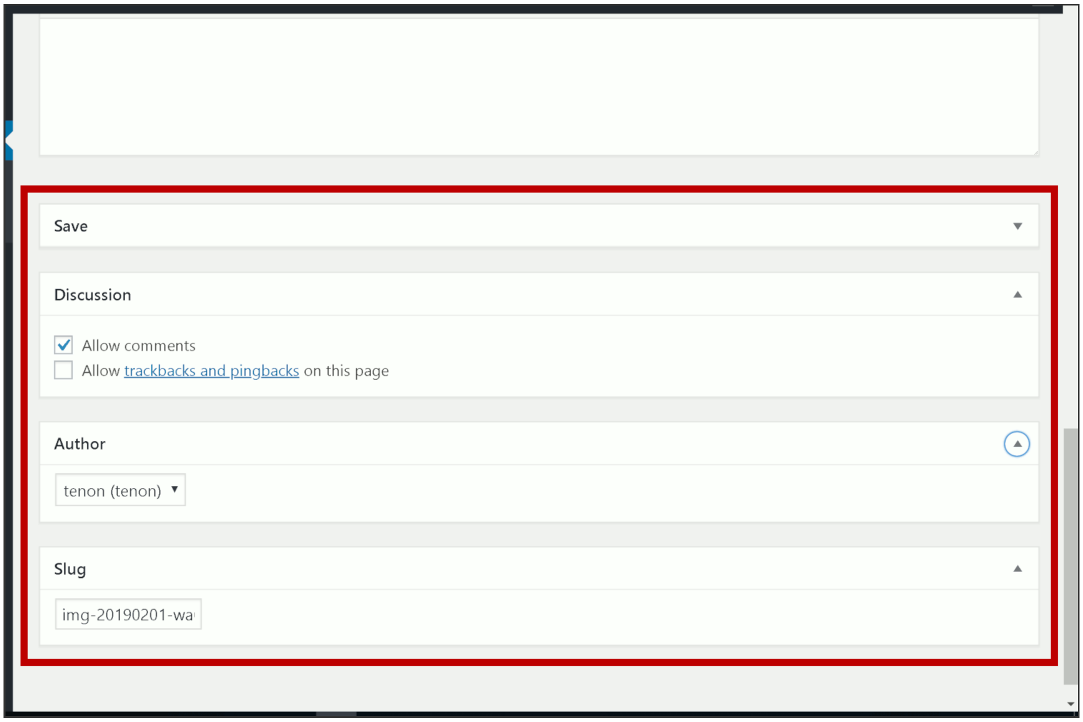
GUT-80 - Drag and drop functionality unavailable to keyboard or touchscreen

- **Severity:** Low
- **Affected Populations:** Blind, Motor Impaired
- **Platform(s):** All/ Universal
- **Components affected:** Edit Media

Issue description

The Edit Media interface includes the ability to re-arrange the boxes by drag and drop, however this can only be achieved by mouse users, since the functionality does not support touch or keyboard users.

Users of touchscreens and keyboards cannot re-order the boxes. Speech recognition users may be able to if they can manipulate the mouse with speech (such as using Mouse Move), however this is onerous and difficult.



☑ Allow comments
□ Allow trackbacks and pingbacks on this page

Author: tenon (tenon)

Slug: img-20190201-wa

Remediation Guidance

Allow keyboard and touchscreen users the ability to re-arrange the boxes. This can either be done with dedicated buttons, such as are used within Gutenberg for moving Blocks; or it could be done by extending drag and drop support to keyboard and touch users (a good example of which is dragondrop: https://schne324.github.io/dragon-drop/demo/).

References

https://github.com/schne324/dragon-drop
https://www.smashingmagazine.com/2018/01/dragon-drop-accessible-list-reordering/#comments-dragon-drop-accessible-list-reordering

Relevant standards

- 2.1.1 Keyboard (Level A)
-

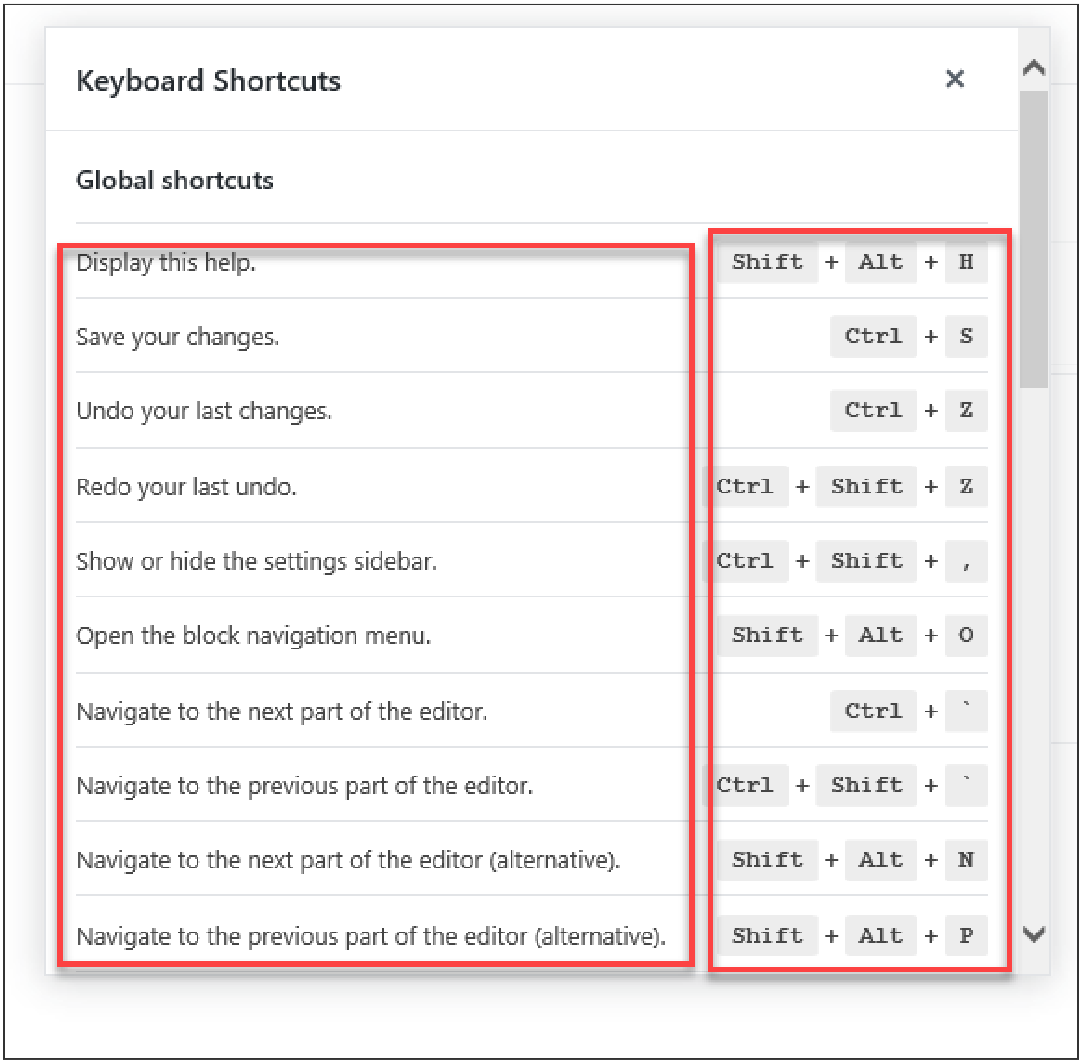
GUT-71 - Reading order does not match visual order

- **Severity:** Low
- **Affected Populations:** Low-Vision, Cognitively Impaired
- **Platform(s):** All/ Universal
- **Components affected:** Tools and Options

Issue description

When assistive technology users are reading through the "Keyboard Shortcuts" list, the items in the list are announced differently than their visual order: each shortcut keystroke is announced before its description, even though the description comes first visually.

Users who need an alternative presentation of content benefit from a logical reading order. Users who have a cognitive disability and have difficulty reading may also be confused by an unexpected or illogical reading order when using text-to-speech programs.



Keyboard Shortcuts

Global shortcuts

Display this help.	Shift + Alt + H
Save your changes.	Ctrl + S
Undo your last changes.	Ctrl + Z
Redo your last undo.	Ctrl + Shift + Z
Show or hide the settings sidebar.	Ctrl + Shift + ,
Open the block navigation menu.	Shift + Alt + O
Navigate to the next part of the editor.	Ctrl + `
Navigate to the previous part of the editor.	Ctrl + Shift + `
Navigate to the next part of the editor (alternative).	Shift + Alt + N
Navigate to the previous part of the editor (alternative).	Shift + Alt + P

Issue Code

```
dl  l    " di       k  b   d  h       h l    h       li  "
```

```
<dl class="edit-post-keyboard-shortcut-help__shortcut-list">
    <div class="...">
        <dt class="edit-post-keyboard-shortcut-help__shortcut-term">
            <kbd class="...">...</kbd>
        </dt>
        <dd class="edit-post-keyboard-shortcut-help__shortcut-description">
            Display this help.
        </dd>
    </div>
    ...
</dl>

...

.edit-post-keyboard-shortcut-help__shortcut-term {
    order: 1;
    ...
}
.edit-post-keyboard-shortcut-help__shortcut-description {
    flex: 1;
    order: 0;
    ...
}
```

Remediation Guidance

Avoid the use of flexbox `order` properties to change the order of elements, because this change is not reflected in the DOM reading order.

It appears as though this has been done so that the description can be the `<dd>` and the keystroke can be the `<dt>`, however this is not necessary. The description can be considered the term, since it's the primary frame of reference for a user (i.e. a user wouldn't encounter a keystroke and wonder what function it controls, they would encounter a function and wonder what keystroke can be used for it).

Therefore, the `order` property can be removed, and the markup inverted to use `<dt>` for the description and `<dd>` for the keystroke.

Recommended Code

```
<dl class="edit-post-keyboard-shortcut-help__shortcut-list">
    <div class="...">
        <dt class="edit-post-keyboard-shortcut-help__shortcut-term">
            Display this help.
        </dt>
        <dd class="edit-post-keyboard-shortcut-help__shortcut-description">
            <kbd class="...">...</kbd>
        </dd>
    </div>
    ...
</dl>
```

Relevant standards

- 1.3.2 Meaningful Sequence (Level A)

GUT-70 - Single visible menu is made up of multiple menus

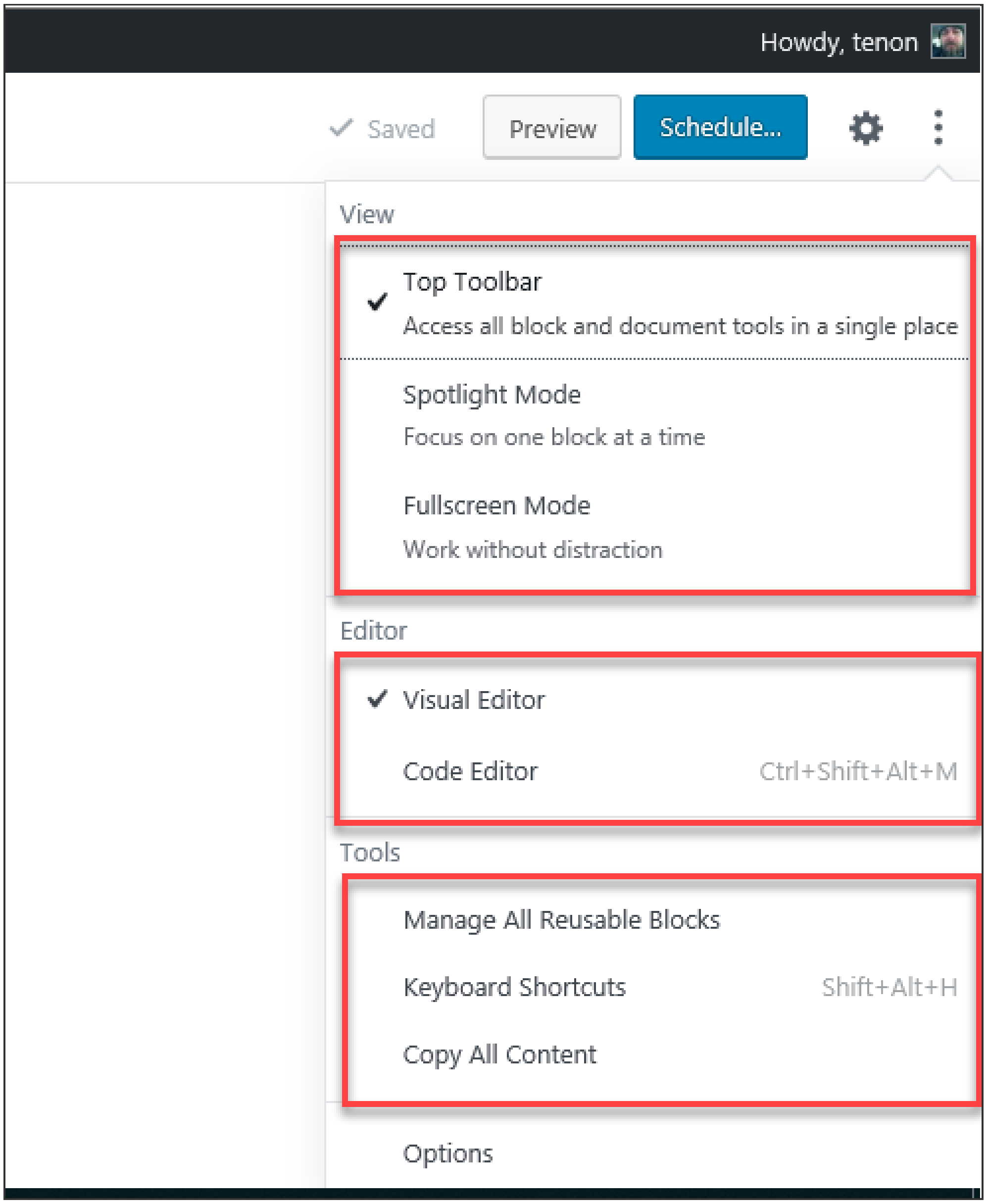
- **Severity:** High
- **Affected Populations:** Blind, Low-Vision, Cognitively Impaired
- **Platform(s):** All/ Universal
- **Components affected:** Tools and Options

Issue description

The Tools and Options menu is split into several sub-groups, each of which uses the ARIA menu pattern, and when a user navigates using the Arrow keys, they are unable to proceed past each group without using Tab (i.e. Arrow keys simple cycle through the items in one sub-group). The user is forced to switch between Tab and Arrow to navigate through the whole menu.

Keyboard users, especially those with cognitive disabilities, may find the current menu unpredictable and inefficient to use, because of its unconventional keystrokes. If they are not using a screen reader, they will not realise that there are multiple separate menus in a single dropdown.

Semantic markup is vital for assistive technology users, to convey the structure and relationships between information. When appropriate semantic markup is not used, it may be more difficult for such users to understand what the content means, and to comprehend the structure that the visual presentation conveys.



View

✓ Top Toolbar
Access all block and document tools in a single place

Spotlight Mode
Focus on one block at a time

Fullscreen Mode
Work without distraction

Editor

✓ Visual Editor

Code Editor Ctrl+Shift+Alt+M

Tools

Manage All Reusable Blocks

Keyboard Shortcuts Shift+Alt+H

Copy All Content

Options

Issue Code

```
di    bi d   "  "  l    "                            "    l  "   h i h          "
```

```
<div tabindex="-1" class="components-popover__content" style="max-height: 247.5px;">
    <div class="...">
        <div class="..." id="components-menu-group-label-0">View</div>
        <div role="menu" aria-labelledby="components-menu-group-label-0" aria-orientation="vertical">
            <button class="..." role="menuitemcheckbox" aria-checked="true" aria-describedby="edit-post-feature-toggle__info-0" aria-label="Top Toolbar" type="button">
                ...
            </button>
            <button class="..." role="menuitemcheckbox" aria-checked="false" aria-describedby="edit-post-feature-toggle__info-1" aria-label="Spotlight Mode" type="button">
                ...
            </button>
            <button class="..." role="menuitemcheckbox" aria-checked="false" aria-describedby="edit-post-feature-toggle__info-2" aria-label="Fullscreen Mode" type="button">
                ...
            </button>
        </div>
    </div>
    <div class="...">
        <div class="..." id="...">Editor</div>
        <div role="menu" aria-labelledby="...." aria-orientation="vertical">
            <button class="..." role="menuitemradio" aria-checked="true" aria-label="Visual Editor" type="button">
                ...Visual Editor
            </button>
            <button class="..." role="menuitemradio" aria-checked="false" aria-label="Code Editor" type="button">
                Code Editor...
            </button>
        </div>
    </div>
    <div class="...">
        <div class="..." id="...">Tools</div>
        <div role="menu" aria-labelledby="..." aria-orientation="vertical">
            ...
        </div>
    </div>
    <div class="...">
        <div role="menu" aria-labelledby="..." aria-orientation="vertical">
            <button class="..." role="menuitem" aria-label="Options" type="button">Options</button>
        </div>
    </div>
</div>
```

Remediation Guidance

Using a single ARIA menu (https://www.w3.org/TR/wai-aria-practices-1.1/#menu) would be better than multiple ARIA menus, because then it would have consistent keyboard behaviours that are recognisable to both sighted keyboard users and assistive technology users. However this pattern does not have any scope for providing sub-groups and group headings, and the current solution of splitting the menu is potentially confusing for users.

So it may be best to have the "Tools and Options" button open a popup, rather than a menu, which contains a form with typical form controls in fieldset groups. This approach is shown in the Recommended Code.

Other possible alternatives to this recommendation are:

* Single unordered-list, where each section is a list-item containing a heading and a child list (navigated via the Tab key, not Arrow keys).
* Popup containing multiple unordered lists grouped with headings.
* Remove the group headings and implement the menu as a single ARIA menu.

In any case, the Options interface should be navigable as a single unit, without needing to switch between Tab and Arrow to navigate through the whole thing.

Recommended Code

```
<div tabindex="-1" class="components-popover__content" style="max-height: 247.5px;">
    <fieldset class="...">
        <legend class="..." id="components-menu-group-label-0">View</legend>
            <div>
                <input type="checkbox" id="edit-post-feature-toggle-0" aria-describedby="edit-post-feature-toggle__info-0">
                <label for="edit-post-feature-toggle-0">Top Toolbar"</label>
                <span id="edit-post-feature-toggle__info-0" class="...">Access all block and document tools in a single place</span>
            </div>
            <div>
                <input type="checkbox" id="edit-post-feature-toggle-1" aria-describedby="edit-post-feature-toggle__info-1">
                <label for="edit-post-feature-toggle-1">Spotlight Mode</label>
                <span id="edit-post-feature-toggle__info-1" class="...">Focus on one block at a time</span>
            </div>
            <div>
                <input type="checkbox" id="edit-post-feature-toggle-2" aria-describedby="edit-post-feature-toggle__info-2">
                <label for="edit-post-feature-toggle-2">Fullscreen Mode</label>
                <span id="edit-post-feature-toggle__info-2" class="...">Work without distraction</span>
        </div>
    </fieldset>
    <fieldset class="...">
        <legend class="..." id="...">Editor</legend>
        <div>
            <input type="radio" checked id="editor_type">
            <label for="editor_type">Visual Editor</label>
        </div>
        <div>
            <input type="radio" id="editor_type-2" aria-labelledby="components-shortcut-2">
            <label for="editor_type-2">Code Editor</label>
             <span id="components-shortcut-2">Ctrl+Shift+Alt+M</span>
        </div>
    </fieldset>
    <fieldset class="...">
        <legend class="..." id="...">Tools</legend>
        <div>
            <a href="edit.php?post_type=wp_block" class="...">Manage All Reusable Blocks</a>
            <button type="button" class="....">Keyboard Shortcuts <span class="components-menu-item__shortcut">Shift+Alt+H</span></button>
            <button type="button" class="...">Copy All Content</button>
        </div>
    </fieldset>
    <fieldsetclass="...">
        <button class="..." type="button">Options</button>
    </fieldset>
</div>
```

References

https://github.com/WordPress/gutenberg/issues/12505

Relevant standards

- 1.3.1 Info and Relationships (Level A)
- 2.1.1 Keyboard (Level A)
- 4.1.2 Name, Role, Value (Level A)

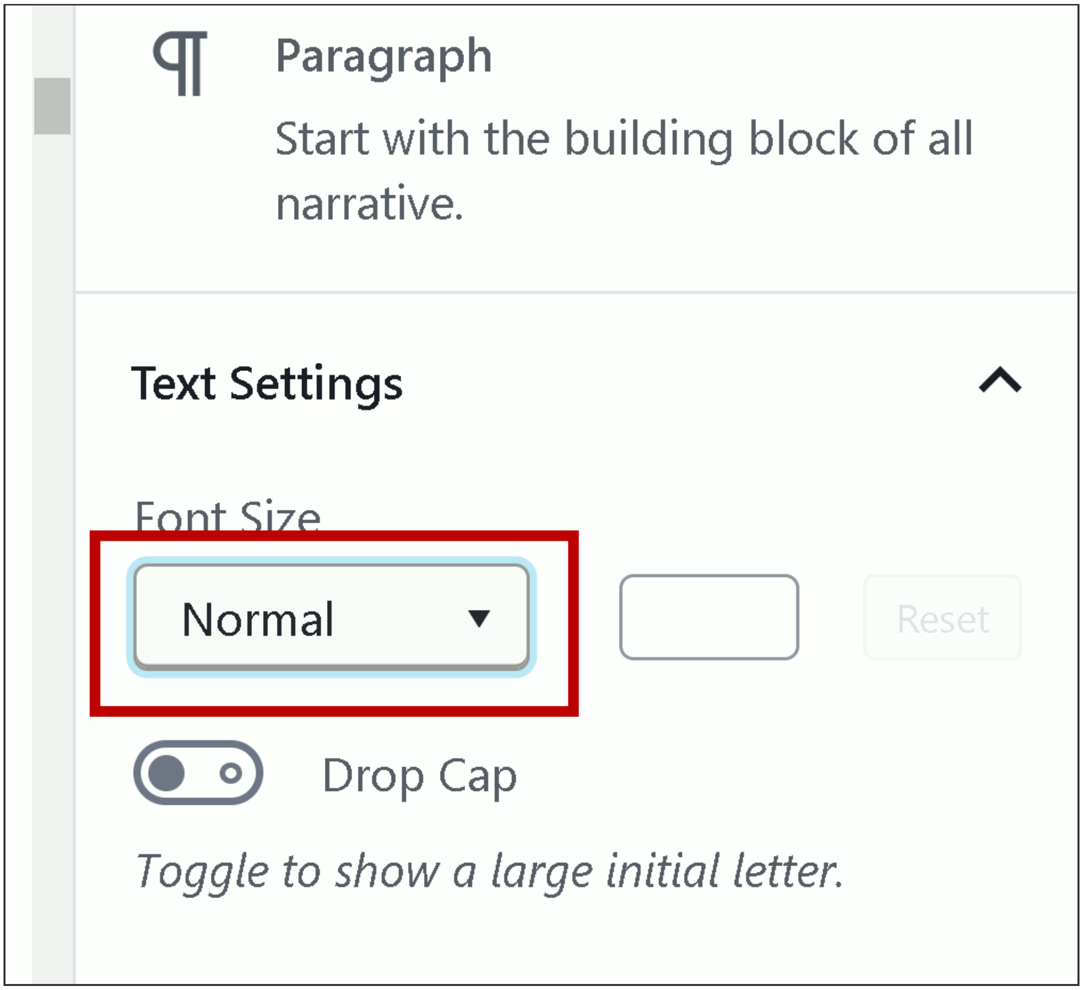
GUT-68 - Custom interactions in control styled as a native control

- **Severity:** Medium
- **Affected Populations:** Low-Vision, Motor Impaired, Cognitively Impaired
- **Platform(s):** All/ Universal
- **Components affected:** Block Panel

Issue description

In the Block panel, where the font size can be selected, users see what appears to be a select element. However when keyboard users reach this component, they cannot use standard keystrokes to change the value, such as Up-Arrow or Down-Arrow, or Alt+Arrow to open the submenu. Instead, users need to press Enter to view the submenu.

Keyboard users will expect this control to work the same as a native select element. Speech recognition users often speak out keystrokes to perform standard actions on native components, and these would not work on custom components like these.



Paragraph

Start with the building block of all narrative.

Text Settings

Font Size

Normal

Drop Cap

Toggle to show a large initial letter.

Issue Code

```
<button type="button" aria-expanded="false" aria-label="Font size: Normal" class="components-button components-
font-size-picker__selector is-button is-default is-large">Normal</button>
```

Remediation Guidance

Use a native `<select>` element in place of a custom button+submenu setup.

Recommended Code

```
<fieldset class="components-base-control__field">
    <legend class="components-base-control__label">Font Size</legend>
    <label for="font-size-picker">Choose basic size</label>
    <select id-"font-size-picker" class="attachment-filters">
        <option value="normal" >Normal</option>
        ...
    </select>
    <label for="custom-font-size">Custom font size</label>
    <input id="custom-font-size" class="components-range-control__number" type="number" value="">
    <button type="button" disabled="" class="components-button components-color-palette__clear is-button is-default is-small">Reset</button>
</fieldset>
```

Relevant standards

- 2.1.1 Keyboard (Level A)

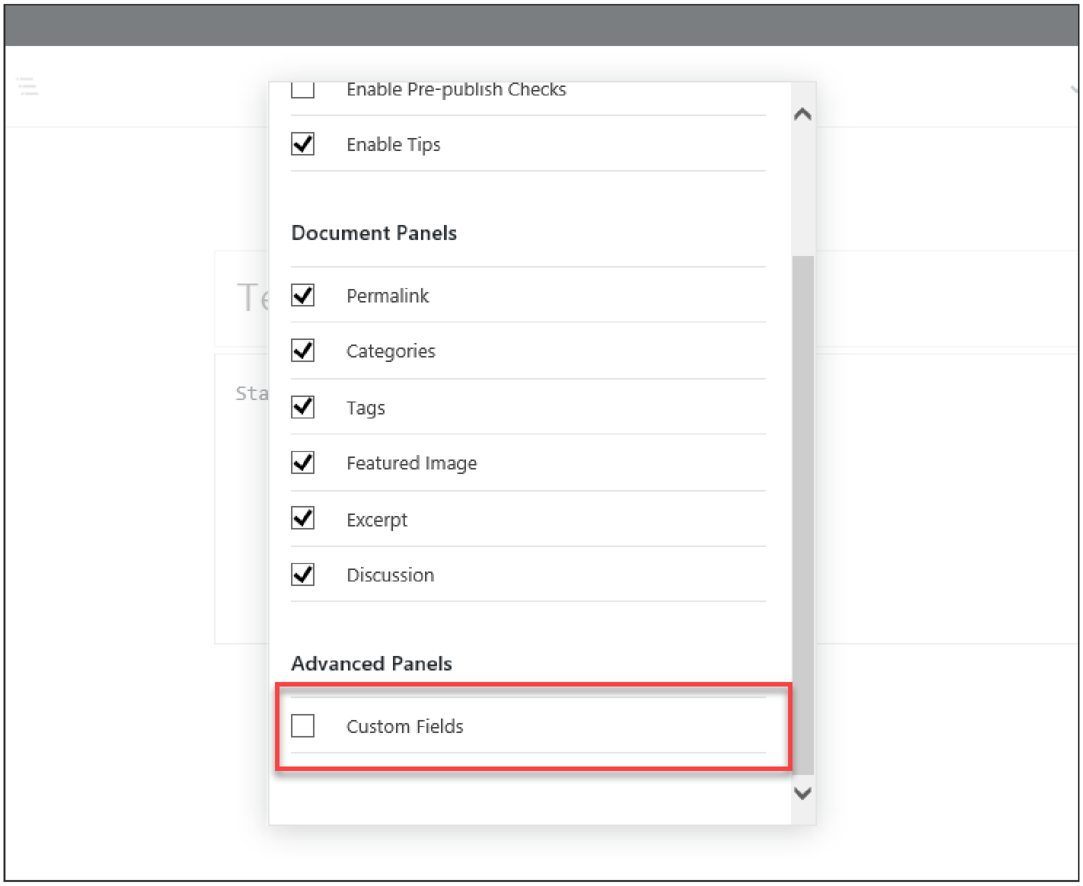
GUT-65 - Custom field check box automatically changes the context on user focus

- **Severity:** High
- **Affected Populations:** Blind, Low-Vision, Motor Impaired, Cognitively Impaired
- **Platform(s):** All/ Universal
- **Components affected:** Tools and Options

Issue description

The "Custom Fields" checkbox within the "Options" dialog causes an unexpected page refresh when the user selects it, and consequently resets the focus position to the top of the page.

This is not expected behaviour for checkboxes, and the sudden change of context may be confusing for assistive technology users, while the loss of focus position will force keyboard users to manually navigate back to where they were before.



□ Enable Pre-publish Checks

☑ Enable Tips

Document Panels

☑ Permalink

☑ Categories

☑ Tags

☑ Featured Image

☑ Excerpt

☑ Discussion

Advanced Panels

□ Custom Fields

Remediation Guidance

If it's not possible to update the page in the background, as happens with other checkboxes in the "Options" dialog, then the user should be warned (with visible text, not visually-hidden) that selecting this option will close the options dialog and reload the page.

Alternatively, an "Apply Changes" button could be added to the dialog, which confirms the user's changes for all checkboxes (i.e. none of the checkboxes cause background updates or page reloads until the user explicitly presses the Apply button).

Relevant standards

- 3.2.5 Change on Request (Level AAA)

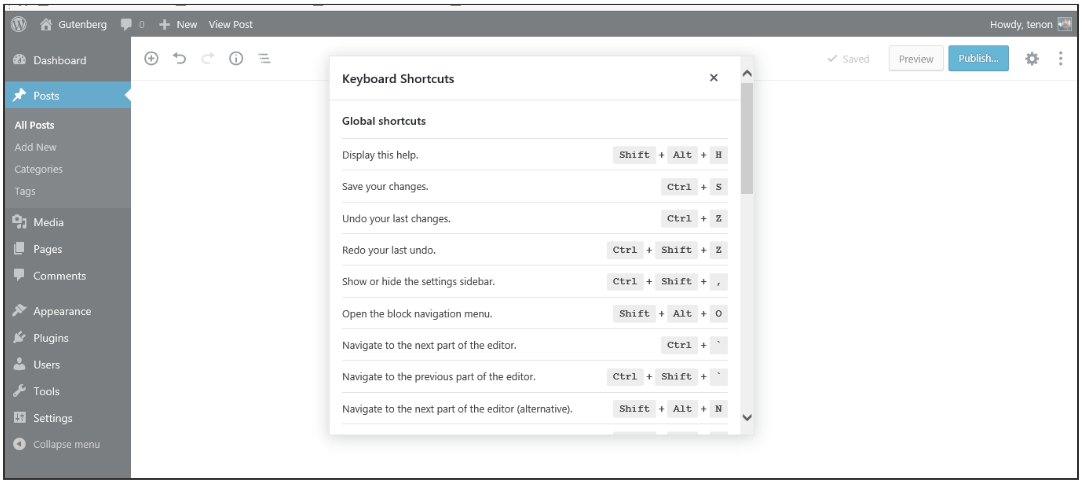
GUT-53 - Keyboard focus is not managed when dialogs close

- **Severity:** Medium
- **Affected Populations:** Blind, Low-Vision, Motor Impaired, Cognitively Impaired
- **Platform(s):** All/ Universal
- **Components affected:** Tools and Options

Issue description

When the "Keyboard Shortcuts" and "Options" dialogs are dismissed, the keyboard focus is reset to the top of the page. This may be confusing or inconvenient for users, who would expect the focus to return to the last element they were focused on.

Keyboard and screen reader users rely on predictable and consistent focus control to navigate the page. When this is absent, users may find it much more difficult or impossible to navigate using the keyboard.



Keyboard Shortcuts

Global shortcuts

Display this help.	Shift + Alt + H
Save your changes.	Ctrl + S
Undo your last changes.	Ctrl + Z
Redo your last undo.	Ctrl + Shift + Z
Show or hide the settings sidebar.	Ctrl + Shift + ,
Open the block navigation menu.	Shift + Alt + O
Navigate to the next part of the editor.	Ctrl + `
Navigate to the previous part of the editor.	Ctrl + Shift + `
Navigate to the next part of the editor (alternative).	Shift + Alt + N

Remediation Guidance

Ideally, use JavaScript to continually track the keyboard focus position, so that when dialogs are opened via shortcut keys, the focus can be returned to the previously-focused element when they close.

If that's not practical, then closing these dialogs should set focus on the "Tools and Options" button.

Relevant standards

- 2.1.1 Keyboard (Level A)

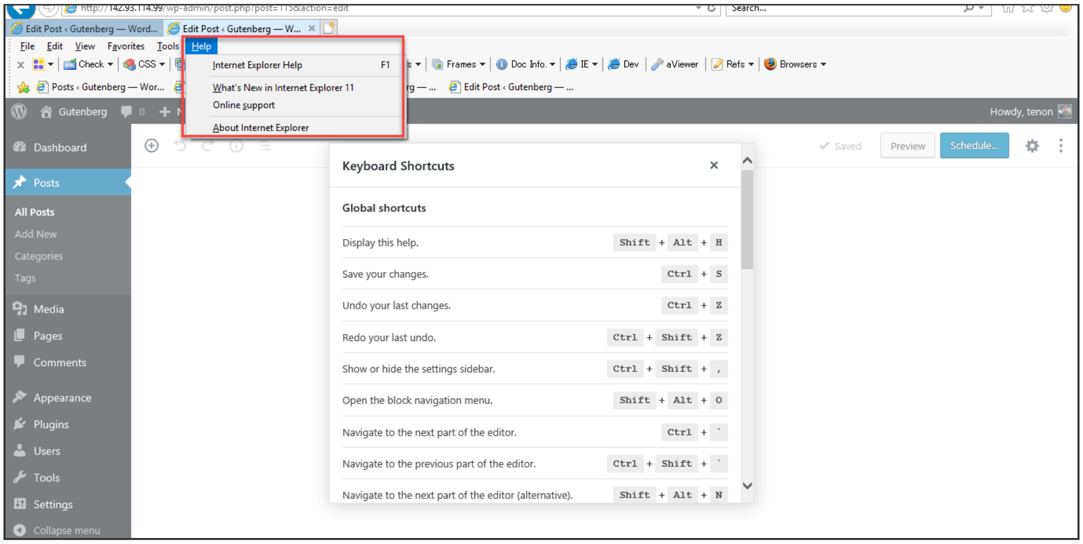
GUT-51 - Shortcut keys conflict with browser shortcuts

- **Severity:** Medium
- **Affected Populations:** Low-Vision, Motor Impaired, Cognitively Impaired
- **Platform(s):** All/ Universal
- **Components affected:** Tools and Options

Issue description

The shortcut key provided for "Keyboard Shortcuts" is `Shift+Alt+H`, however this also triggers the browser "Help" dialog in Windows browsers such as Internet Explorer.

This may be confusing for users who rely on shortcut keys, since they would have to manually escape the native menu in order to reach the Keyboard Shortcuts information.



Keyboard Shortcuts ×

Global shortcuts

Display this help.	Shift + Alt + H
Save your changes.	Ctrl + S
Undo your last changes.	Ctrl + Z
Redo your last undo.	Ctrl + Shift + Z
Show or hide the settings sidebar.	Ctrl + Shift + ,
Open the block navigation menu.	Shift + Alt + O
Navigate to the next part of the editor.	Ctrl + `
Navigate to the previous part of the editor.	Ctrl + Shift + `
Navigate to the next part of the editor (alternative).	Shift + Alt + N

Remediation Guidance

Replace this shortcut key with something else which does not conflict with browser defaults, or allow users to change the shortcut keys.

GUT-45 - Focus is invisible in high contrast views

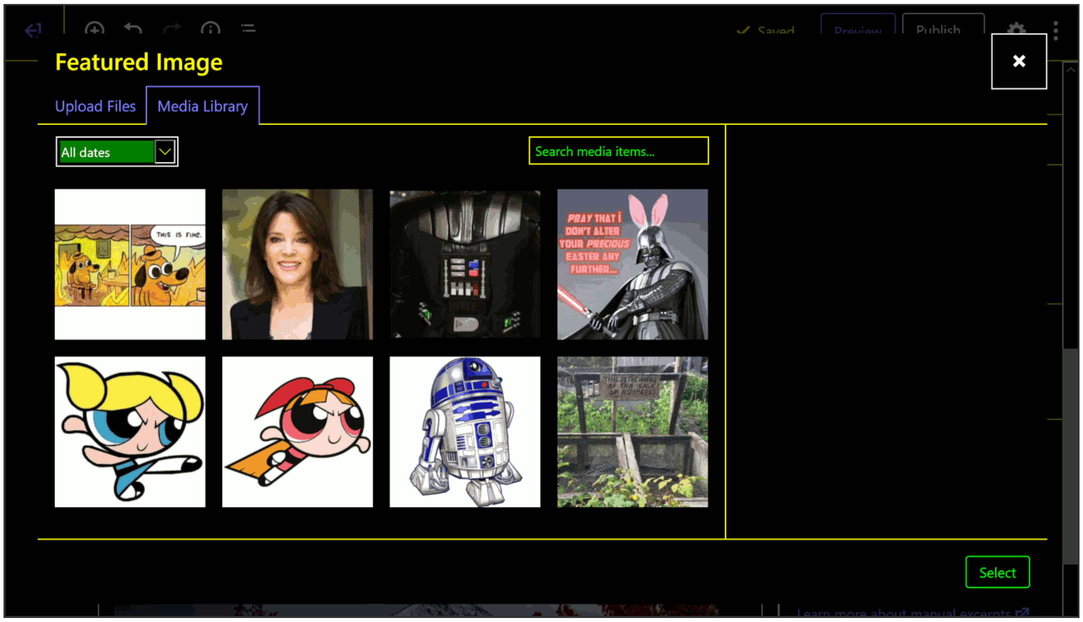
- **Severity:** High
- **Affected Populations:** Low-Vision, Motor Impaired, Cognitively Impaired
- **Platform(s):** Windows - High Contrast
- **Components affected:** Edit Media, Media Dialog

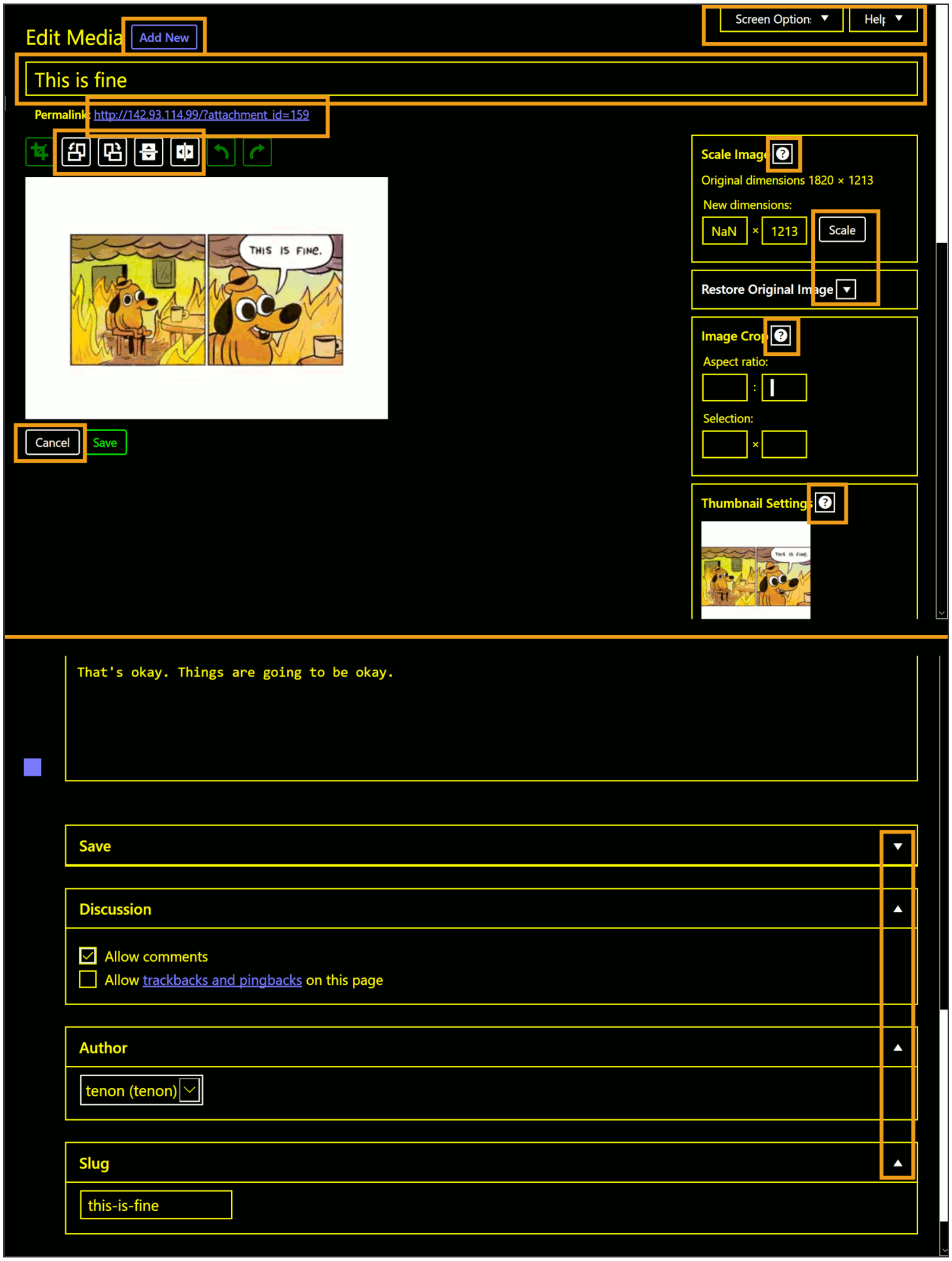
Issue description

Users of Windows High Contrast themes who open the "Featured Image" modal or the Edit Media page cannot see where the keyboard focus is in most cases. The exceptions to this are the date filter dropdown, pre-filled text inputs, and the search typeahead-inputs, but these exceptions are only because the browser tends to automatically select and highlight the text inside them, which should not be relied on.

This problem occurs because `box-shadow` is the sole visible method of showing which element has focus, however High Contrast themes remove box-shadows along with background colors to ensure that foregrounds and backgrounds have sufficiently high contrast.

All interactive elements should have a visible and obvious focus indicator, so that sighted keyboard users have a clear indication of which element has focus.


Saved
Preview
Publish
Featured Image
×
Upload Files
Media Library
All dates
Search media items...
THIS IS FINE.
PRAY THAT I DON'T ALTER YOUR PRECIOUS EASTER ANY FURTHER...
Select
Learn more about manual excerpts



This is fine

Permalink: http://142.93.114.99/?attachment_id=159

Scale Image

Original dimensions 1820 × 1213

New dimensions:

NaN × 1213

Image Crop

Aspect ratio:

:

Selection:

×

Thumbnail Settings

That's okay. Things are going to be okay.

Discussion ▲

☑ Allow comments

□ Allow trackbacks and pingbacks on this page

Author ▲

tenon (tenon)

Slug ▲

this-is-fine

Issue Code

i h d il f

```
.wp-core-ui .attachment.details:focus,

.wp-core-ui .attachment:focus,

.wp-core-ui .selected.attachment:focus {

    box-shadow: inset 0 0 2px 3px #fff, inset 0 0 0 7px #5b9dd9;

    outline: 0;

}

...

.wrap .page-title-action:focus {

    border-color: #5b9dd9;

    box-shadow: 0 0 2px rgba(30,140,190,.8);

}

...

.image-editor .imgedit-settings .imgedit-help-toggle:focus {

    color: #0074a2;

    border-color: #5b9dd9;

    outline: 0;

    box-shadow: 0 0 3px rgba(0,115,170,.8);

}

...

input[type=checkbox]:focus, input[type=color]:focus, input[type=date]:focus, input[type=datetime-local]:focus,
input[type=datetime]:focus, input[type=email]:focus, input[type=month]:focus, input[type=number]:focus,
input[type=password]:focus, input[type=radio]:focus, input[type=search]:focus, input[type=tel]:focus,
input[type=text]:focus, input[type=time]:focus, input[type=url]:focus, input[type=week]:focus, select:focus,
textarea:focus {

    border-color: #5b9dd9;

    box-shadow: 0 0 2px rgba(30,140,190,.8);

}

...

.wp-core-ui .button-primary.focus, .wp-core-ui .button-primary:focus {

    box-shadow: 0 1px 0 #0073aa, 0 0 2px 1px #33b3db;

}

...

a, div {

    outline: 0;

}

...

.wp-person a:focus .gravatar, a:focus, a:focus .media-icon img {

    color: #124964;

    box-shadow: 0 0 0 1px #5b9dd9, 0 0 2px 1px rgba(30,140,190,.8);

}

...

.js .postbox .handlediv:focus {

    box-shadow: none;

    outline: 0;

}
```

```
.js .postbox .handlediv:focus .toggle-indicator:before {
    box-shadow: 0 0 0 1px #5b9dd9, 0 0 2px 1px rgba(30,140,190,.8);
}
```

Remediation Guidance

Add a transparent (or visible) `outline` to elements which are using `box-shadow` to show focus.

For elements which are selected and focussed, selected elements should also have a `border`.

Ensure the outline is not given a negative `outline-offset`, as this would place it directly on top of the border, making it visibly indistinguishable by users.

Recommended Code

```
.wp-core-ui .attachment.details:focus,
.wp-core-ui .attachment:focus,
.wp-core-ui .selected.attachment:focus {
    box-shadow: inset 0 0 2px 3px #fff, inset 0 0 0 7px #5b9dd9;
    outline: 3px solid transparent;
}

.wp-core-ui .attachment.details {
    box-shadow: inset 0 0 0 3px #fff, inset 0 0 0 7px #0073aa;
    border: 1px solid transparent;
}
...
.wrap .page-title-action:focus {
    border-color: #5b9dd9;
    box-shadow: 0 0 2px rgba(30,140,190,.8);
    outline: 3px solid transparent;
}
...
.image-editor .imgedit-settings .imgedit-help-toggle:focus {
    color: #0074a2;
    border-color: #5b9dd9;
    outline: 1px solid transparent;
    box-shadow: 0 0 3px rgba(0,115,170,.8);
}
...
input[type=checkbox]:focus, input[type=color]:focus, input[type=date]:focus, input[type=datetime-local]:focus,
input[type=datetime]:focus, input[type=email]:focus, input[type=month]:focus, input[type=number]:focus,
input[type=password]:focus, input[type=radio]:focus, input[type=search]:focus, input[type=tel]:focus,
input[type=text]:focus, input[type=time]:focus, input[type=url]:focus, input[type=week]:focus, select:focus,
textarea:focus {
     border-color: #3d94b7;
    box-shadow: 0 0 2px rgba(30,140,190,.8);
    outline: 1px solid transparent;
}
...
.wp-core-ui .button-primary.focus, .wp-core-ui .button-primary:focus {
    box-shadow: 0 1px 0 #0073aa, 0 0 2px 1px #33b3db;
    border: 1px solid #000;
    outline: 1px solid transparent;
}
...
a, div {
    outline: 0;
}
...
```

```
.wp-person a:focus .gravatar, a:focus, a:focus .media-icon img {
    color: #124964;
    box-shadow: 0 0 0 1px #5b9dd9, 0 0 2px 1px rgba(30,140,190,.8);
    outline: 1px solid transparent;
}
...
.js .postbox .handlediv:focus {
    box-shadow: none;
    outline: 0;
}
.js .postbox .handlediv:focus .toggle-indicator:before {
    box-shadow: 0 0 0 1px #5b9dd9, 0 0 2px 1px rgba(30,140,190,.8);
    border: 1px solid transparent;
}
```

Relevant standards

- 2.4.7 Focus Visible (Level AA)

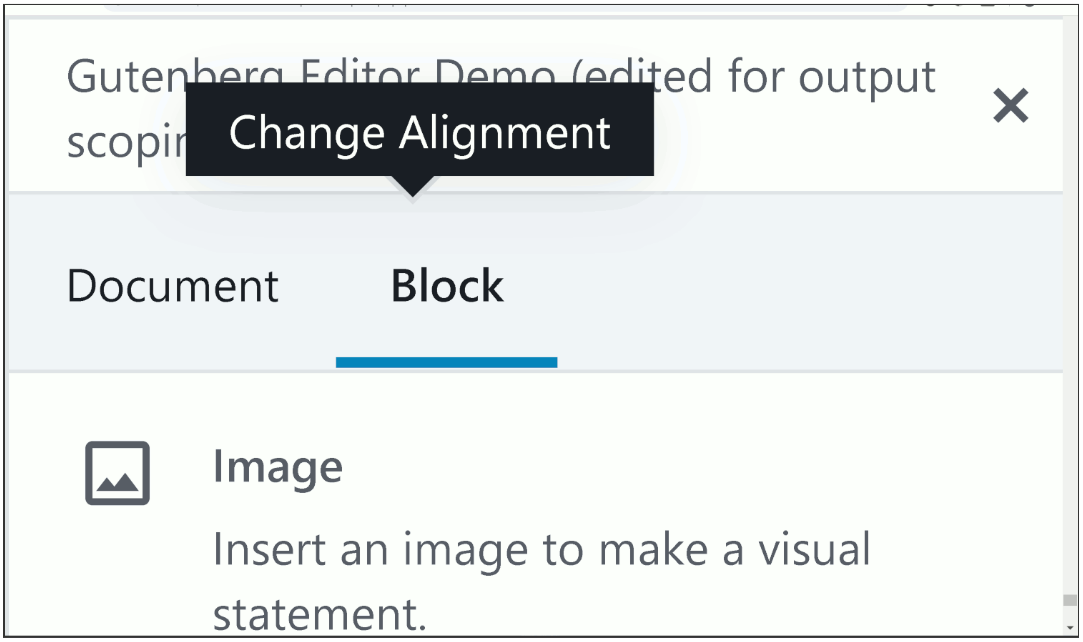
GUT-33 - Keyboard focus cannot easily reach Settings panel after opening

- **Severity:** High
- **Affected Populations:** Blind, Low-Vision, Motor Impaired, Cognitively Impaired
- **Platform(s):** All/ Universal
- **Components affected:** Block Panel, Document Panel

Issue description

Keyboard users who use the "Settings" button to open the Settings panel cannot easily reach the newly-opened panel using typical keyboard navigation. The Settings content does not follow the triggering button in the DOM, nor is focus programmatically moved there once it opens.

This is especially problematic for users who are zoomed in above 175%, where the mobile styles apply, showing the Settings panel taking up the entire page similar to a dialog. Users who attempt to tab through the panel find themselves tabbing through elements behind the panel. The screenshot shows a tooltip from something getting focus behind the panel.



Insert an image to make a visual statement.

Issue Code

```
<div role="region" aria-label="Editor top bar" class="edit-post-header" tabindex="-1">
    <div role="toolbar" ..." aria-label="Document and block tools">...</div>
    <div class="edit-post-header__settings">
        ...
        <button type="button" aria-label="Settings" aria-expanded="true" class="...">...</button>
    </div>
    <div class="edit-post-more-menu">
        <button type="button" aria-label="Show more tools &amp; options" aria-expanded="false" class="...">...
</button>
    </div>
</div>
...
<div class="edit-post-layout__content" role="region" aria-label="Editor content" tabindex="-1">...</div>
...
<div class="edit-post-sidebar" role="region" aria-label="Editor settings" tabindex="-1">
    <div class="...">
        <span class="edit-post-sidebar-header__title">Gutenberg Editor Demo (edited for output scoping)</span>
        <button type="button" aria-label="Close settings" class="...">...</button>
    </div>
    <div class="...">
        <ul>
            <li>
                <button class="..." aria-label="Document (selected)" data-label="Document">Document</button>
            </li>
            <li>
                <button class="... " aria-label="Block" data-label="Block">Block</button>
            </li>
        </ul>
        <button type="button" aria-label="Close settings" class="...">...</button>
    </div>
    ...
</div>
```

Remediation Guidance

Either move the dropdown contents to come immediately after their activating buttons in source order OR programmatically move focus to the first focusable item inside the dropdowns when they've been opened (even if opened by a shortcut key).

Shortcut keys for moving around a document should be considered extra UX features for power users, rather than essential for basic navigation. Focus should follow a predictable order, generally following visible items, and should not be able to reach invisible areas.

Relevant standards

- 1.4.10 Reflow (Level AA)
- 2.4.3 Focus Order (Level A)
- 2.4.7 Focus Visible (Level AA)

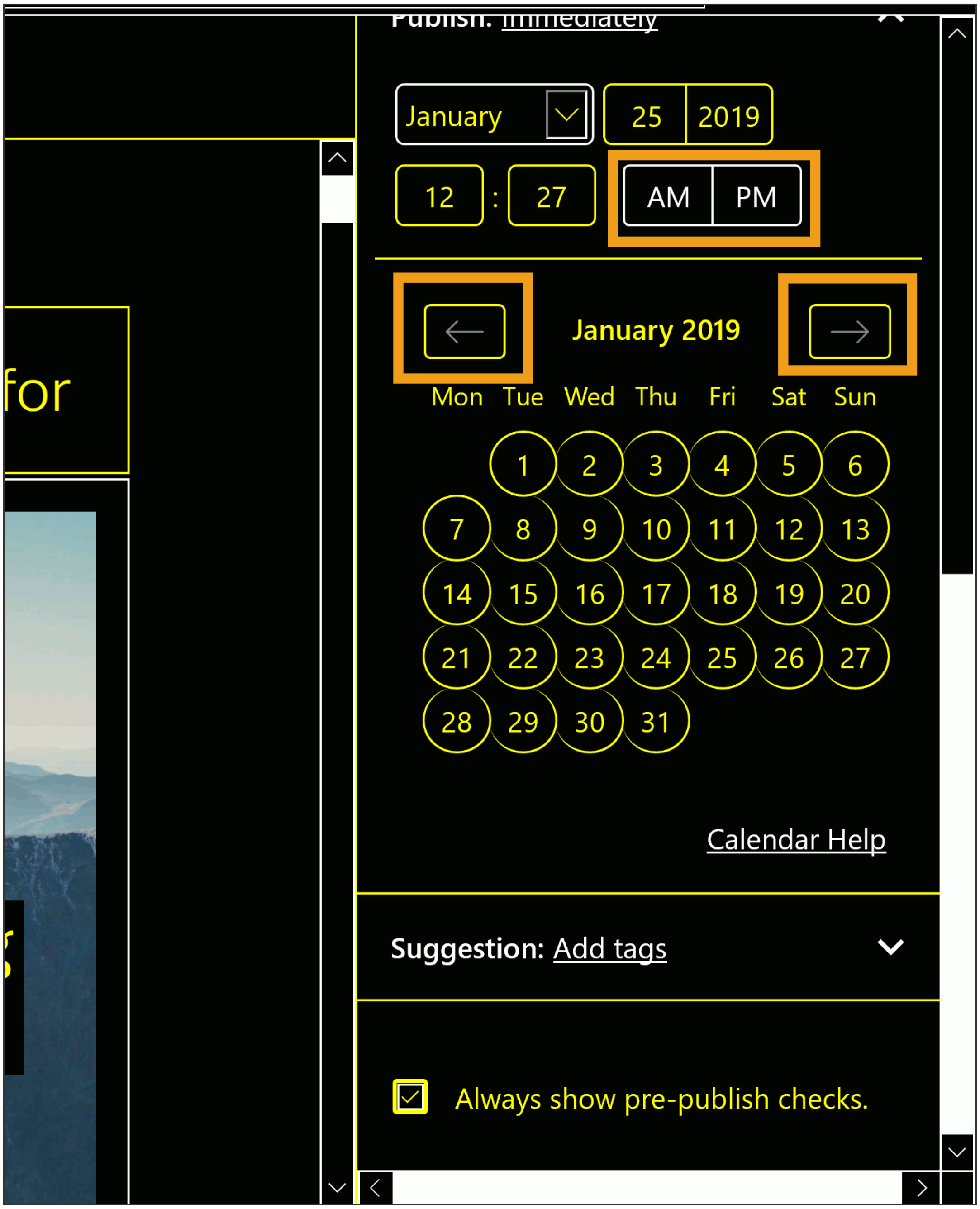
GUT-29 - States are not visible in High Contrast views

- **Severity:** High
- **Affected Populations:** Low-Vision, Motor Impaired, Cognitively Impaired
- **Platform(s):** Windows - High Contrast
- **Components affected:** Block Editing, Block Panel, Document Panel, Publish and Unpublish

Issue description

Keyboard users who rely on Windows High Contrast themes cannot see their focus or the pressed state on parts of the components listed. Specifically, the focus states cannot be differentiated on the "AM" and "PM" buttons, the next-month and previous-month buttons, or the "Move to Trash" buttons, and there is no visual differentiation between pressed and not on the "AM" and "PM" buttons.

All interactive elements should have visible and obvious state indicators, so that sighted keyboard users have a clear indication of which element has focus, and which state it's in.



Publish: immediately

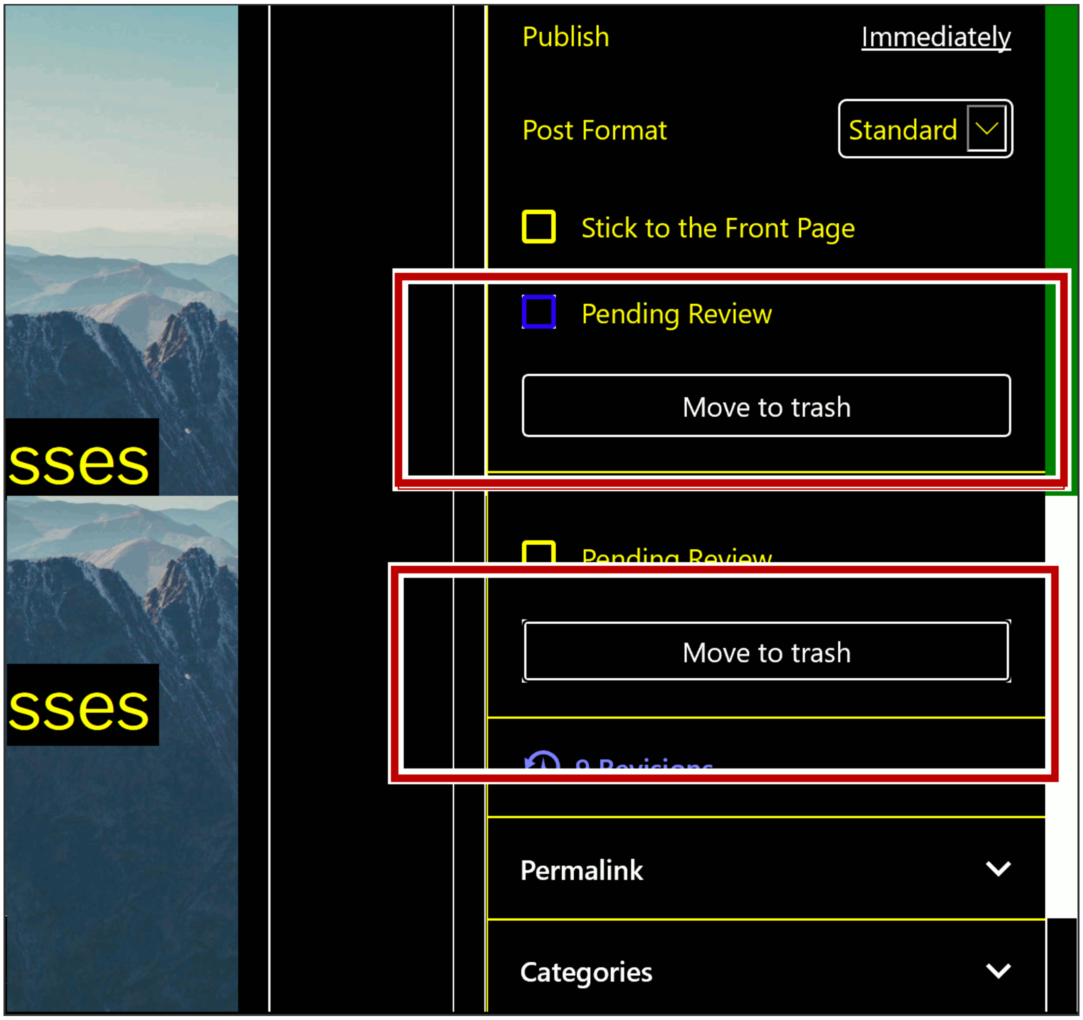
January 25 2019

12 : 27 AM PM

Mon	Tue	Wed	Thu	Fri	Sat	Sun
	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	31			

Suggestion: Add tags

☑ Always show pre-publish checks.



Publish Immediately

Post Format Standard

Issue Code

```
/* i  i ibl        d        f   /   b       */
/* invisible pressed state of AM/PM buttons */
...,
.components-datetime__time .components-datetime__time-wrapper .components-datetime__time-pm-button.is-toggled {
    ...
    box-shadow: inset 0 2px 5px -3px #555d66;
}

/* invisible focus states on Next/Previous Month buttons */
/* default */
.DayPickerNavigation_button__default {
    border: 1px solid #e4e7e7;
    ...
}
/* focussed */
.DayPickerNavigation_button__default:focus, .DayPickerNavigation_button__default:hover {
    border: 1px solid #c4c4c4;
}

/* Move to Trash button */
.components-button:focus:enabled {
    ...
   box-shadow: inset 0px 0px 0px 1px #6c7781, inset 0px 0px 0px 2px #fff;
    outline: 2px solid transparent;
    outline-offset: -2px;
}

/* buttons using SVG */
<button type="button" aria-label="Heading 6" data-subscript="6" aria-pressed="true" class="...l is-active">
    <svg aria-hidden="true" role="img" focusable="false" ...>...</svg></button>
...
.components-toolbar__control.components-button:not(:disabled).is-active>svg {
    outline: 0;
    color: #fff;
    box-shadow: none;
    background: #555d66;
}

/* color pickers */
.components-color-palette__item.is-active {
    box-shadow: inset 0 0 0 4px;
}
```

Remediation Guidance

As mentioned in ~~GUT-20~~, darken the `border` or `box-shadow` colors to meet minimum color ratios for non-text contrast.

Add a thick transparent outline to focussed buttons for Windows High Contrast users. For buttons such as "Move to Trash" where there is an `outline`, remove the negative `outline-offset` which is currently preventing the new outline from being visible against the border.

For the toggled or `pressed` state of the AM/PM buttons, use a visible difference in high contrast to show the `pressed` state (such as a persistent outline) and something different such as a dotted outline to reveal the focussed state.

The Recommended Code suggests one way of achieving this, but the important take-away is that changes in border color are not revealed in High Contrast views, whereas changes in border thickness and styles are visible.

Recommended Code

```
/* pressed state of AM/PM buttons */
...,
.components-datetime__time .components-datetime__time-wrapper .components-datetime__time-pm-button.is-toggled {
    ...
    box-shadow: inset 0 2px 5px -3px #00435d;
    outline: 3px solid transparent; /* or a thicker border than the non-toggled button... */
}
.components-datetime__time .components-datetime__time-wrapper .components-datetime__time-pm-button.is-toggled:focus {
    ...
    outline: 1px dotted transparent;
    outline-offset: 3px;
}

/* invisible focus states on Next/Previous Month buttons */
/* default */
.DayPickerNavigation_button__default {
    border: 1px solid #e4e7e7;
    ...
}
/* focussed */
.DayPickerNavigation_button__default:focus, .DayPickerNavigation_button__default:hover {
    border: 1px solid #8E8E8E;
    outline: 3px solid transparent;
}
/* Move to Trash button */
.components-button:focus:enabled {
    box-shadow: inset 0px 0px 0px 1px #00435d, inset 0px 0px 0px 2px #fff;
    outline: 2px solid transparent;
}

/* buttons using SVG */
.components-toolbar__control.components-button:not(:disabled).is-active>svg {
    outline: 0;
    color: #fff;
    box-shadow: none;
    background: #555d66;
    border: 1px solid transparent;
}


/* color pickers */
.components-color-palette__item.is-active {
    box-shadow: inset 0 0 0 4px;
    border: 1px dotted transparent;

}


<!-- displayed color -->

<span class="components-base-control__label">Background Color

    <span class="component-color-indicator" aria-label="(current background color: Vivid red)" style="color: transparent;border: 0;background: rgb(207, 46, 46);">Vivid red</span>

</span>
```

Relevant standards

- 1.3.3 Sensory Characteristics (Level A)
- 2.4.7 Focus Visible (Level AA)

Related Issues

- Related to "Buttons are low contrast and focus not visible"

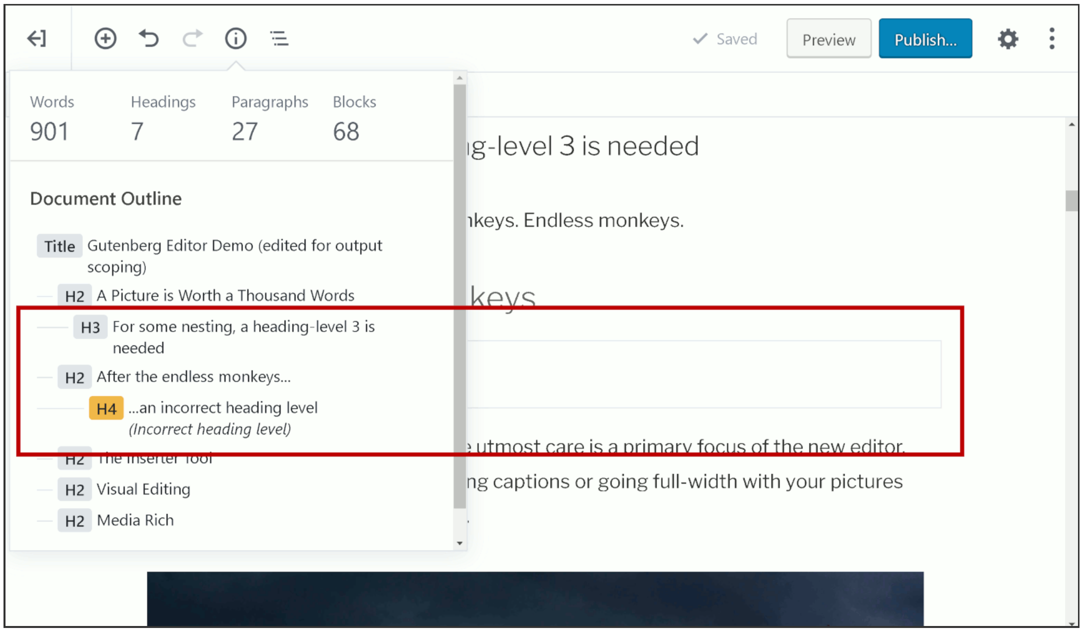
GUT-27 - Content Structure dropdown remains open after focus moves to page

- **Severity:** High
- **Affected Populations:** Low-Vision, Motor Impaired, Cognitively Impaired
- **Platform(s):** All/ Universal
- **Components affected:** Structure and Navigation

Issue description

When users open the "Block Navigation" dropdown and click on an element, the dropdown closes and focus can be seen on which Block was selected. However, when users open the Content Structure dropdown and click on an element, the dropdown remains open while focus moves to the specified block. The open dropdown obscures the focussed element, and the Escape key can no longer close the dropdown.

Obscuring the focus position is potentially confusing for sighted keyboard users, since it may appear as though no item has focus, or make it unclear as to where the focus position is.



Words	Headings	Paragraphs	Blocks
901	7	27	68

Document Outline

Title Gutenberg Editor Demo (edited for output scoping)

H2 A Picture is Worth a Thousand Words

H3 For some nesting, a heading-level 3 is needed

H2 After the endless monkeys...

H4 ...an incorrect heading level *(Incorrect heading level)*

H2 The Inserter Tool

H2 Visual Editing

H2 Media Rich

✓ Saved
Preview
Publish...
ıg-level 3 is needed
ıkeys. Endless monkeys.
keys
e utmost care is a primary focus of the new editor.
ng captions or going full-width with your pictures

Remediation Guidance

Depending on the reason why the Content Structure remains open, there are two potential solutions:

1. If the idea is to keep the document outline open while users' focus is on the page, then shrinking the page's width so that the open dropdown is on the left while the page is on the right (without horizontal scrollbars on non-zoomed desktops at least) allows users to view both the outline and their current focus position.
2. Otherwise, the dropdown should close as the "Block Navigation" does, so that focus on the page element is clearly identifiable.

Relevant standards

- 2.4.7 Focus Visible (Level AA)

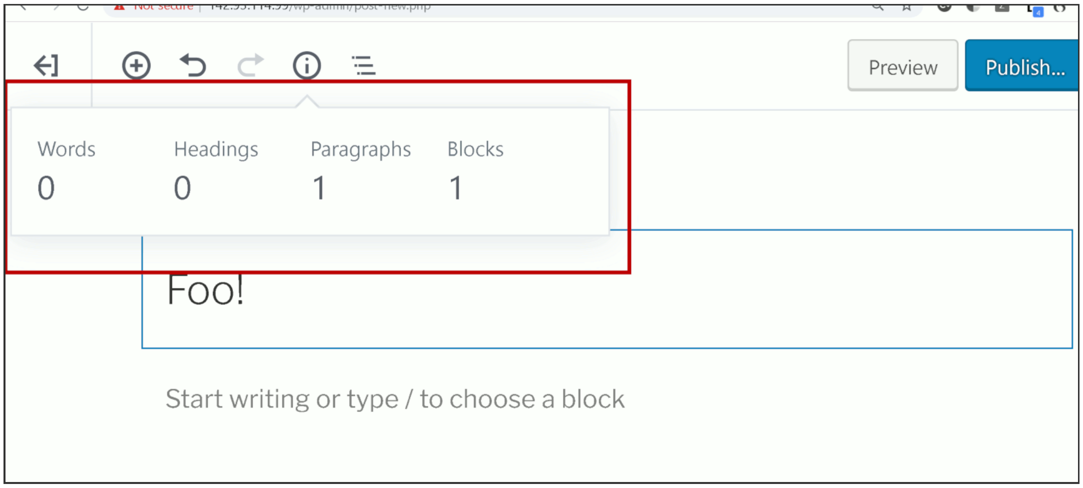
GUT-26 - Focus not visible on Content Structure pane

- **Severity:** Medium
- **Affected Populations:** Low-Vision, Motor Impaired, Cognitively Impaired
- **Platform(s):** All/ Universal
- **Components affected:** Structure and Navigation

Issue description

Keyboard users cannot see their focus position within the Content Structure popover, when focus is on the popover itself, or on the "Document Statistics" section within it.

All focused elements should have a visible and obvious focus indicator, so that sighted keyboard users have a clear indication of which element has focus.



Words	Headings	Paragraphs	Blocks
0	0	1	1

Foo!

Start writing or type / to choose a block

Issue Code

```
<div class="components-popover__content" tabindex="-1">
    <div class="table-of-contents__counts" role="note" aria-label="Document Statistics" tabindex="0">
        ...
    </div>
</div>
...
a, div {
    outline: 0;
}
```

Remediation Guidance

Remove the CSS which negates the focus style on these elements, or override it with specific focus styles on the applicable elements.

Recommended Code

```
components-popover__content {
    /* some outline style */
}
```

Relevant standards

- 2.4.7 Focus Visible (Level AA)

GUT-20 - Buttons are low contrast and focus not visible

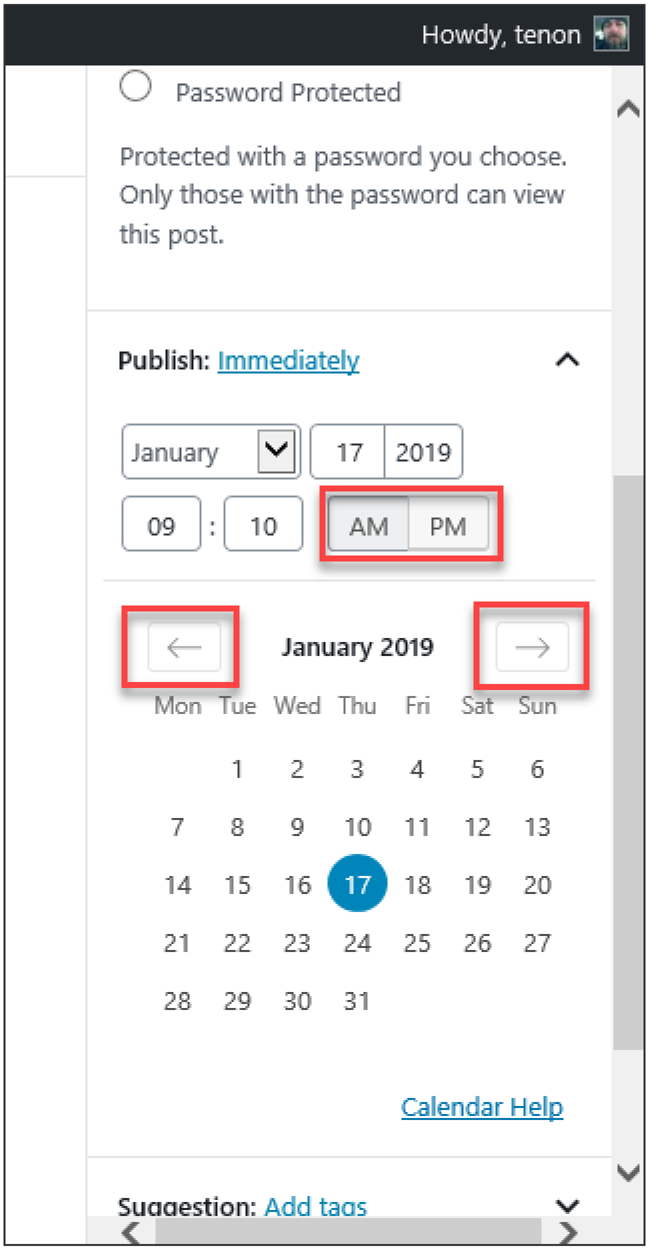
- **Severity:** Medium
- **Affected Populations:** Low-Vision, Motor Impaired, Cognitively Impaired
- **Platform(s):** All/ Universal
- **Components affected:** Document Panel, Publish and Unpublish

Issue description

The focus outline for several buttons within the Publish panel have insufficient color contrast, below the minimum threshold of 3:1 for interface components:

- Date picker month navigation buttons : #c4c4c4 (ash) against #fff (white): **1.74:1**.
- Copy link & view post buttons: #bfe7f3 (light blue) against #fff (white): **1.31:1**.
- AM & PM button when not pressed : #bfe7f3 (light blue) against #fff (white): **1.31:1**.
- AM & PM button when pressed : focus outline cannot be seen

All interactive elements should have a visible and obvious focus indicator, so that sighted keyboard users have a clear indication of which element has focus. Sufficient color contrast showing states is important for users who have low-vision or are color-blind, because states with a low contrast ratio may be difficult or impossible for such users to see.



Howdy, tenon

○ Password Protected

Protected with a password you choose. Only those with the password can view this post.

Publish: Immediately

January | 17 | 2019

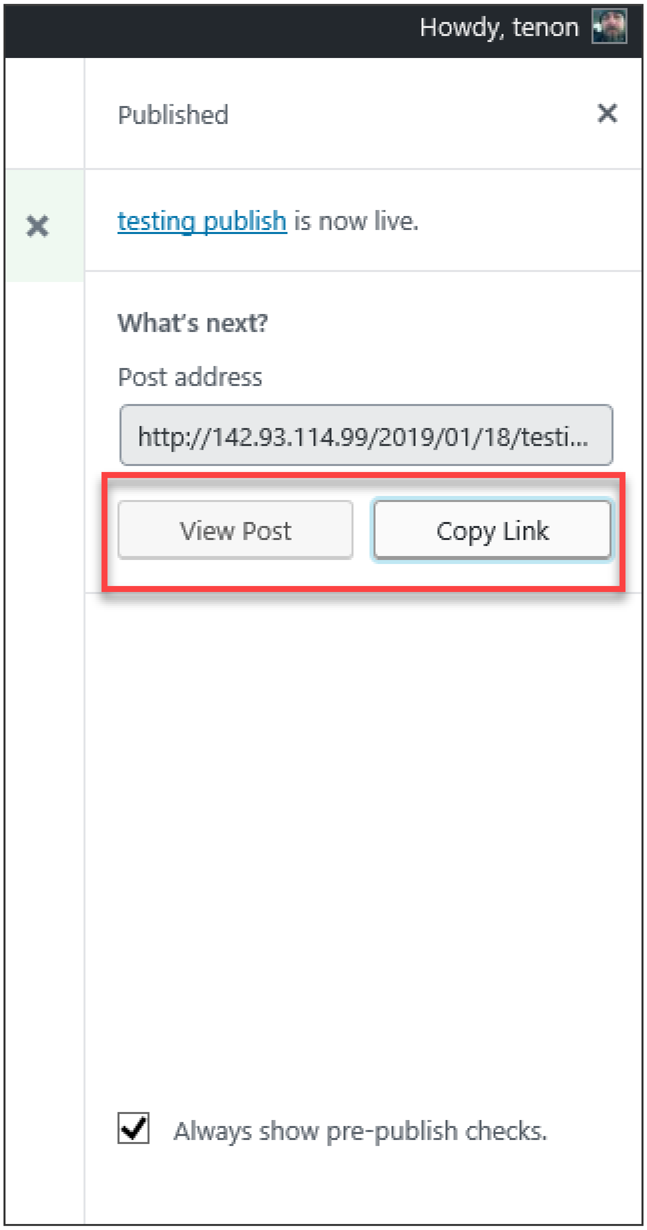
09 : 10 | AM | PM

January 2019

Mon	Tue	Wed	Thu	Fri	Sat	Sun
	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	31			

Calendar Help

Suggestion: Add tags



Howdy, tenon

Published

testing publish is now live.

What's next?

Post address

http://142.93.114.99/2019/01/18/testi...

☑ Always show pre-publish checks.

Issue Code

```
/*          h           i            d         i k   */
/* AM, PM buttons, View Post and Copy Links */
.components-button.is-default:focus:enabled {
    ...
    border-color: #999;
    box-shadow: inset 0 -1px 0 #999, 0 0 0 2px #bfe7f3;
    text-decoration: none;
}

/* date picker month navigation buttons */
.DayPickerNavigation_button__default:focus, .DayPickerNavigation_button__default:hover {
    border: 1px solid #c4c4c4;
    border-bottom-color: ;
}
```

Remediation Guidance

Darken the `border` or `box-shadow` colors used for focus states to achieve a minimum contrast ratio of 3:1 relative to the white backgrounds.

Add transparent outlines to ensure that the focus can still be seen by Windows High Contrast users (see ~~GUT-29~~) and remove the negative `outline-offset` so that the outline doesn't overlap the border.

The combination of `:focus` with `:enabled` seems unsupported for elements which do not have a `disabled` state in most browsers, so use another method to style focus states until browser support has improved.

Recommended Code

```
/* AM, PM buttons, View Post and Copy Links */
.components-button.is-default:focus {
    ...
    box-shadow: inset 0px 0px 0px 1px #00435d;
    outline: 2px solid transparent;
}

/* date picker month navigation buttons */
.DayPickerNavigation_button__default:focus, .DayPickerNavigation_button__default:hover {
    border: 1px solid #8E8E8E;
    outline: 3px solid transparent;
}
```

Relevant standards

- 1.4.1 Use of Color (Level A)
- 1.4.11 Non-text Contrast (Level AA)
- 2.4.7 Focus Visible (Level AA)

Related Issues

- Related to "States are not visible in High Contrast views"
- Related to "Focus not visible on Preview button, and low contrast on various buttons"

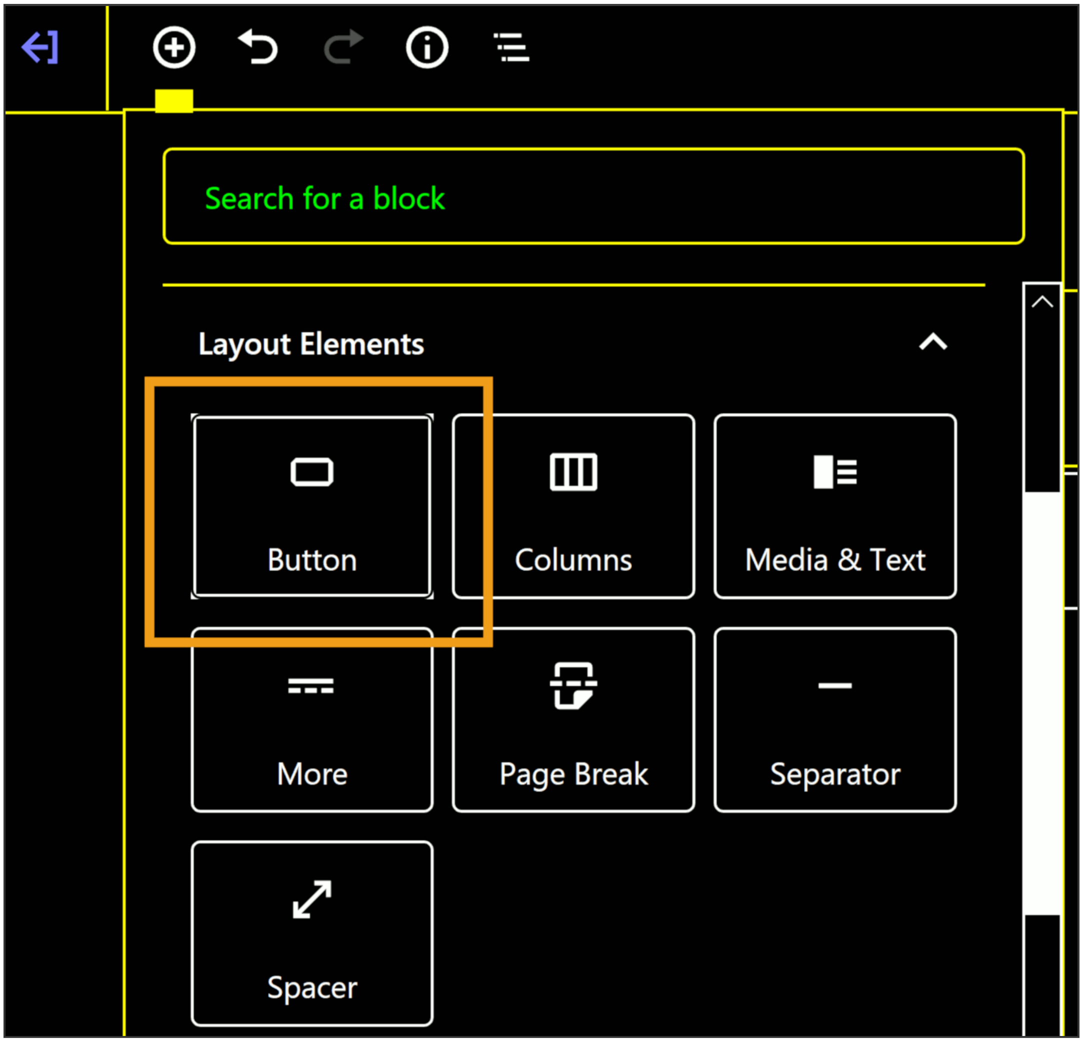
GUT-17 - Focus not visible on block types in High Contrast modes

- **Severity:** Medium
- **Affected Populations:** Low-Vision, Motor Impaired
- **Platform(s):** Windows - High Contrast
- **Components affected:** Add Block

Issue description

When users of Windows High Contrast mode navigate between potential block types, an outline is added to the currently-focussed block. However this outline is pulled inwards with an `outline-offset` property, which sets it directly over the border. This makes it difficult to tell which element is focussed in browsers which support that property (currently all except IE).

All interactive elements should have a visible and obvious focus indicator even for high contrast users, so that sighted keyboard users have a clear indication of which element has focus.



Issue Code

```
.editor-block-types-list__item:not(:disabled).is-active, .editor-block-types-list__item:not(:disabled):active,
.editor-block-types-list__item:not(:disabled):focus {

    outline: 0;

    ...

    outline: 2px solid transparent;

    outline-offset: -2px;

}
```

Remediation Guidance

Remove the `outline-offset` rule and allow the outline to sit outside the border.

Recommended Code

```
.editor-block-types-list__item:not(:disabled).is-active, .editor-block-types-list__item:not(:disabled):active,
.editor-block-types-list__item:not(:disabled):focus {

    ...

    outline: 2px solid transparent;

}
```

Relevant standards

- 2.4.7 Focus Visible (Level AA)

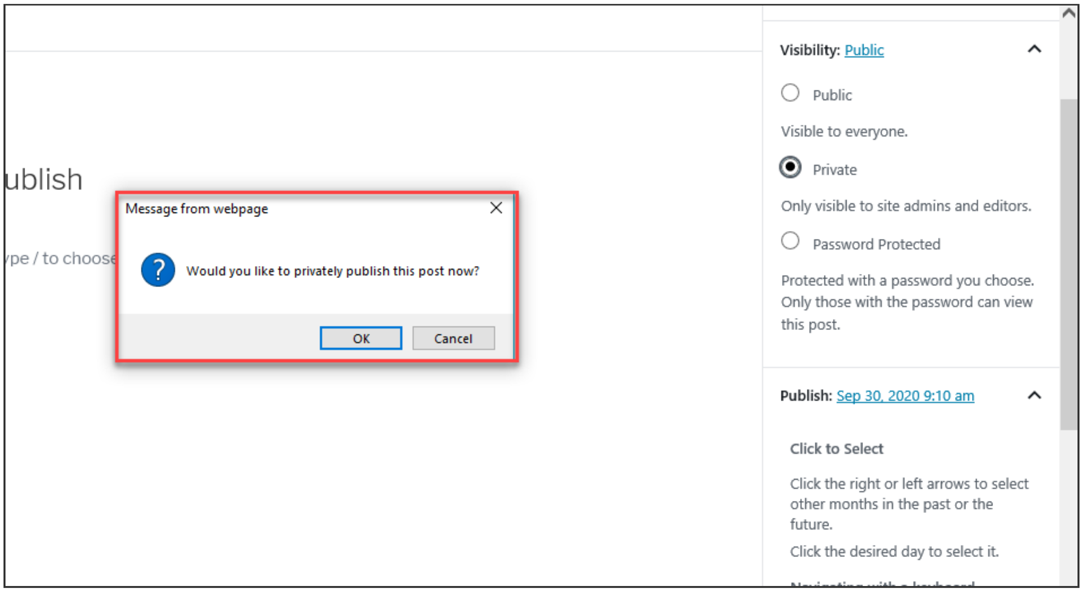
GUT-16 - Radio button automatically changes the context on user focus

- **Severity:** High
- **Affected Populations:** Blind, Low-Vision, Motor Impaired, Cognitively Impaired
- **Platform(s):** All/ Universal
- **Components affected:** Document Panel, Publish and Unpublish

Issue description

Under the "Visibility" section of the Publish panel, there are three radio buttons named "Public", "Private" and "Password". When the user selects the "Private" option, an alert appears asking if they want to publish the post privately, and clicking "Yes" then publishes the post immediately. This behaviour is not consistent with the expected behaviour of radio controls, and is not consistent with the behavior of the other two radios in that group.

Additionally, if the user clicks "No" in the alert, the focus is shifted to the "Public" radio control, rather than shifting back to the "Private" radio as would be expected.



ublish

ype / to choose

Message from webpage ✕

? Would you like to privately publish this post now?

OK Cancel

Visibility: Public

○ Public

Visible to everyone.

● Private

Only visible to site admins and editors.

○ Password Protected

Protected with a password you choose. Only those with the password can view this post.

Publish: Sep 30, 2020 9:10 am

Click to Select

Click the right or left arrows to select other months in the past or the future.

Click the desired day to select it.

Navigating with a keyboard

Remediation Guidance

Remove the alert-and-publish behaviour from the "Private" radio, so that it behaves like a standard radio control. This will also resolve the focus management issue.

Relevant standards

- 3.2.2 On Input (Level A)
- 3.2.5 Change on Request (Level AAA)

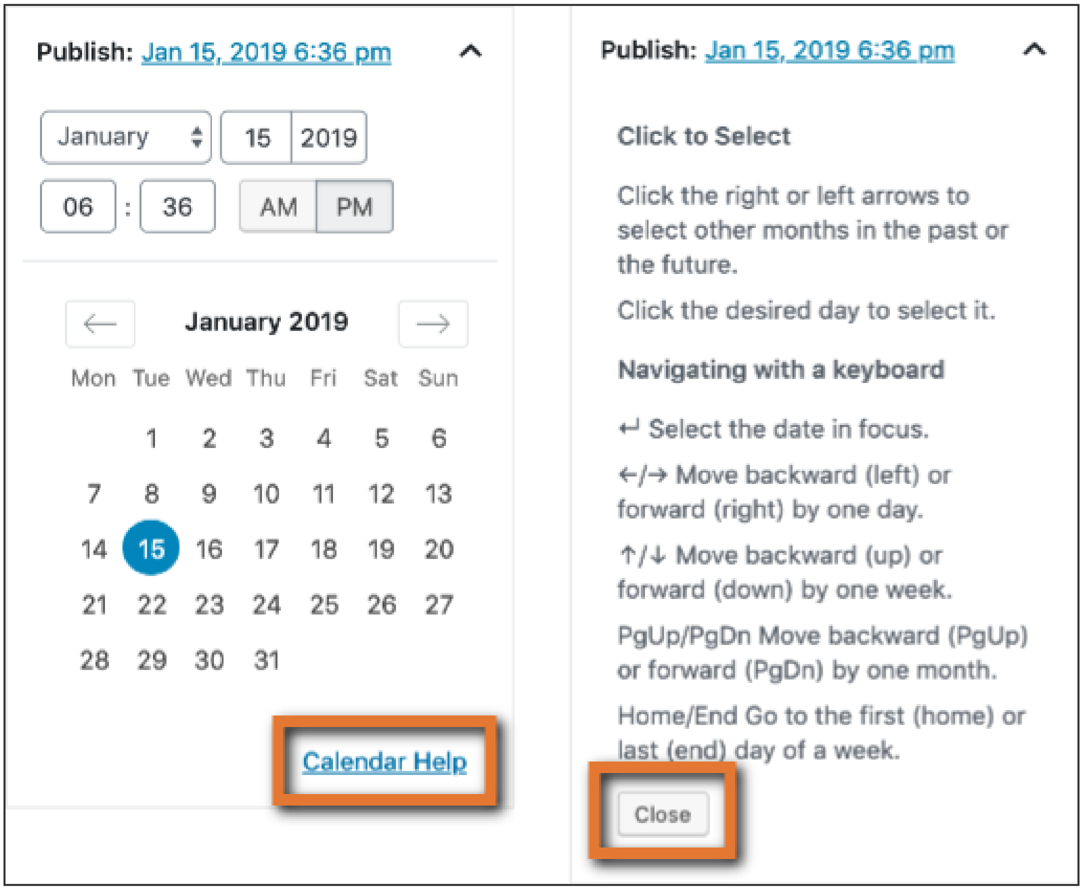
GUT-15 - Keyboard focus is not managed when dynamic content disappears

- **Severity:** Medium
- **Affected Populations:** Blind, Low-Vision, Motor Impaired, Cognitively Impaired
- **Platform(s):** All/ Universal
- **Components affected:** Publish and Unpublish

Issue description

When the "Calendar Help" link or the "Close" button on its popup is clicked, the focus is reset to the top of the page, in all tested browsers except Chrome and chromium.

Keyboard and screen reader users rely on predictable and consistent focus control to navigate the page. When this is absent, users may find it much more difficult or even impossible to navigate using the keyboard.



Publish: Jan 15, 2019 6:36 pm

Publish: Jan 15, 2019 6:36 pm

Click to Select

Click the right or left arrows to select other months in the past or the future.

Click the desired day to select it.

Navigating with a keyboard

↵ Select the date in focus.

←/→ Move backward (left) or forward (right) by one day.

↑/↓ Move backward (up) or forward (down) by one week.

PgUp/PgDn Move backward (PgUp) or forward (PgDn) by one month.

Home/End Go to the first (home) or last (end) day of a week.

Issue Code

```
<!-- when Calendar help is clicked... -->

<button type="button" class="components-button components-datetime__date-help-button is-link">Calendar
Help</button>


<!-- ...focus should be brought to the popup -->

<div class="components-popover__content" tabindex="-1" style="max-height: 187.361px;">

    ...<button type="button" class="...">Close</button>...

</div>


<!-- when the Close button is clicked, focus should move to Month control -->

<div class="components-datetime__time-field components-datetime__time-field-month">

    <select aria-label="Month" class="...">...</select>

</div>
```

Remediation Guidance

When the "Calendar Help" link is clicked, focus should move to the popup which is opened (which has `tabindex="-1"`).

When the "Close" button is clicked, focus should move back to the Calendar Help's first focusable element (the month control).

Relevant standards

- 2.1.1 Keyboard (Level A)

GUT-11 - SVGs are focusable in IE

- **Severity:** Low
- **Affected Populations:** Blind, Low-Vision, Motor Impaired, Cognitively Impaired
- **Platform(s):** Windows - Screen Reader, Windows - ZoomText
- **Components affected:** Output Markup, Publish and Unpublish

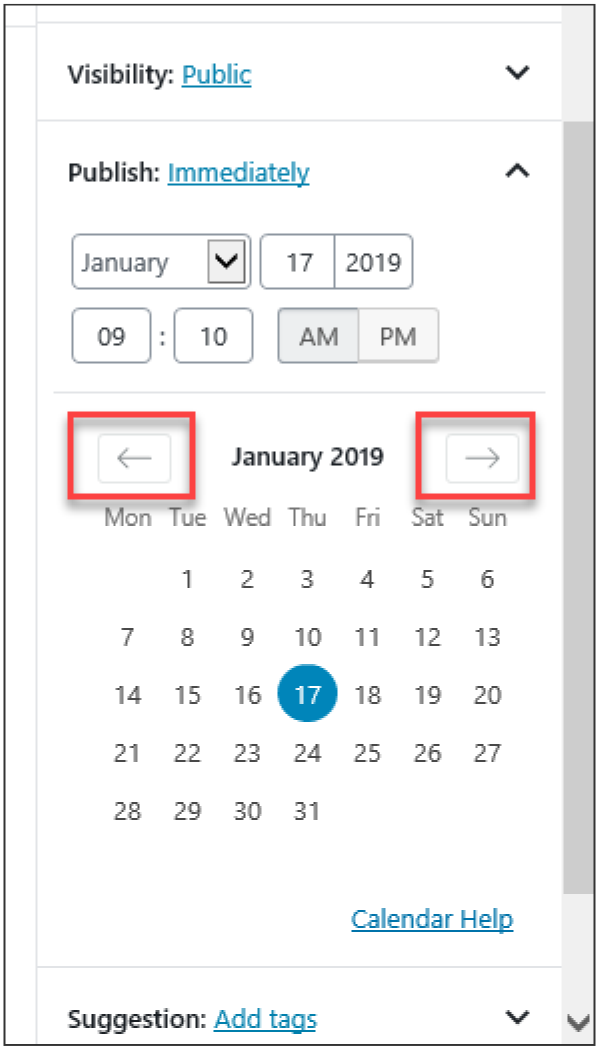
Issue description

The next-month and previous-month buttons within the datepicker contain SVG elements, which are focusable in IE11 because of an anomaly in how IE handles SVG.

This issue also occurs with the "Previous" and "Next" post navigation links on published pages.

This causes additional tab stops for keyboard users. These Tab stops are also not described for assistive technologies, because they have no accessible names.

The SVG's role ("graphic" or "diagram") may be exposed to screen reader users, but when they are used as decorative icons they should not be announced at all.



Visibility: Public

Publish: Immediately

January 17 2019

09 : 10 AM PM

January 2019

Mon	Tue	Wed	Thu	Fri	Sat	Sun
	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	31			

Calendar Help

Suggestion: Add tags

Issue Code

```
<svg xmlns="http://www.w3.org/2000/svg" class="DayPickerNavigation_svg__horizontal
DayPickerNavigation_svg__horizontal_1" viewBox="0 0 1000 1000"><path d="..." /></svg>
```

```
<svg xmlns="http://www.w3.org/2000/svg" class="DayPickerNavigation_svg__horizontal
DayPickerNavigation_svg__horizontal_1" viewBox="0 0 1000 1000"><path d="...></svg>
```

```
<!-- published page-->
<a href="...2018/08/13/hello-world/" rel="prev">
    <span class="screen-reader-text">Previous Post</span>
    ...
    <span class="...">
        <span class="...">
            <svg class="icon icon-arrow-left" aria-hidden="true" role="img">... </svg>
        </span>
        Hello world!
    </span>
</a>
...
<button type="submit" class="search-submit">
    <svg class="icon icon-search" aria-hidden="true" role="img"> ...</svg>
    <span class="screen-reader-text">Search</span>
</button>
```

Remediation Guidance

Add `focusable="false"` and `aria-hidden=true` to the SVG elements, so that they're not focusable nor announced by assistive technologies.

Recommended Code

```
<svg focusable="false" aria-hidden="true" xmlns="http://www.w3.org/2000/svg"
class="DayPickerNavigation_svg__horizontal DayPickerNavigation_svg__horizontal_1" viewBox="0 0 1000 1000">

   <path d="..." />

</svg>
```

```
<svg focusable="false" aria-hidden="true" xmlns="http://www.w3.org/2000/svg"
class="DayPickerNavigation_svg__horizontal DayPickerNavigation_svg__horizontal_1" viewBox="0 0 1000 1000">

   <path d="...>

</svg>
```

```
<!-- published page-->

<a href="...2018/08/13/hello-world/" rel="prev">

    <span class="screen-reader-text">Previous Post</span>

    ...

    <span class="...">

        <span class="...">

            <svg focusable="false" class="icon icon-arrow-left" aria-hidden="true" role="img">... </svg>

        </span>

        Hello world!

    </span>

</a>

...

<button type="submit" class="search-submit">

    <svg focusable="false" class="icon icon-search" aria-hidden="true" role="img"> ...</svg>

    <span class="screen-reader-text">Search</span>

</button>
```

GUT-8 - Toolbar elements are tabbable

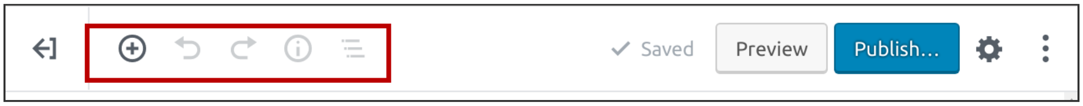
- **Severity:** Medium
- **Affected Populations:** Blind, Motor Impaired, Cognitively Impaired
- **Platform(s):** All/ Universal
- **Components affected:** Editor Top Bar

Issue description

Keyboard-only users who enter the Editor Top Bar must Tab from button to button. Although this is manageable when there's a small number of buttons, the group of buttons is given the role of `toolbar`, whose purpose (besides grouping related controls) is to lessen the number of Tab stops keyboard users must make to navigate past widgets.

Screen reader users may be told they are entering a toolbar and expect to be able to Tab past everything in it. Their experience does not match the promise of the role.

Screen reader users may also miss the toolbar role when meeting the first item (an anchor), because roles such as toolbar are sometimes only announced when users meet items which can trigger their forms/focus mode. Since the anchor's behaviour is semantically different from the buttons' (the buttons deal with the editable content, the anchor simply takes users back), it may make more sense to keep the anchor out of the toolbar group.



Issue Code

```
<div role="region" aria-label="Editor top bar" class="edit-post-header" tabindex="-1">

    <div role="toolbar" aria-orientation="horizontal" class="edit-post-header-toolbar" aria-label="Document
tools">

        <div class="components-toolbar components-toolbar edit-post-fullscreen-mode-close__toolbar">

            <a href="edit.php?post_type=post" aria-label="View Posts" class="..."><svg aria-hidden="true"...>...
</svg></a>

        </div>

    <div>

        <div class="editor-inserter">

            <button type="button" aria-label="Add block" aria-haspopup="true" aria-expanded="false"
class="...">...</button>

        </div>

    </div>

    <button type="button" aria-label="Undo" aria-disabled="false" class="...">...</button>

    <button type="button" aria-label="Redo" aria-disabled="true" class="...">...</button>

    <div class="table-of-contents">

        <button type="button" aria-label="Content structure" aria-expanded="false" aria-disabled="false"
class="...">...</button>

    </div>

    ...

</div>
```

Remediation Guidance

Move the "View Posts..." anchor out of the `role=toolbar` section, so the toolbar only has `<button>` children.

Although ideally it would be a better use of the `toolbar` pattern to implement Arrow-keys only, and not support Tab navigation between items, this pattern may not be familiar to all users. Adding support for the Home and End keys (for jumping to the first and last item in the toolbar) will help here: this will allows users who know the keyboard pattern for ARIA toolbars to efficiently get past the toolbar, while leaving the Tab pattern for users who do not.

Recommended Code

```
<div role="region" aria-label="Editor top bar" class="edit-post-header" tabindex="-1">
    <div class="...">
        <a href="edit.php?post_type=post" aria-label="View Posts" class="..."><svg aria-hidden="true"...>...</svg></a>
    </div>
    <div role="toolbar" aria-orientation="horizontal" class="edit-post-header-toolbar" aria-label="Document tools">
        <div>
            <div class="editor-inserter">
                <button type="button" aria-label="Add block" aria-haspopup="true" aria-expanded="false" class="...">...</button>
            </div>
        </div>
        <button type="button" aria-label="Undo" aria-disabled="false" class="...">...</button>
        <button type="button" aria-label="Redo" aria-disabled="true" class="...">...</button>
        <div class="table-of-contents">
            <button type="button" aria-label="Content structure" aria-expanded="false" aria-disabled="false" class="...">...</button>
        </div>
    </div>
    ...
</div>
```

References

https://www.w3.org/TR/wai-aria-practices-1.1/#toolbar

Relevant standards

- 2.1.1 Keyboard (Level A)

Content Type: Links and Navigation

GUT-28 - Page content can be accessed while dropdowns are open

- **Severity:** Medium
- **Affected Populations:** Blind, Motor Impaired, Cognitively Impaired
- **Platform(s):** Windows - Screen Reader, Windows - Dragon, Mac - VoiceOver, Android - TalkBack, iOS - VoiceOver
- **Components affected:** Structure and Navigation

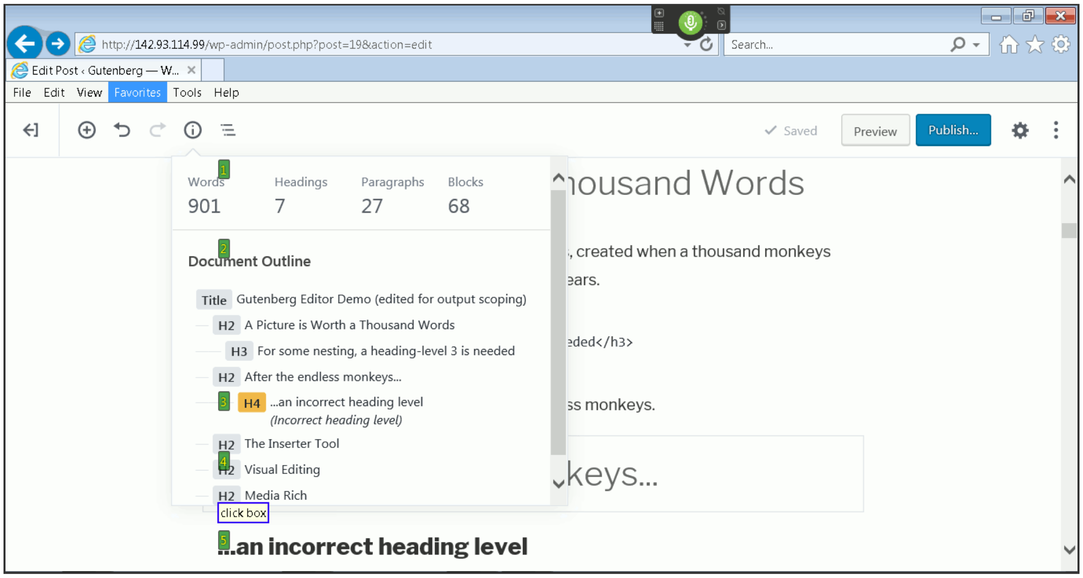
Issue description

Sighted keyboard users who open either the Content Structure or Navigation Block dropdowns find their focus trapped within these dropdowns until they either select an item or close the dropdown.

This suggests the idea is to treat these dropdowns similar to modal dialogs, in that users are not meant to move out of the dropdowns in any linear fashion. However users of assistive technologies such as screen readers and speech recognition can find themselves interacting with elements outside the dropdowns unexpectedly.

Speech recognition users may unintentionally activate elements on the page which are hidden by the open dropdowns, as these are still available to speech recognition.

Screen reader users can browse their way out of the dropdowns without realising they have left, and they end up in the page (and not, perhaps as expected, on the next item in the Editor Top Bar). Because focus reaches an otherwise-hidden "Open Publish Panel" button, a user attempting to move backwards back into the dropdown content, especially by Shift-Tab, find themselves instead moving backwards through the entire editable document, and not back to the dropdown. This interferes with expected navigation techniques.


http://142.93.114.99/wp-admin/post.php?post=19&action=edit
Search...
Edit Post ‹ Gutenberg — W...
File
Edit
View
Favorites
Tools
Help
Saved
Preview
Publish...
Words
901
Headings
7
Paragraphs
27
Blocks
68
1
2
Document Outline
Title
Gutenberg Editor Demo (edited for output scoping)
H2
A Picture is Worth a Thousand Words
H3
For some nesting, a heading-level 3 is needed
H2
After the endless monkeys...
3
H4
...an incorrect heading level
(Incorrect heading level)
H2
The Inserter Tool
4
H2
Visual Editing
H2
Media Rich
click box
5
...an incorrect heading level
housand Words
, created when a thousand monkeys
ears.
eded</h3>
ss monkeys.
keys...

Remediation Guidance

The ideal setup would be to have the dropdowns in source order with the triggering button and the rest of the Editor Top Bar, and allow users to Tab out of the menu straight to the next item in the Editor Top Bar.

For non-Tab navigation in assistive technologies, this will allow users who have navigated away to simply go back to where they expect to be. Then anything which moves the keyboard focus out of the dropdown (such as a voice action, or Tab navigation away from it) should close the dropdown.

Relevant standards

- 1.3.3 Sensory Characteristics (Level A)

Content Type: Name, State, Role, and Value

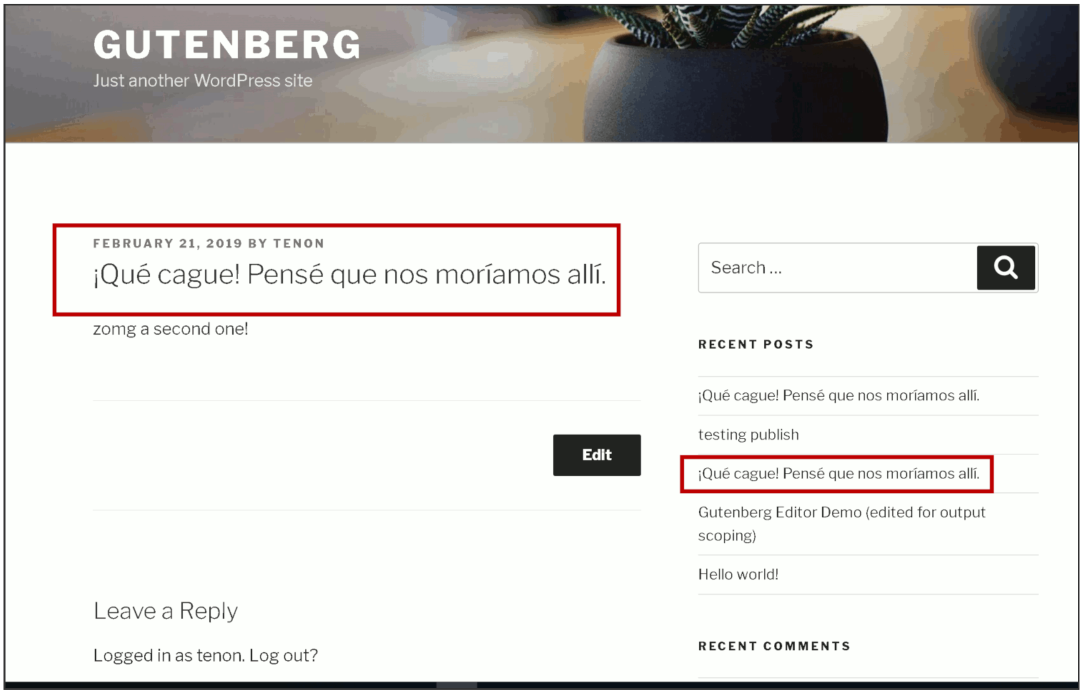
GUT-105 - Current-page link is not programmatically labelled as such

- **Severity:** Medium
- **Affected Populations:** Low-Vision, Cognitively Impaired
- **Platform(s):** All/ Universal
- **Components affected:** Output Markup

Issue description

When a page is created and users view it, its name appears in the sidebar under "Recent Posts", however there is no visual or semantic indication that this link relates to the current page.

Since it's possible for authors to create multiple articles with the same name, the lack of indication may cause confusion for users with cognitive disabilities and screen reader users.



GUTENBERG

Just another WordPress site

FEBRUARY 21, 2019 BY TENON

¡Qué cague! Pensé que nos moríamos allí.

zomg a second one!

Edit

Leave a Reply

Logged in as tenon. Log out?

Search ...

RECENT POSTS

¡Qué cague! Pensé que nos moríamos allí.

testing publish

¡Qué cague! Pensé que nos moríamos allí.

Gutenberg Editor Demo (edited for output scoping)

Hello world!

RECENT COMMENTS

Issue Code

```
<li>
    <a href="...">...</a>
</li>
```

Remediation Guidance

Whenever the current page is reflected in a menu on that page, add `aria-current="page"` to the applicable link.

Use the `aria-current` attribute as a hook to add a distinctive visual style to separate it from other links.

Recommended Code

```
<li>
    <a href="..." aria-current="page">...</a>
</li>
...
widget_recent_entries a[aria-current] {
    /* styles */
}
```

Relevant standards

- 2.4.4 Link Purpose (In Context) (Level A)
- 2.4.8 Location (Level AAA)

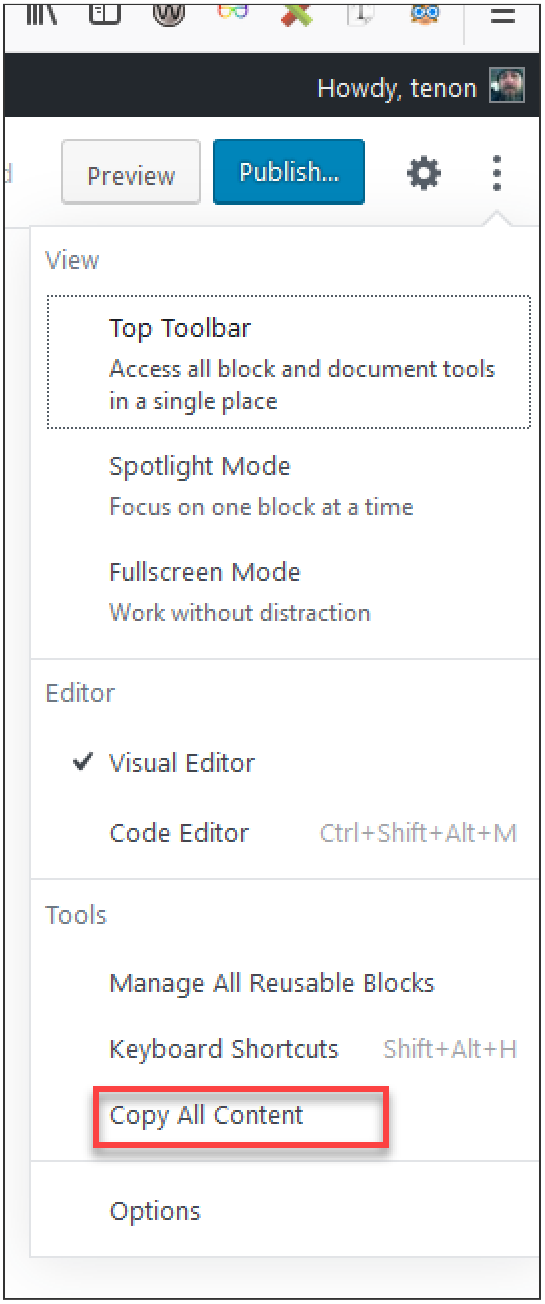
GUT-67 - Missing role on menu item

- **Severity:** Low
- **Affected Populations:** Blind, Low-Vision, Cognitively Impaired
- **Platform(s):** Windows - Screen Reader, Mac - VoiceOver, Android - TalkBack, iOS - VoiceOver
- **Components affected:** Tools and Options

Issue description

The "Copy All Content" menu item is not announced to screen readers as a menu item, it's announced as a button, because it lacks the "menuitem" role that the other items have. This also affects the way the menu as a whole is described, since the announced number of items does not include this button.

Assistive technologies rely on role information to understand the purpose of content, and if this information is incomplete or inaccurate it may create a confusing experience for users.


Howdy, tenon
Preview
Publish...
View
Top Toolbar
Access all block and document tools in a single place
Spotlight Mode
Focus on one block at a time
Fullscreen Mode
Work without distraction
Editor
✓ Visual Editor
Code Editor Ctrl+Shift+Alt+M
Tools
Manage All Reusable Blocks
Keyboard Shortcuts Shift+Alt+H
Copy All Content
Options

Issue Code

```
<button class="components-button components-clipboard-button components-menu-item__button" type="button">Copy All
Content</button>
```

Remediation Guidance

Add `role="menuitem"` to the `<button>` element, consistent with the other items in the menu.

Recommended Code

```
<button class="components-button components-clipboard-button components-menu-item__button" role="menuitem"
type="button">Copy All Content</button>
```

Relevant standards

- 4.1.2 Name, Role, Value (Level A)

GUT-18 - Unnecessary use of aria-label

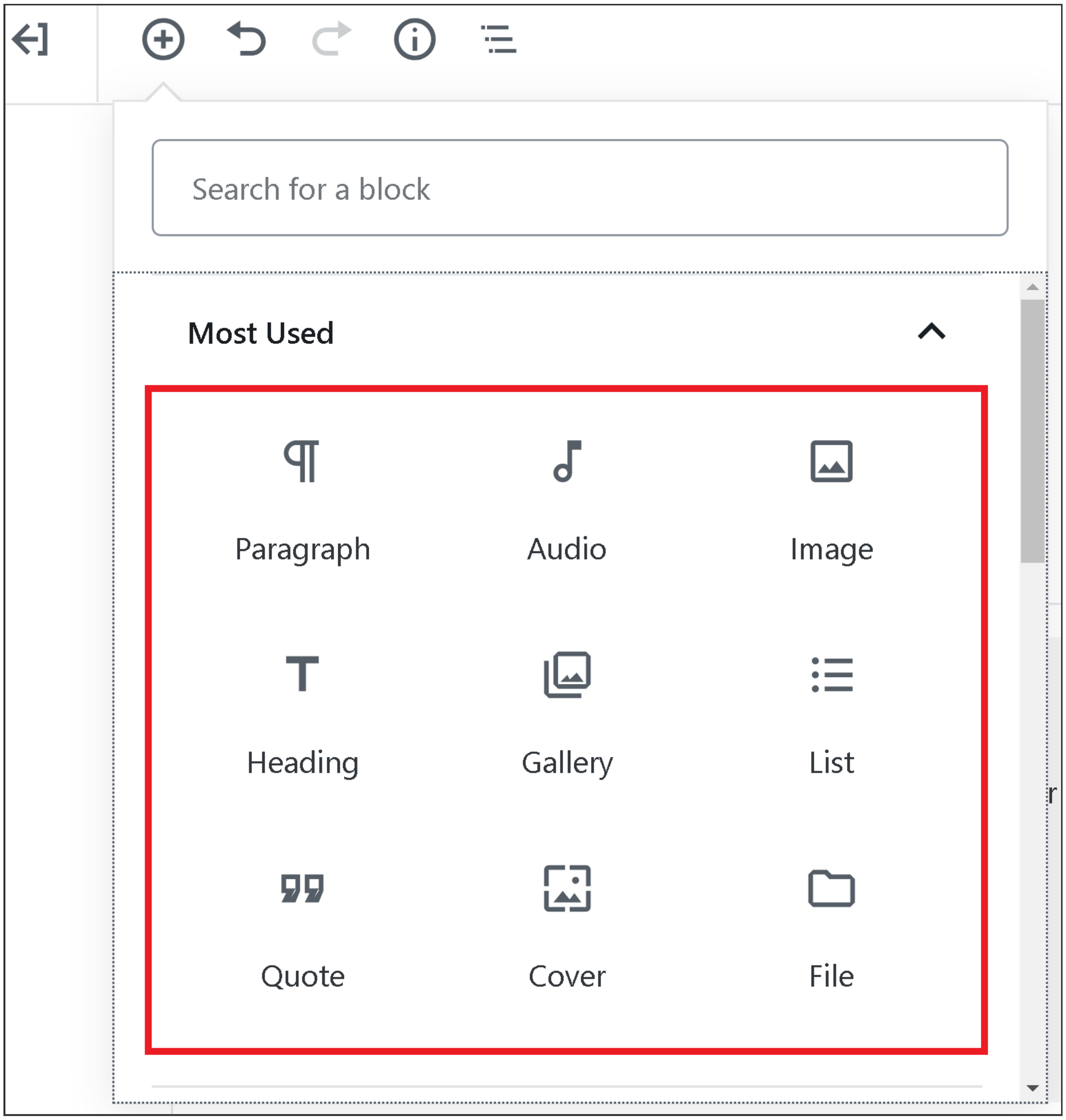
- **Severity:** Low
- **Affected Populations:** Blind, Low-Vision, Cognitively Impaired
- **Platform(s):** Windows - Screen Reader, Windows - Dragon, Mac - VoiceOver, Android - TalkBack, iOS - VoiceOver
- **Components affected:** Add Block

Issue description

The content inside the dropdown/popup of the "Add Block" button uses redundant `aria-label` attributes on the block type buttons. This text is also already visible even in the mobile view, and the SVGs are correctly hidden from assistive technology, therefore the `aria-label` text is not needed.

It appears the normally-unnecessary `role=list` on the unordered list is essential to making Firefox expose the list semantics.

While redundant labels do not affect end-users, unnecessary `aria-label` attributes create the possibility of mismatches between visible labels and accessible names when content is updated.



Issue Code

```
<button class="..." aria-label="Paragraph">
    <span class="...">
        <span class="...">
            <svg aria-hidden="true"...></svg>
        </span>
    </span>
    <span class="editor-block-types-list__item-title">Paragraph</span></button>
```

Remediation Guidance

Remove the `aria-label` from the button and allow the visible text inside to label the button.

This prevents future potential mismatches between an invisible and a visible name.

Recommended Code

```
<button class="...">
    <span class="...">
        <span class="...">
            <svg aria-hidden="true"...></svg>
        </span>
    </span>
    <span class="editor-block-types-list__item-title">Paragraph</span></button>
```

Content Type: Other

GUT-114 - Users are not warned against autoplaying videos

- **Severity:** Low
- **Affected Populations:** Blind, Low-Vision, Motor Impaired, Cognitively Impaired
- **Platform(s):** All/ Universal
- **Components affected:** Block Types, Output Markup

Issue description

Authors who create Video or Audio blocks can, in the Block panel, set the media to autoplay.

Autoplaying video can interfere with users of text-to-speech software such as screen readers and screen magnifiers, as well as make focussing on content difficult for users with cognitive disabilities.

Remediation Guidance

When users are adding video or audio content and select the "Autoplay" checkbox in the Block panel, add a warning that informs users of some of the accessibility issues with autoplay, similar to the color contrast warnings shown when users choose low-contrast color combinations.

GUT-109 - Blocks are difficult to edit with speech recognition

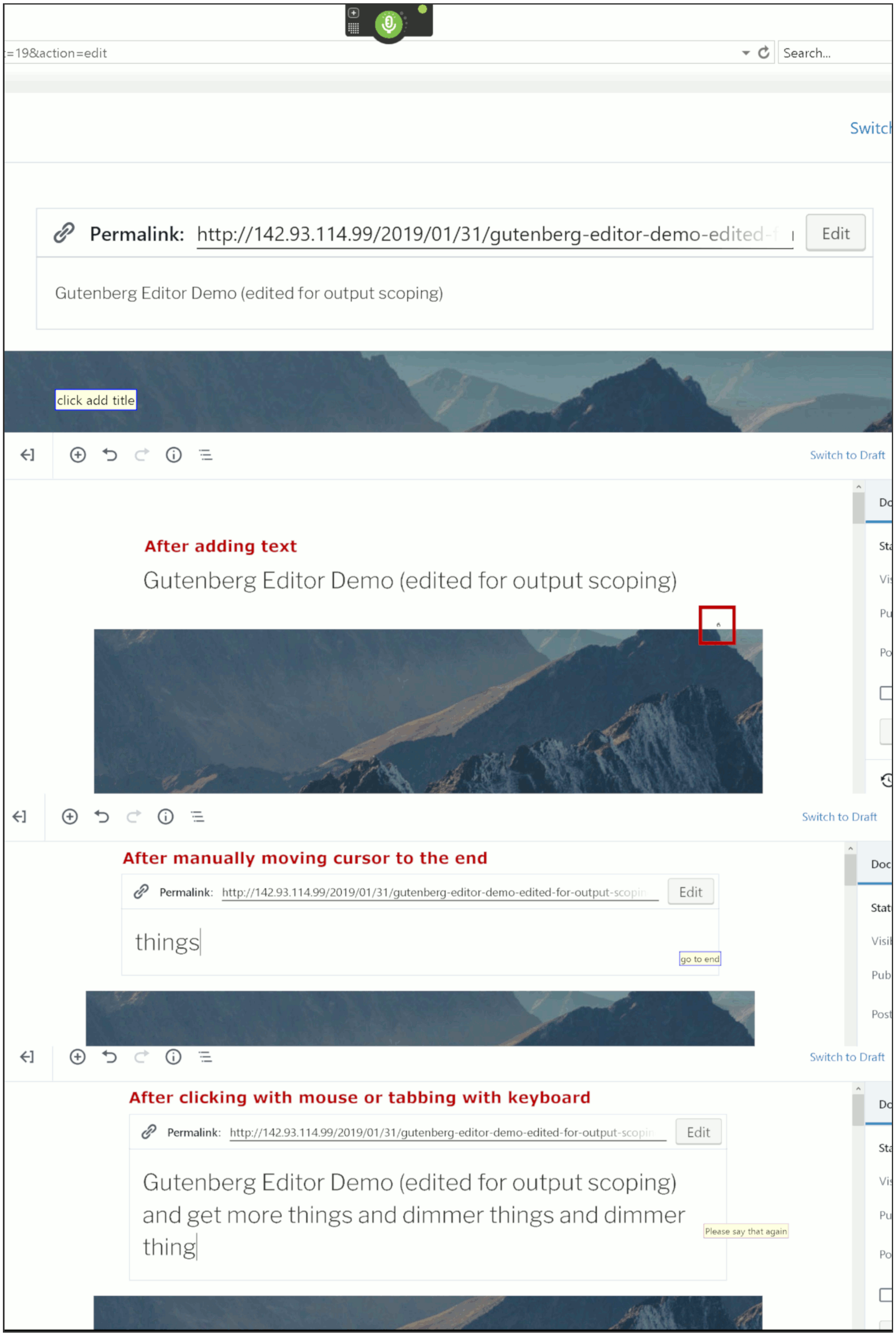
- **Severity:** Medium
- **Affected Populations:** Motor Impaired
- **Platform(s):** Windows - Dragon
- **Components affected:** Block Editing, Title Block

Issue description

Users of speech recognition software such as Dragon NaturallySpeaking, who are attempting to edit blocks, have difficulty bringing their focus into the blocks. The Title block in particular does not retain added text when users attempt to move focus elsewhere (for example, by saying "press tab").

Pre-filled Title blocks which have text added to them do not seem to adjust their size when dictation adds more text. This makes the additional text not visible to users unless they manually move their cursor to the end. Only by switching to keyboard or mouse does the Title block adjust its size to show all current text.

Currently, the only way to retain added text in the Title block, is to speak keyboard commands for moving focus to something else.



Gutenberg Editor Demo (edited for output scoping)

After adding text

Gutenberg Editor Demo (edited for output scoping)

After manually moving cursor to the end

Permalink: http://142.93.114.99/2019/01/31/gutenberg-editor-demo-edited-for-output-scopin

things

After clicking with mouse or tabbing with keyboard

Permalink: http://142.93.114.99/2019/01/31/gutenberg-editor-demo-edited-for-output-scopin

Gutenberg Editor Demo (edited for output scoping) and get more things and dimmer things and dimmer thing

Remediation Guidance

Speech recognition users should be able to use speakable keyboard commands such as "press tab" and get the same results as a keyboard user pressing the Tab key (i.e. where dictated text remains, just as typed text would).

When focus is moved into a block via speech commands, the block should adjust its size to ensure that added text is visible.

References

https://github.com/WordPress/gutenberg/issues/5468

Relevant standards

-

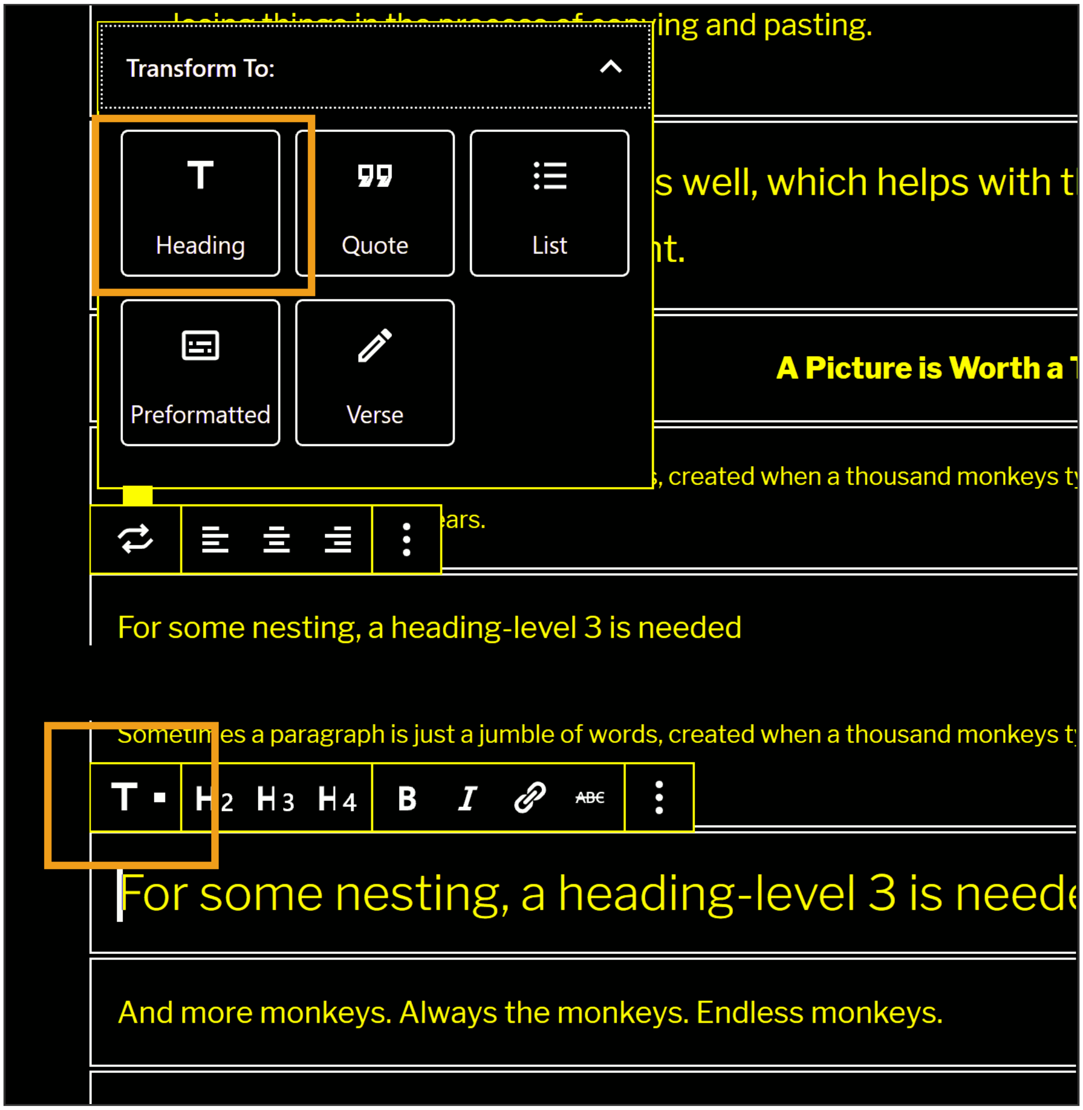
GUT-102 - Heading block uses a counter-intuitive icon

- **Severity:** Low
- **Affected Populations:** Low-Vision, Cognitively Impaired
- **Platform(s):** All/ Universal
- **Components affected:** Block Editing

Issue description

The "Block Type" menu, and block editor type button, use icons for each block, and in most cases these are good representations of the block. However the "Heading" block type uses the letter "T", which is potentially confusing, since "Heading" doesn't start with "T".

Users with cognitive disabilities, especially those who are more icon-reliant, may mistake the heading type for something else (such as "Table" or "Text"), especially in cases where only the icon is visible (such as in the block editor button, or when the "Block Type" menu list is scrolled so that only the icon is visible).



For some nesting, a heading-level 3 is needed

Sometimes a paragraph is just a jumble of words, created when a thousand monkeys t

For some nesting, a heading-level 3 is neede

And more monkeys. Always the monkeys. Endless monkeys.

Remediation Guidance

Use the letter "H" as the icon for the Heading block type. This also matches the other H icons used for setting the level (H2, H3, etc.).

Relevant standards

- 1.3.3 Sensory Characteristics (Level A)

GUT-88 - Image cropping requires mouse and keyboard

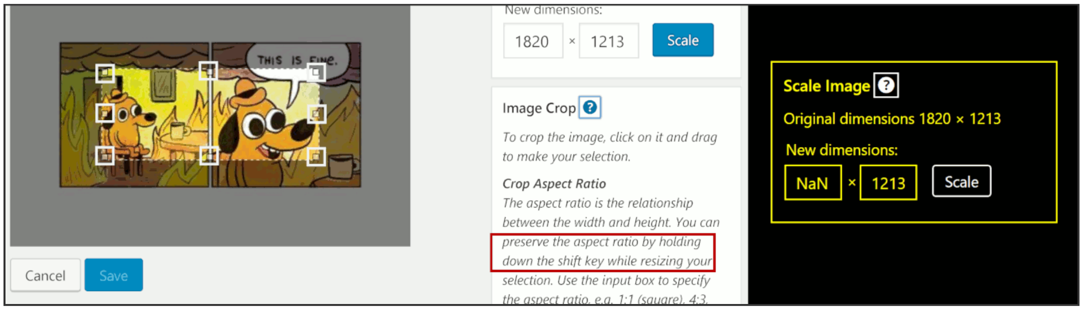
- **Severity:** Medium
- **Affected Populations:** Blind, Low-Vision, Motor Impaired, Cognitively Impaired
- **Platform(s):** All/ Universal
- **Components affected:** Edit Media

Issue description

On the Edit Media page, the functionality to crop images relies on a combination of mouse and keyboard interaction.

Although touch can be used to create an initial selection, the handles to adjust the size are only available to mouse users. The instructions in the cropped section also mention that the Shift key can be used to maintain aspect ratio while cropping, however this assumes that users can manipulate both mouse and keyboard at the same time. (And while testing, this functionality did not appear to work anyway; aspect ratio was only maintained after explicitly setting a ratio in the text inputs.)

Users with disabilities may not be able to use a mouse, or may not be able to use a mouse and keyboard at the same time. Making tasks such as this input-agnostic will allow users more freedom and ability to perform the task.



To crop the image, click on it and drag to make your selection.

Crop Aspect Ratio

The aspect ratio is the relationship between the width and height. You can preserve the aspect ratio by holding down the shift key while resizing your selection. Use the input box to specify the aspect ratio, e.g. 1:1 (square), 4:3,

Remediation Guidance

Firstly, it would be beneficial to low-vision users and users with cognitive disabilities if the crop/aspect ratio and selection inputs were closers to each other (both in layout, and in source order). It would then be easier for such users to move between sections and adjust the input values after making an initial imperfect selection. It would also be more logical to place these fields after the image editing area, rather than before it.

The cropping feature should function for any input mode, i.e. it should work from only the keyboard, only the mouse, only touch, or any combination of those.

For combined functions such as maintaining an aspect ratio while resizing the crop region, have a separate checkbox which users can select to maintain (or ignore) aspect ratio. The current Shift key functionality can be retained, but only as a shortcut for users who can use mouse and keyboard at the same time, it should not be the only way of doing this.

Relevant standards

- 2.1.1 Keyboard (Level A)
-

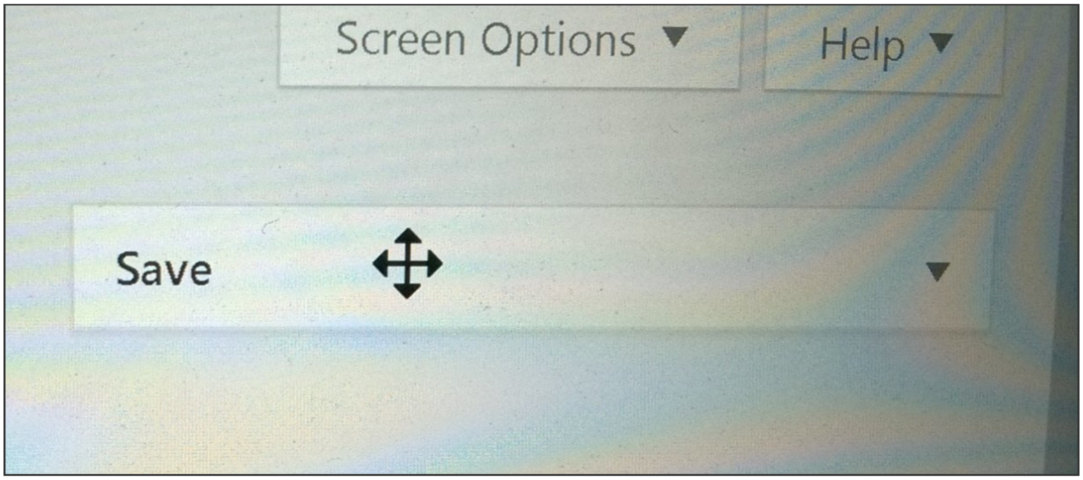
GUT-78 - Cursor suggests functionality even when functionality is not present

- **Severity:** Medium
- **Affected Populations:** Low-Vision, Cognitively Impaired
- **Platform(s):** All/ Universal
- **Components affected:** Edit Media

Issue description

Users who mouse over the "Save" heading on the Edit Media page see a "move" cursor, which suggests to users that the "Save" section can be dragged and dropped to somewhere else. However, this is only the case if additional boxes are shown, re-ordering is not possible if "Save" is the only visible box (although the element can be dragged, there is nowhere else to drop it).

The use of consistent and relevant cursors may be important for users who have a cognitive disability, since cursors give a visual clue as to an element's functionality. Using the move cursor for elements which cannot be moved may be confusing or counter-intuitive for users.



Issue Code

```
.js .postbox .hndle, .js .widget .widget-top {
    cursor: move;
}
```

Remediation Guidance

Only allow the `move` cursor to appear to users when re-ordering is functionally possible.

Ideally, the drag and drop functionality should not be available at all when only one box is present (i.e. it should not even be possible to drag that one box), however this may be unrealistic to fix, and therefore the solution of removing the cursor style will go a long way towards matching user expectations.

Related Issues

- Related to "Options vanish at higher zoom levels"

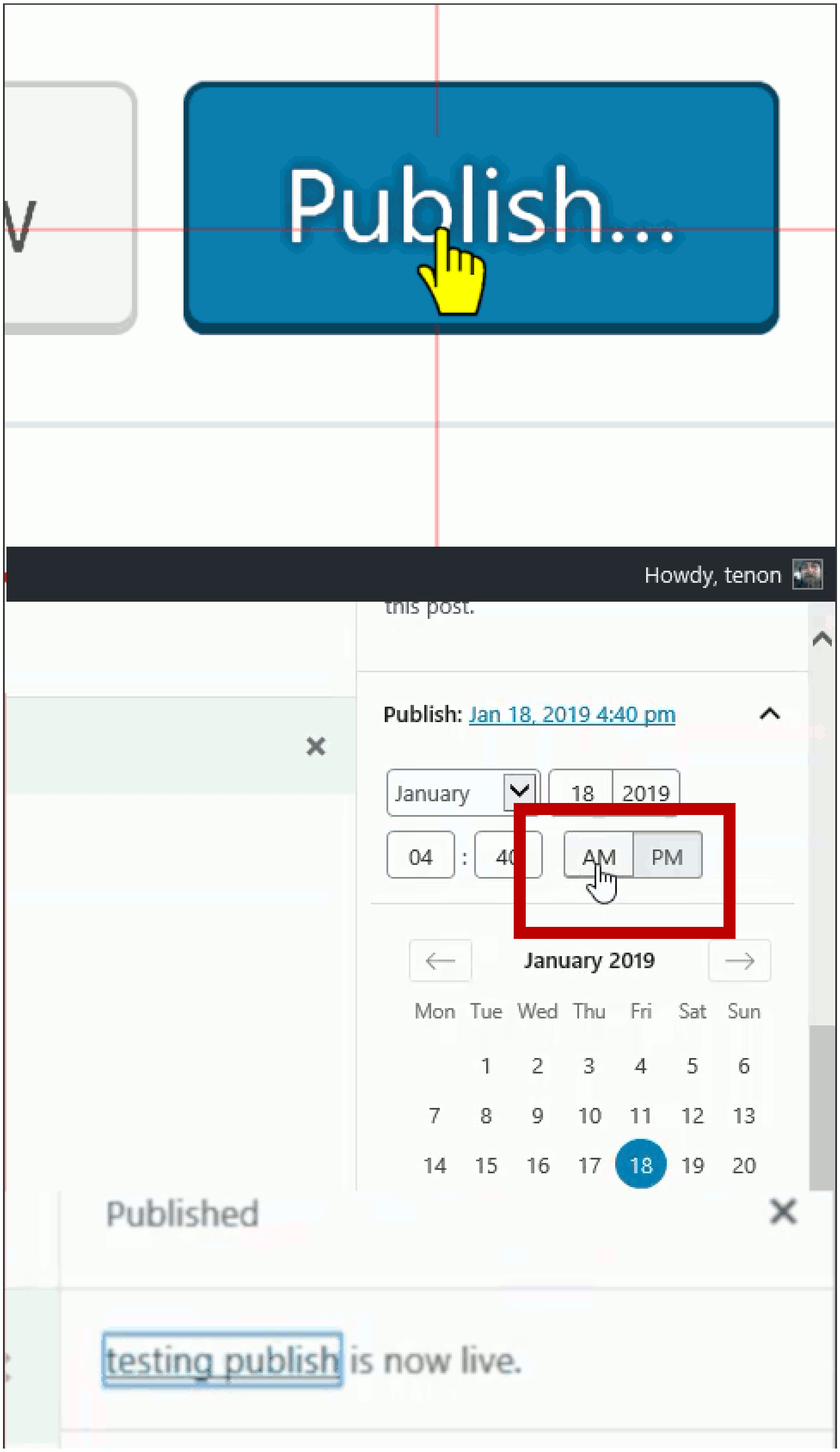
GUT-9 - Incorrect cursor used on buttons

- **Severity:** Low
- **Affected Populations:** Low-Vision, Cognitively Impaired
- **Platform(s):** All/ Universal
- **Components affected:** Global

Issue description

Buttons throughout the application use the pointer cursor instead of the default.

The use of consistent and relevant cursors may be important for users who have a cognitive disability, since cursors give a visual clue as to an element's functionality. Using the pointer cursor for elements which do not typically show that cursor may be confusing or counter-intuitive for users.



Publish: Jan 18, 2019 4:40 pm

January 18 2019

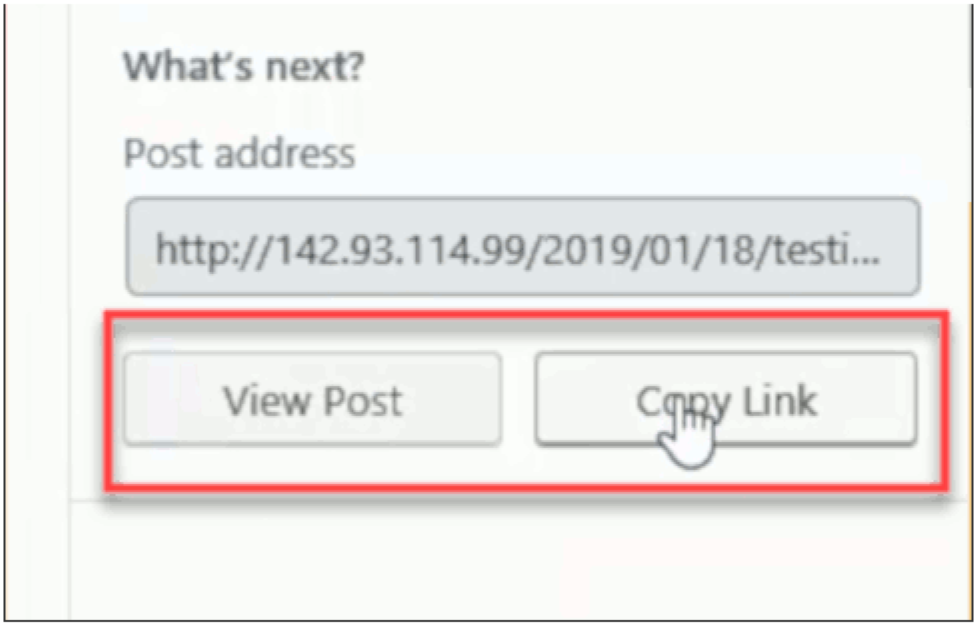
04 : 40

January 2019

Mon	Tue	Wed	Thu	Fri	Sat	Sun
	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20

Published

testing publish is now live.



Issue Code

```
label {
    cursor: pointer;
}

.wp-core-ui .button,
.wp-core-ui .button-primary,
.wp-core-ui .button-secondary {
    cursor: pointer;
    ...
}

input[type=checkbox],
input[type=radio] {
    cursor: pointer;
    ...
}

.components-button {
   cursor: pointer;
    ...
}
```

Remediation Guidance

Only controls that navigate users to new pages/views should have the cursor set to `pointer`, while controls performing push-button type actions use `default`.

Recommended Code

```
label {
    cursor: default;
}

.wp-core-ui .button,
.wp-core-ui .button-primary,
.wp-core-ui .button-secondary {
    cursor: default;
    ...
}

input[type=checkbox],
input[type=radio] {
    cursor: default;
    ...
}

.components-button {
   cursor: default;
    ...
}
```

References

https://www.w3.org/TR/CSS22/ui.html#cursor-props

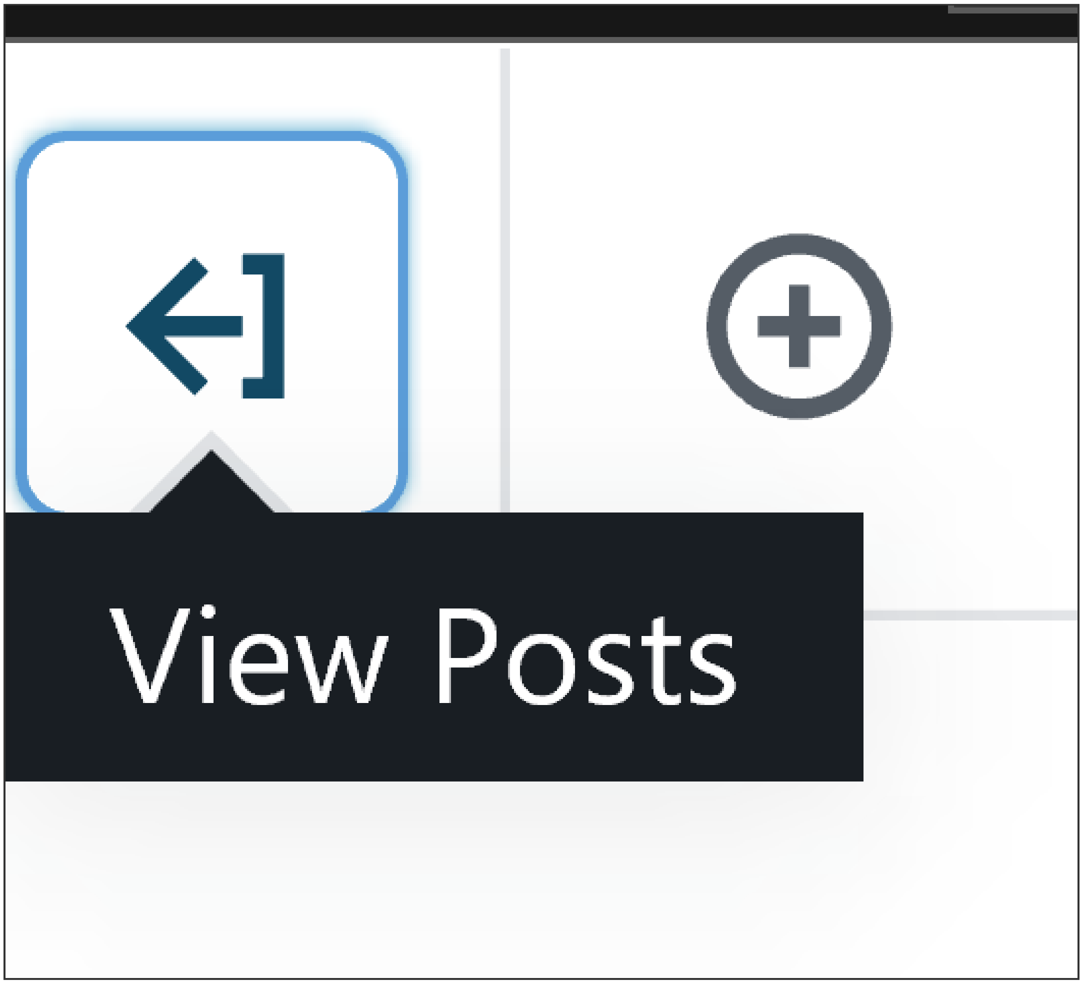
GUT-6 - Tooltips cannot be dismissed without moving the pointer or focus

- **Severity:** Low
- **Affected Populations:** Low-Vision, Motor Impaired, Cognitively Impaired
- **Platform(s):** All/ Universal
- **Components affected:** Add Block, Block Editing, Block Options, Block Panel, Block Types, Editor Top Bar, Structure and Navigation

Issue description

Users who trigger tooltips in the components listed, using pointer hover or keyboard focus, cannot dismiss the tooltip without moving their pointer or keyboard focus.

Users of high browser zoom or magnification software may only see a small area of the page at any one time. Content which appears on hover or focus may cover the content underneath, but moving the pointer or focus (in order to dismiss the tooltip) also moves the visible area.



Remediation Guidance

Add a method for users to hide the tooltips without moving the pointer or keyboard focus. The simplest way to achieve this is to add an Escape key handler which closes the currently-visible tooltip.

Relevant standards

- 1.4.13 Content on Hover or Focus (Level AA)

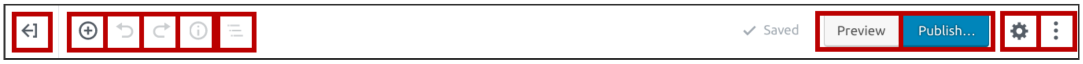
GUT-5 - Buttons are smaller than the minimum target size

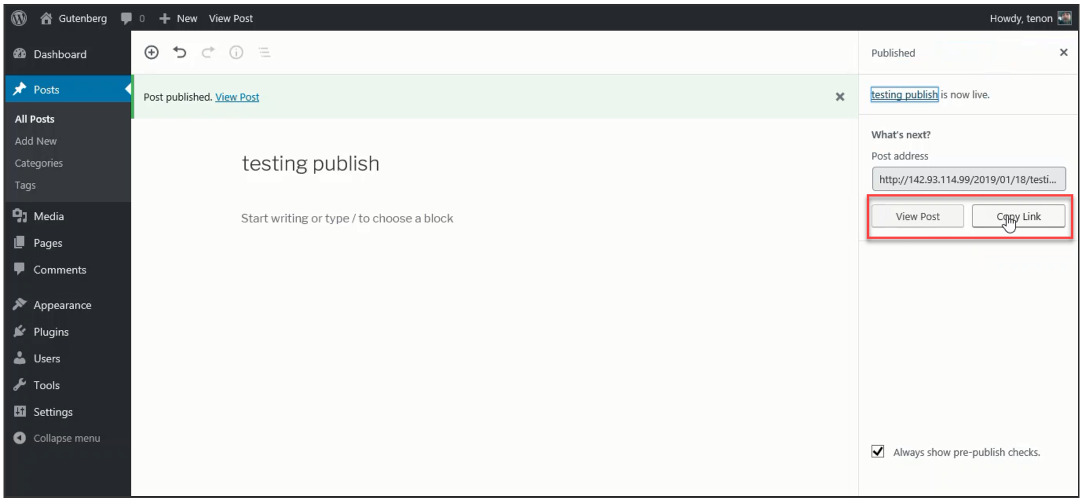
- **Severity:** Low
- **Affected Populations:** Low-Vision, Motor Impaired
- **Platform(s):** All/ Universal
- **Components affected:** Global

Issue description

Many of the buttons in the site are smaller than the recommended 44px by 44px. This is the case for all buttons in the components tested where no minimum size has been specified.

All actionable elements (apart from inline links and native widgets) should have a minimum size of 44px square, because smaller targets may be difficult for users who have a motor impairment and can't use a pointer accurately. Small targets are also more difficult for users on touch devices, since fingers are not very accurate pointers.





Post published. View Post

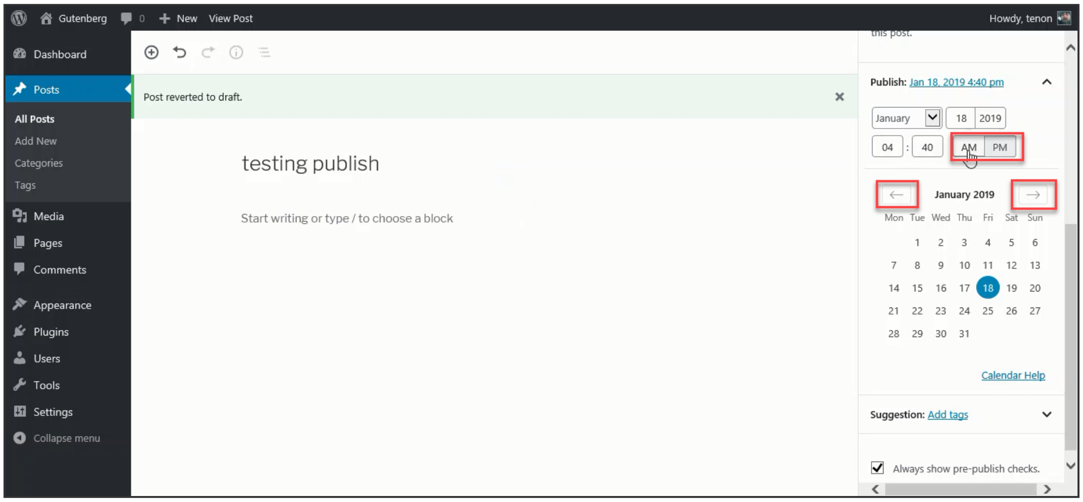
testing publish

Start writing or type / to choose a block

testing publish is now live.

http://142.93.114.99/2019/01/18/testi...

☑ Always show pre-publish checks.



Post reverted to draft.

testing publish

Start writing or type / to choose a block

Publish: Jan 18, 2019 4:40 pm

January 18 2019

04 : 40 AM PM

January 2019

Mon	Tue	Wed	Thu	Fri	Sat	Sun
	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	31			

☑ Always show pre-publish checks.

Issue Code

```
/* example: toolbar button */
.components-icon-button {
    ...
    padding: 8px;
    margin: 0;
    ...
}
/* example: Publish button */
.edit-post-header .components-button.editor-post-publish-panel__toggle, ... {
    margin: 2px;
    height: 33px;
    line-height: 32px;
    font-size: 13px;
}
.edit-post-header .components-button.editor-post-publish-panel__toggle {
    padding: 0 12px 2px;
}
```

Remediation Guidance

Increase the size of the targets. Most are already around ~35px in size, so techniques such as increasing their vertical padding or simply setting a minimum height (as shown in the Recommended Code) will allow controls to meet their minimum target sizes.

Recommended Code

```
/* example: toolbar button */
.components-icon-button {
    ...
    min-height: 44px
    ...
}
/* example: Publish button */
.edit-post-header .components-button.editor-post-publish-panel__toggle {
    min-height: 44px;
}
/* AM, PM buttons */
.components-button.is-button {
    padding: 0 10px 1px;
}
/* Move back/ front to month buttons*/
.components-datetime__date .DayPickerNavigation_button__horizontalDefault {
    padding: 2px 8px;
    top: 20px;
}
```

Relevant standards

- 2.5.5 Target Size (Level AAA)

GUT-4 - Mobile resolution and page zoom will cause content to be hidden or a horizontal scroll to show

- **Severity:** High
- **Affected Populations:** Low-Vision, Motor Impaired, Cognitively Impaired
- **Platform(s):** All/ Universal
- **Components affected:** Global

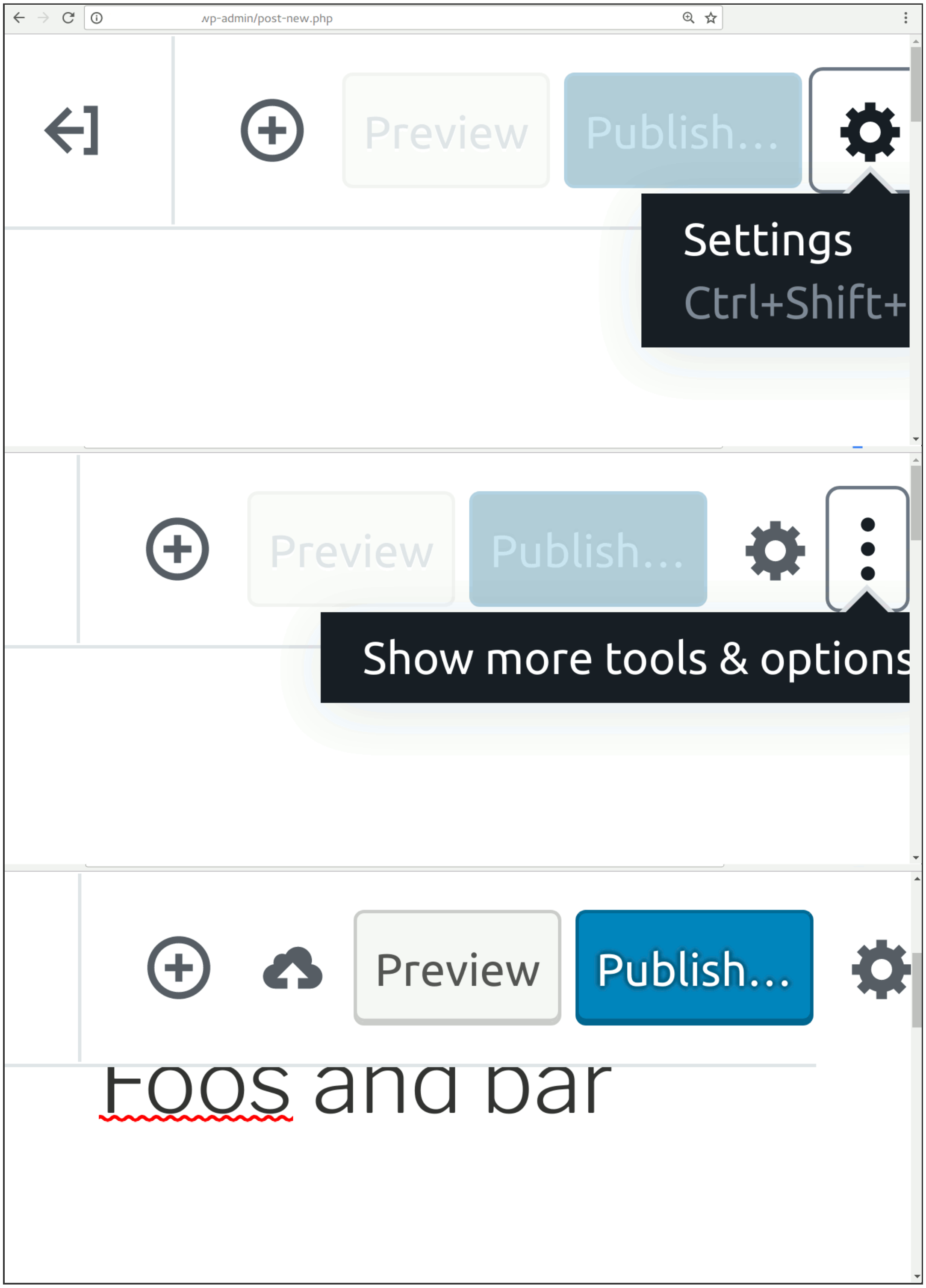
Issue description

The Editor Top Bar remains sticky to the top of the page even as users zoom up to 400%. Two problems occur from this:

- Depending on viewport resolution, the last button or two may go offscreen to the right (this is due to flexbox). Keyboard users can tab to these buttons, however touchscreen users and speech recognition users cannot reach these buttons, as there is no scrollbar.
- Depending on viewport height, users attempting to write content find a large part of their viewport is blocked by the sticky bar and partially covers typed text.

When the pages are zoomed, or in smaller windows, there is a horizontal scrollbar. In IE 11 it is shown as an inner scrollbar that overlaps or has a limited width so cannot always be used. On mobile the finger swipes work as expected. This also affects wide panels that do not wrap into new lines but overlap or go off the page.

The ability to zoom content may be essential for users with low-vision, and may be helpful for users who have a cognitive disability.


wp-admin/post-new.php
Preview
Publish…
Settings
Ctrl+Shift+
Preview
Publish…
Show more tools & options
Preview
Publish…
Foos and bar

```
.edit-post-header {
    ...
    display: flex;
    flex-direction: row;
    align-items: stretch;
    justify-content: space-between;
    position: -webkit-sticky;
    position: sticky;
}

/* media all */
.edit-post-layout__content {
    position: relative;
    display: flex;
    min-height: 100%;
    flex-direction: column;
    padding-bottom: 50vh;
    overflow-y: auto;
    -webkit-overflow-scrolling: touch;
}
```

Remediation Guidance

For the post header set the `position` attribute to `block` instead of `sticky` when a media query detects the viewport is small (20-30 em). This will hit zoom levels under 400% and allow users who zoom quite a lot to simply scroll past the bar.

Use the same media query to adjust the `flex` settings to allow wrapping; or implement a different design solution to the same effect. Currently, flex elements' alignment does not account for viewport edges, and allows content to go off screen without creating scrollbars; this effectively hides the overflowing content.

In general, when the screen is zoomed-in or in a low resolution, do not rely on horizontal scrolling: fit items into the screen by wrapping the content onto new lines when the resolution is smaller than the width of the content.

Allow links styled as buttons to wrap their text. Actual HTML buttons wrap naturally but have a set `line-height` of `1`, making their text difficult to read when wrapping. Set this to the same `line-height` as link-buttons (`1.66`).

Allow tables to be a width where no individual words need to be broken, by setting the wrapper of the table to have `overflow-x: scroll` so the table does not cause the page itself to have a horizontal scrollbar. Remove the `word-break` rule used in tables.

Relevant standards

- 1.4.10 Reflow (Level AA)

GUT-3 - Buttons vanish at larger zoom levels and on mobile

- **Severity:** High
- **Affected Populations:** Low-Vision, Cognitively Impaired
- **Platform(s):** All/ Universal
- **Components affected:** Editor Top Bar, Structure and Navigation

Issue description

Users relying on browser zoom to view the page content will eventually trigger a mobile style at higher zoom levels. This mobile style hides many of the buttons in the Editor Top Bar. There doesn't appear to be another method of performing the Undo, Redo, Content Structure or Block Navigation functions when the mobile view is triggered. Users with keyboards may never see the shortcuts for using these controls, while touchscreen users cannot use them at all.

The ability to zoom content may be essential for users with low-vision, and may be helpful for users who have a cognitive disability. Avoiding the need to scroll in the reading direction is also important, since this significantly increases the effort required to read the content.

Issue Code

```
.edit-post-header-toolbar>.components-button {
    display: none;
}
```

Remediation Guidance

Find an equivalent way for low-vision desktop users (and mobile users) to be aware of and use these controls, such as adding them under another menu.

Relevant standards

- 1.4.10 Reflow (Level AA)

Content Type: Parsing

GUT-12 - Date values in the datepicker are not fully parsed

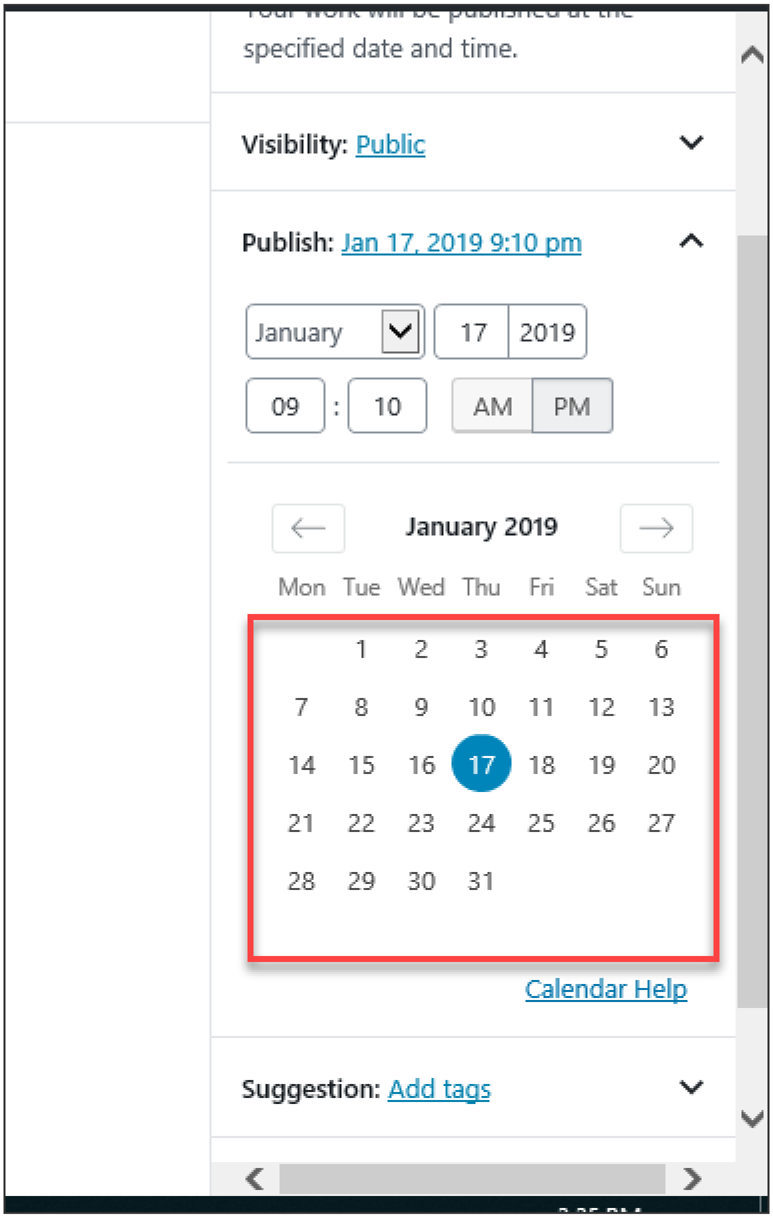
- **Severity:** High
- **Affected Populations:** Blind, Low-Vision, Cognitively Impaired
- **Platform(s):** Windows - Screen Reader, Mac - VoiceOver, Android - TalkBack, iOS - VoiceOver
- **Components affected:** Document Panel, Publish and Unpublish

Issue description

When navigating through the datepicker using a screen reader, the current date is announced as (for example) "Tuesday, F j, 2019" rather than "Tuesday, January 15, 2019".

These same "F j" tokens are present in all the `aria-label` attributes within the date picker.

This is almost certainly a technical problem, whereby date tokens are not being parsed into their equivalent language.



specified date and time.

Visibility: Public

Publish: Jan 17, 2019 9:10 pm

January ▾ 17 2019

09 : 10 AM PM

January 2019

Mon	Tue	Wed	Thu	Fri	Sat	Sun
	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	31			

Calendar Help

Suggestion: Add tags

Issue Code

```
<tr>

   <td tabindex="-1" class="..." role="button" style="width: 30px; height: 29px;" aria-label="Tuesday, F j,
2019">15</td>

   <td tabindex="-1" class="..." role="button" style="..." aria-label="Wednesday, F j, 2019">16</td>

   ...

</tr>
```

Remediation Guidance

Ensure that all date tokens are parsed into their equivalent language.

Recommended Code

```
<tr>

   <td tabindex="-1" class="..." role="button" style="width: 30px; height: 29px;" aria-label="Tuesday,  January
15, 2019">15</td>

   <td tabindex="-1" class="..." role="button" style="..." aria-label="Wednesday, January 16, 2019">16</td>

   ...

</tr>
```

Content Type: Text
Content and Typography

GUT-111 - Language and text direction are not editable in the visual editor

- **Severity:** Medium
- **Affected Populations:** Blind, Low-Vision, Cognitively Impaired
- **Platform(s):** All/ Universal
- **Components affected:** Block Editing, Title Block

Issue description

Users who need to add content in multiple languages cannot set the `lang` or `dir` attribute for a block using the visual editor. Instead, users must manually switch to the HTML editor.

In the HTML editor, adding a `dir` or `lang` attribute brings up an error and the offer to resolve the error by either removing the `dir` and `lang` attributes or converting the block to HTML. This allows users to manually add in these attributes, however the editor suggesting that these are errors may lead users to believe they cannot use these attributes at all. It also requires users to be familiar with HTML.

Remediation Guidance

Add the ability to define `lang` and `dir` attributes in the visual editor (probably via the block editing options), ideally with user-friendly options such as choosing from language names and then choosing between "left to right", "right to left" or "vertical".

Relevant standards

- 3.1.2 Language of Parts (Level AA)

Related Issues

- Related to "Language cannot be set on posts"

GUT-110 - Language cannot be set on posts

- **Severity:** High
- **Affected Populations:** Blind, Low-Vision, Cognitively Impaired
- **Platform(s):** All/ Universal
- **Components affected:** Output Markup

Issue description

Posts written in languages other than the editor's own setting cannot have the correct language attributes defined.

Declaring the correct language of a page is vital, so that browsers can determine how to render the textual content, and assistive technologies can determine how it should be pronounced.

Issue Code

```
<article id="post-19" class="...">...</article>
```

Remediation Guidance

Allow authors to set `lang` and `dir` attributes that apply to the whole post.

Recommended Code

```
<article lang="es" id="post-19" class="...">...</article>
```

Relevant standards

- 3.1.1 Language of Page (Level A)
- 3.1.2 Language of Parts (Level AA)

Related Issues

- Related to "Language and text direction are not editable in the visual editor"

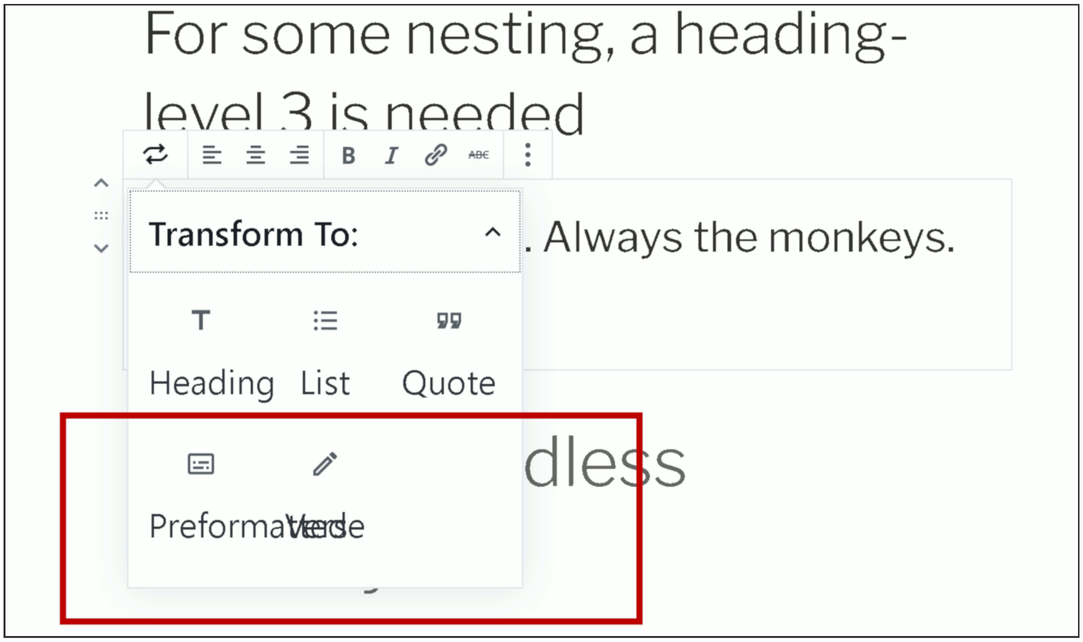
GUT-96 - Some content becomes unreadable with text enlarge

- **Severity:** Medium
- **Affected Populations:** Low-Vision
- **Platform(s):** All/ Universal
- **Components affected:** Block Editing

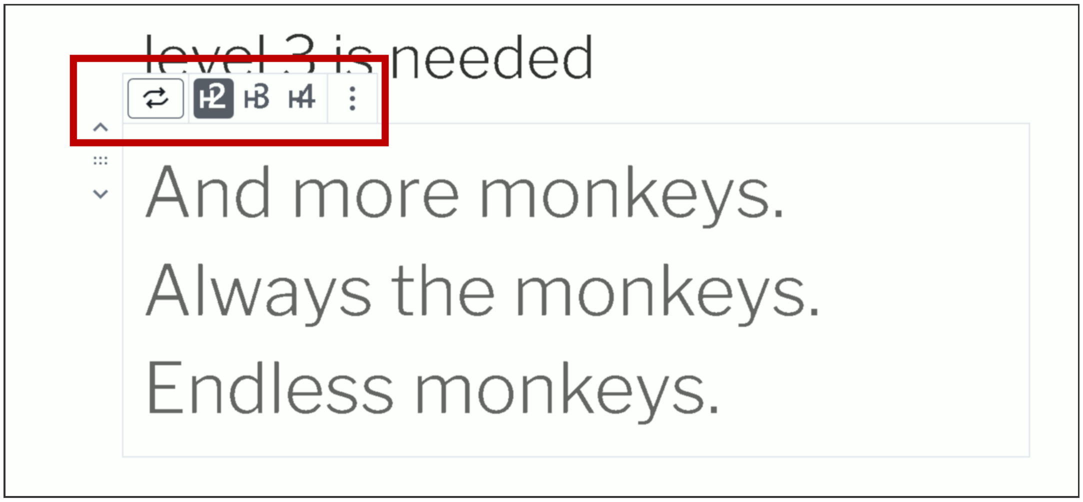
Issue description

Users who need to increase the text size will find that some block content overlaps its containers, and other content.

Specifically, the "Change Block Type" dropdown does not grow to encompass the Type blocks, causing them to overflow the dropdown container. Without the container background, it is difficult to read the block type name against the editable page below, when the item is not being hovered over (since hovering adds a background). The words of the type also spill over the text of adjacent types, making the text unreadable.



Some of the actionable buttons, such as setting a heading level within the toolbar, also become difficult to read due to text overlapping.



level 3 is needed

And more monkeys.
Always the monkeys.
Endless monkeys.

The ability to resize text is essential for users with low-vision, and may be helpful for users who have a cognitive disability. Catering to zoom alone is not sufficient because the browser's font-size may be increased independently of zoom level.

Issue Code

```
.editor-block-types-list__list-item {
    ...
    width: 33.33%;
    ...
}
...
/* hx buttons */
<button type="button" aria-label="Heading 2" data-subscript="2" aria-pressed="true" class="...">
    <svg ..." width="20" height="20" viewBox="0 0 20 20">...</svg>
</button>
...
.components-toolbar__control.components-button[data-subscript]::after {
    content: attr(data-subscript);
    ...
    font-size: 13px;
    font-weight: 600;
    line-height: 12px;
    position: absolute;
    right: 8px;
    bottom: 10px;
}
```

Remediation Guidance

Keep width and height minimums to 44px or greater. Set minimum sizes using `min-width` and `min-height`, rather than fixed sizes using `width` and `height`, to allow containers to expand with increases in their content size.

Set SVG sizes in units that will scale with text, such as `em` or `rem`.

Avoid using absolute positioning for text that is close enough to other text to overlap at larger font sizes. Absolutely-positioned elements are outside the layout flow, and cannot be adjusted for by the layout of other elements.

Recommended Code

```
.editor-block-types-list__list-item {
    ...
    min-width: 33.33%;
    ...
}
...
/* hx buttons */
<button type="button" aria-label="Heading 2" data-subscript="2" aria-pressed="true" class="...">
    <svg ..." viewBox="0 0 20 20">...</svg>
</button>
.components-toolbar__control.components-button:not(:disabled).is-active>svg {
    ...
    height: 1em;
    width: 1em;
}
...
.components-toolbar__control.components-button[data-subscript]::after {
    content: attr(data-subscript);
    ...
}
```

Relevant standards

- 1.4.4 Resize text (Level AA)

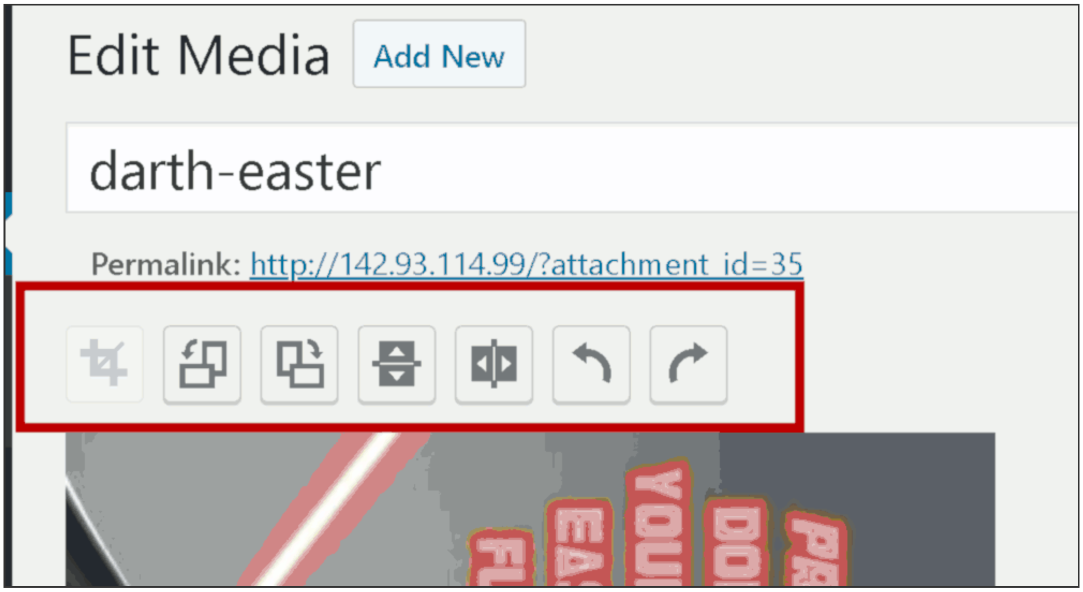
GUT-85 - Visible labels are not available on editing buttons

- **Severity:** Medium
- **Affected Populations:** Low-Vision, Cognitively Impaired
- **Platform(s):** All/ Universal
- **Components affected:** Edit Media

Issue description

On the Edit Media page, buttons for editing the selected media image have accessible names to label them for assistive technology users, however these names are not available to sighted users.

Since some of the icons are quite cryptic, and there is an ambiguity between two sets of icons which both appear to mean Rotate (see GUT-84 Done), this may be confusing to users with cognitive disabilities. Such users may find themselves having to interact with buttons just to figure out what they do.



Remediation Guidance

Add visible text labels to the image editing buttons.

One method, which would be consistent with the rest of the editor, would be to use keyboard-accessible tooltips for this.

A better method would be to provide a user setting for all icon controls, so that users can specify whether to show icons-only, labels-only, or icons and labels together.

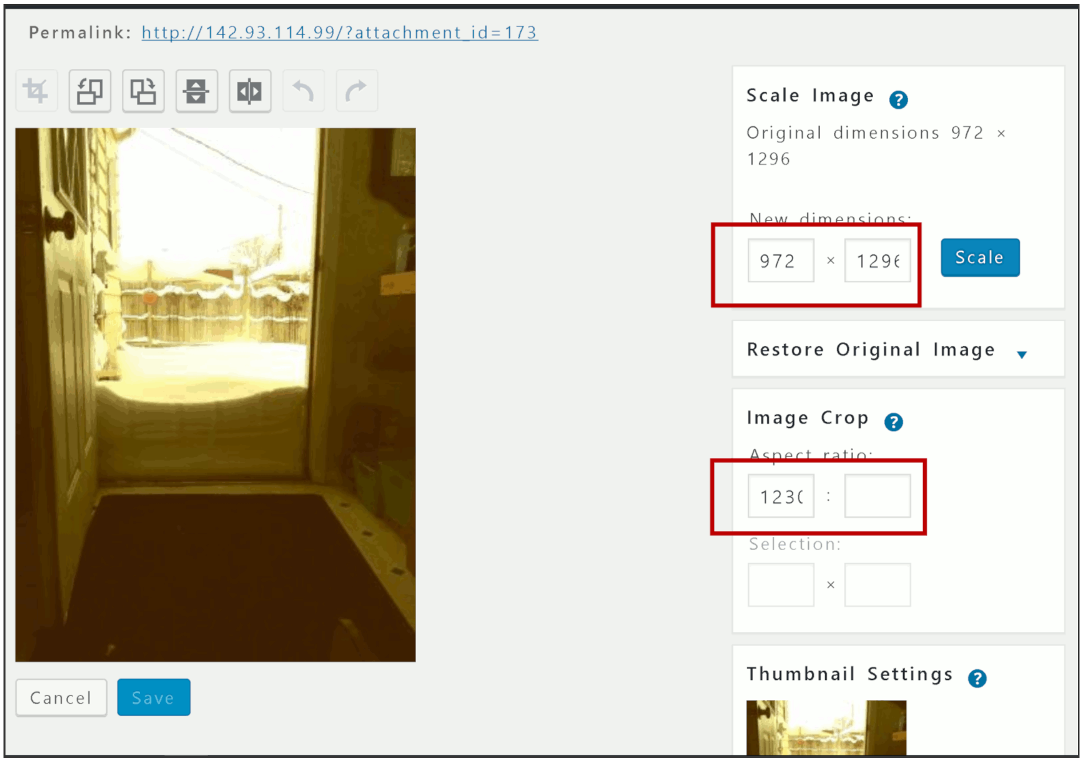
GUT-77 - Text overlaps or is clipped when text spacing or size is enlarged

- **Severity:** Medium
- **Affected Populations:** Low-Vision, Cognitively Impaired
- **Platform(s):** All/ Universal
- **Components affected:** Edit Media

Issue description

On the Edit Media page, users who need to change text spacing (such as distance between words, characters, or lines of text) or who need to enlarge the text (via text settings) are unable to view all the text on the page, due to text overlapping or being clipped by its containers.

The ability to resize or adjust spacing of text is essential for users with low-vision, and may be helpful for users who have a cognitive disability. Catering to zoom alone is not sufficient because the browser's font-size may be increased independently of zoom level.



Permalink: http://142.93.114.99/?attachment_id=173

Scale Image

Original dimensions 972 × 1296

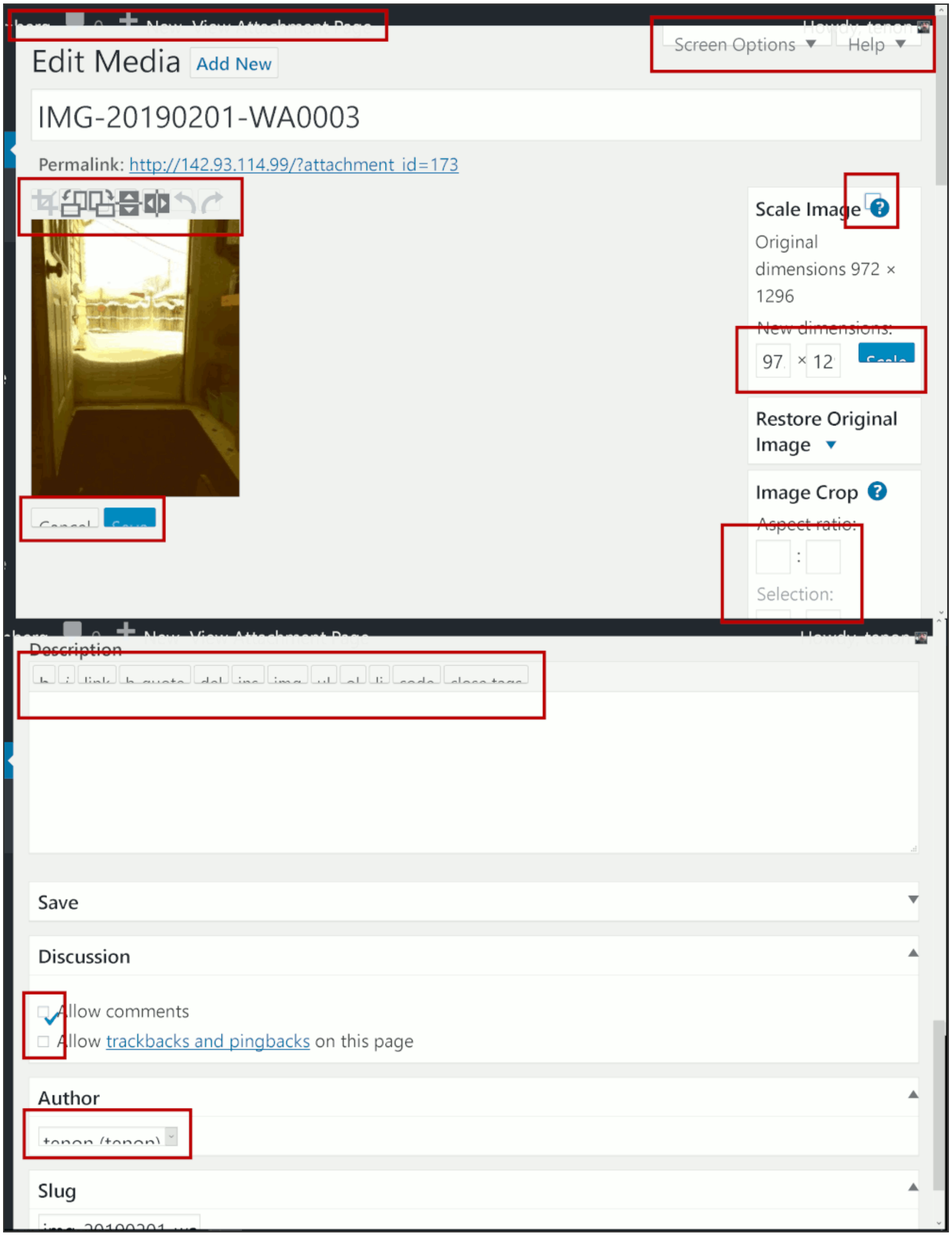
New dimensions: 972 × 1296

Image Crop

Aspect ratio: 123(:

Selection: ×

Thumbnail Settings



IMG-20190201-WA0003

Permalink: http://142.93.114.99/?attachment_id=173

Original dimensions 972 × 1296

New dimensions: 97 × 12

☑ Allow comments

□ Allow trackbacks and pingbacks on this page

Author: tenon (tenon)

Slug: img-20190201-wa

Issue Code

```
.wp-core-ui .button, .wp-core-ui .button-primary, .wp-core-ui .button-secondary {
    ...
    font-size: 13px;
    line-height: 26px;
    height: 28px;
   ...
}
.image-editor .imgedit-menu .button {
    ...
    width: 32px;
    height: 32px;
    ...
    line-height: 16px;
    ...
}
.imgedit-settings .imgedit-crop-ratio input[type="text"], .imgedit-settings .imgedit-crop-sel input[type="text"],
.imgedit-settings .imgedit-scale input[type="text"] {
    width: 50px;
    font-size: 14px;
    padding: 5px 8px;
}
...
.wp-core-ui .quicktags-toolbar input.button.button-small {
    font-size: 12px;
    height: 26px;
    line-height: 24px;
}
```

Remediation Guidance

Keep width and height minimums to 44px or greater. Set minimum sizes using `min-width` and `min-height`, rather than fixed sizes using `width` and `height`, to allow containers to naturally expand with increases in content size.

Recommended Code

```
.wp-core-ui .button, .wp-core-ui .button-primary, .wp-core-ui .button-secondary {
    ...
    line-height: 1.5;
    min-height: 44px;
   ...
}
.image-editor .imgedit-menu .button {
    ...
    min-width: 44px;
    min-height: 44px;
    ...
    line-height: 1;
    ...
}
.imgedit-settings .imgedit-crop-ratio input[type="text"], .imgedit-settings .imgedit-crop-sel input[type="text"],
.imgedit-settings .imgedit-scale input[type="text"] {
    min-width: 50px;
    ...
    padding: 5px 8px;
}
...
.wp-core-ui .quicktags-toolbar input.button.button-small {
    ...
    min-height: 44px;
    line-height: 1.5;
}
```

Relevant standards

- 1.4.4 Resize text (Level AA)
- 1.4.12 Text Spacing (Level AA)

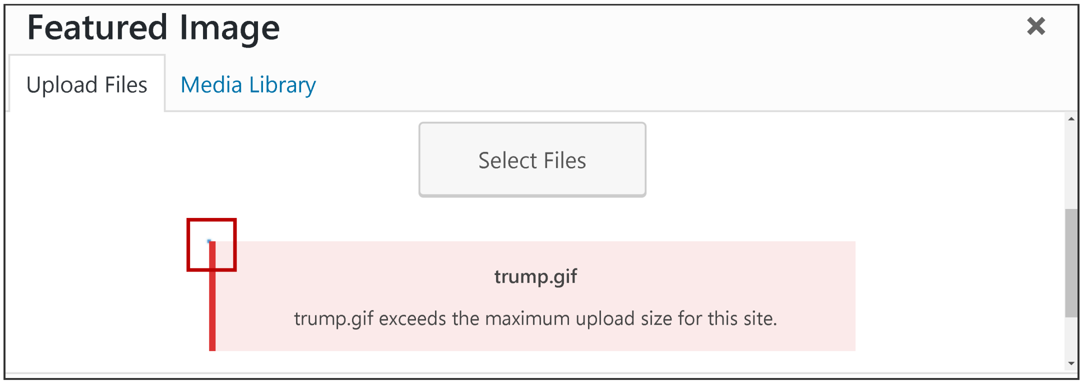
GUT-62 - Control is invisible

- **Severity:** High
- **Affected Populations:** Low-Vision, Motor Impaired, Cognitively Impaired
- **Platform(s):** All/ Universal
- **Components affected:** Media Dialog

Issue description

When users attempt to upload an image in the "Featured Image" modal dialog, and receive an error, a close button is present to remove the error message. This button is invisible to most users, although it does have visually-hidden text for screen reader users.

Sighted keyboard-only users will see only the box-shadow when they happen to focus on it, while Window High Contrast users see nothing (because box-shadow is removed by High Contrast themes).



trump.gif

trump.gif exceeds the maximum upload size for this site.

Issue Code

```
b           "b    "  l   "b     li k   l  d di  i         "
```

```
<button type="button" class="button-link upload-dismiss-errors">
    <span class="screen-reader-text">Dismiss Errors</span>
</button>
...
.wp-core-ui .button-link:focus {
    color: #124964;
    box-shadow: 0 0 0 1px #5b9dd9, 0 0 2px 1px rgba(30,140,190,.8);
}
```

Remediation Guidance

Add an icon or text to make the button visible, such as an "x" icon.

Add an `outline` property to the button when it's focused so that Windows High Contrast keyboard users can see where the focus position is.

Ensure the button is at least 44px by 44px to meet minimum target-size requirements.

Recommended Code

```
.wp-core-ui .button-link:before {
    content: "\f158";
    font: normal 20px/1 dashicons;
    speak: none;
    vertical-align: middle;
    -webkit-font-smoothing: antialiased;
    -moz-osx-font-smoothing: grayscale;
}
...
.wp-core-ui .button-link:focus {
    color: #124964;
    box-shadow: 0 0 0 1px #5b9dd9, 0 0 2px 1px rgba(30,140,190,.8);
    outline: 1px dotted transparent;
}
```

Relevant standards

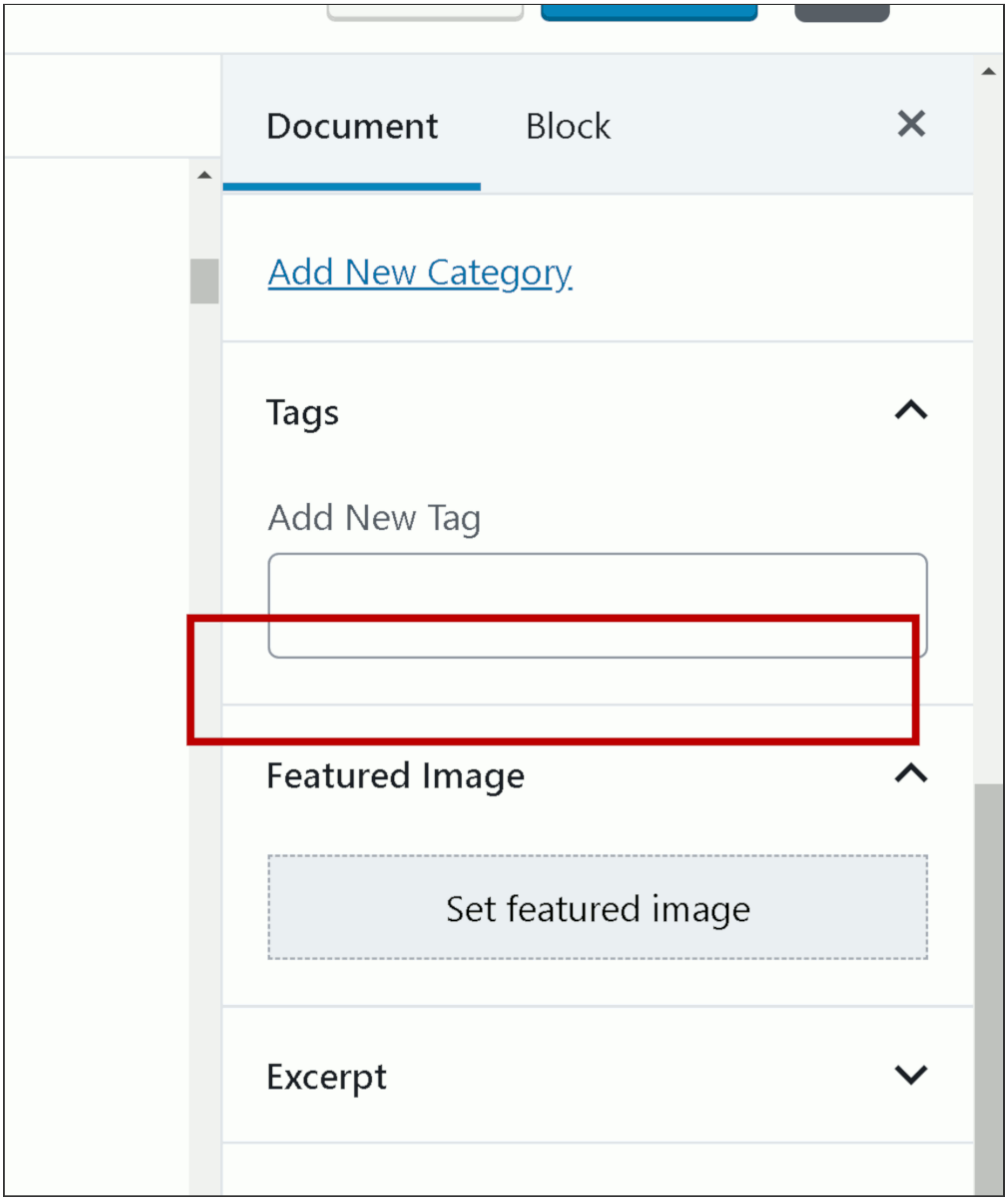
- 1.3.3 Sensory Characteristics (Level A)
- 2.4.7 Focus Visible (Level AA)
- 2.5.5 Target Size (Level AAA)

GUT-40 - Useful content is unnecessarily hidden

- **Severity:** Low
- **Affected Populations:** Motor Impaired, Cognitively Impaired
- **Platform(s):** All/ Universal
- **Components affected:** Block Panel

Issue description

After the tags input, where users can add tags to their document, hidden text states that these can be comma-separated. This text is only available to screen reader users, even though these instructions are useful to many sighted users, such as those with a cognitive disability and keyboard-only users.



Document
Block
Add New Category
Tags
Add New Tag
Featured Image
Set featured image
Excerpt

Issue Code

```
<div class="components-form-token-field" tabindex="-1">

    <label for="..." class="...">Add New Tag</label>

    <div class="..." tabindex="-1">

        <input id="..." type="text" size="1" class="..." role="combobox" aria-expanded="false" aria-
autocomplete="list" aria-describedby="..." value="">

    </div>

    <div id="..." class="screen-reader-text">Separate with commas</div>

</div>
```

Remediation Guidance

Allow the instructions to be available to all users, not just screen reader users.

Recommended Code

```
<div class="components-form-token-field" tabindex="-1">

    <label for="..." class="...">Add New Tag</label>

    <div class="..." tabindex="-1">

        <input id="..." type="text" size="1" class="..." role="combobox" aria-expanded="false" aria-
autocomplete="list" aria-describedby="..." value="">

    </div>

    <div id="...">Separate with commas</div>

</div>
```

Related Issues

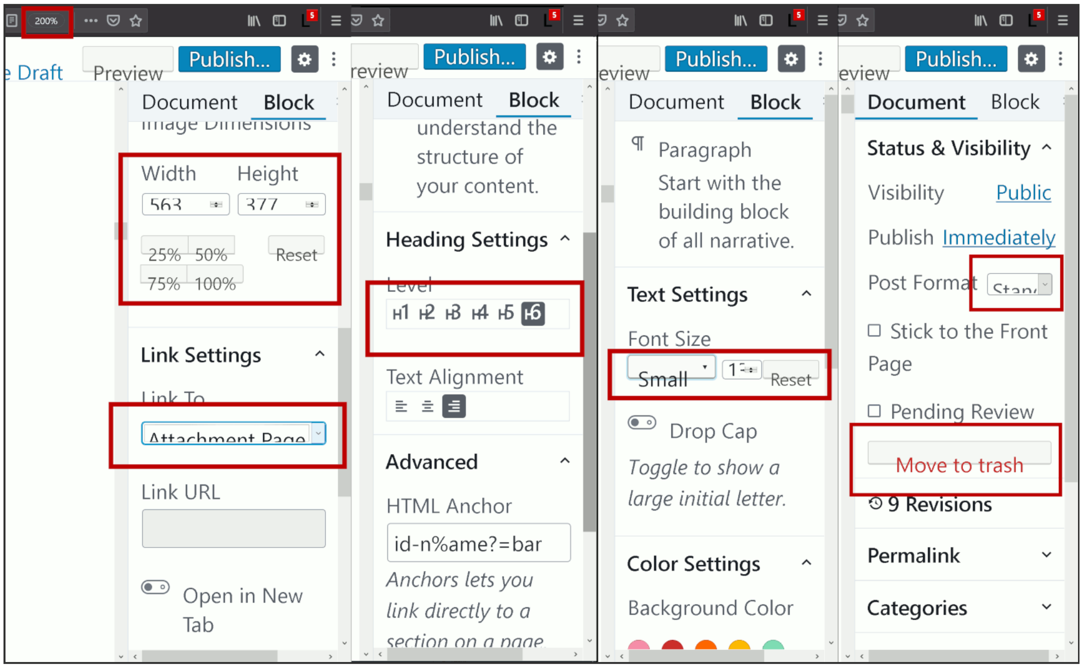
- Related to "Suggested tags should be a list"

GUT-32 - Content overflows and is cut off at 200% text enlarge

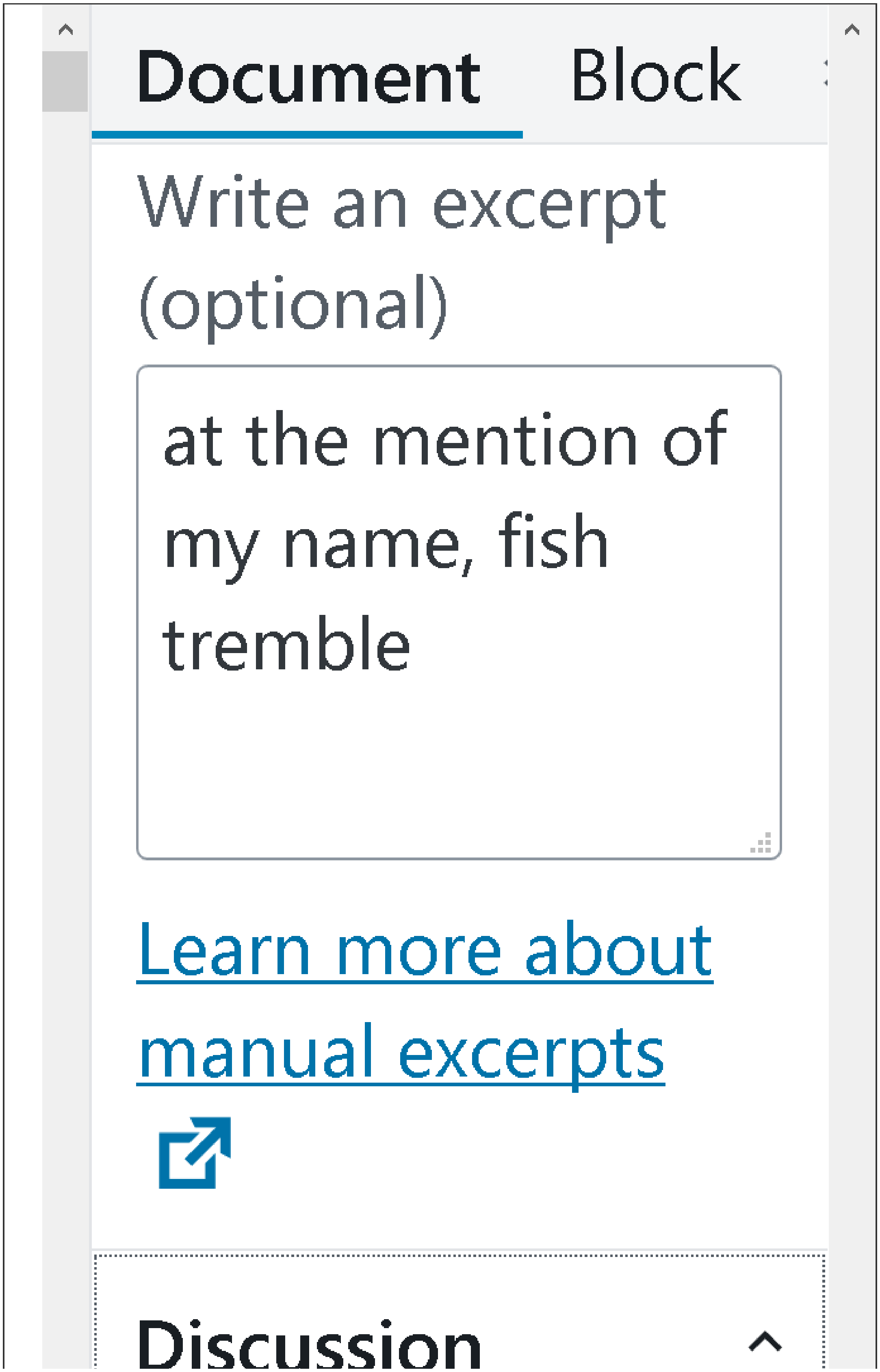
- **Severity:** Medium
- **Affected Populations:** Low-Vision, Cognitively Impaired
- **Platform(s):** All/ Universal
- **Components affected:** Block Panel, Document Panel

Issue description

Several controls allow text to overflow out of them, or clip the text, at 200% text enlarge. This is due to containers being set in fixed pixel heights, which cannot grow with their content.



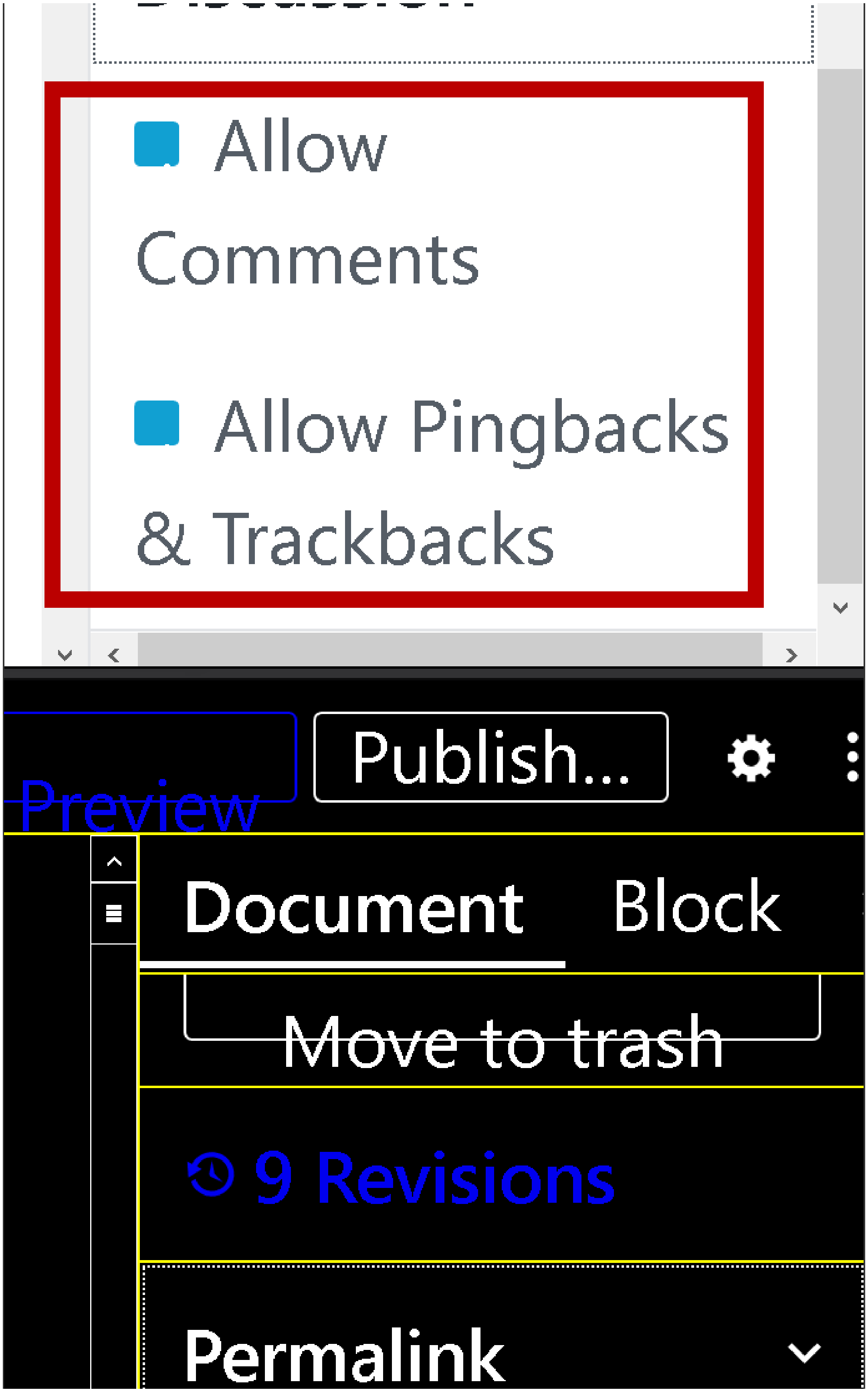
The checkmark icons move out of their checkboxes as they grow, leaving a white checkmark against a white page background.



Write an excerpt (optional)

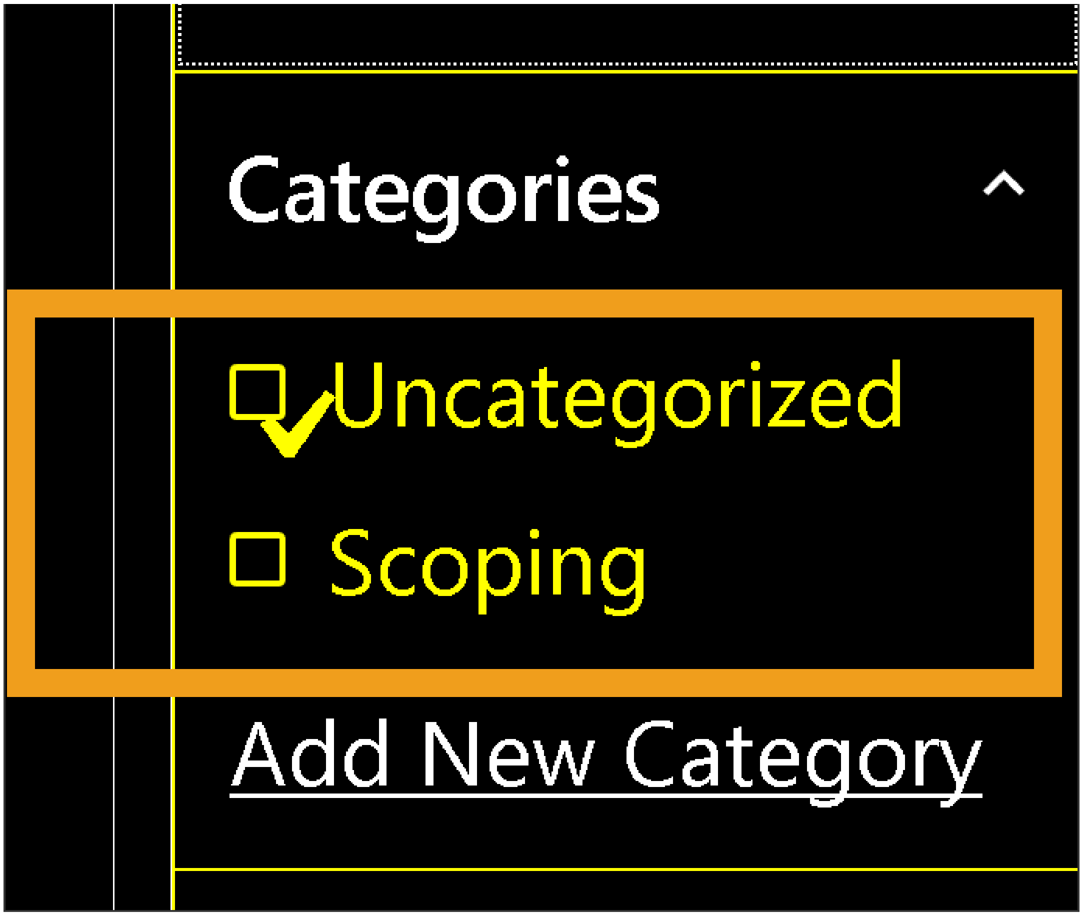
at the mention of my name, fish tremble

Learn more about manual excerpts



☑ Allow Comments

☑ Allow Pingbacks & Trackbacks



Categories

☑ Uncategorized

□ Scoping

Add New Category

The ability to resize text is essential for users with low-vision, and may be helpful for users who have a cognitive disability. Catering to zoom alone is not sufficient because the browser's font-size may be increased independently of zoom level.

Issue Code

```
/* selects */
.wp-admin select {
    ...
    line-height: 28px;
    height: 28px;
    ...
}

/* checkboxes/radios */
input[type=checkbox], input[type=radio] {
    ...
    height: 16px;
    ...
    width: 16px;
    min-width: 16px;
}

/* buttons */
.components-button.is-large {
    height: 30px;
    line-height: 28px;
    ...
}

.components-button.is-small {
    height: 24px;
    line-height: 22px;
    ...
    font-size: 11px;
}

/* pressable buttons */
.components-toolbar__control.components-button {
    ...
    width: 36px;
    height: 36px;
}

/* number inputs */
input[type=number] {
    height: 28px;
    line-height: 1;
}
```

Remediation Guidance

Avoid setting fixed heights on elements (even inputs), and especially in `px` units. Instead, set min-heights on containers, allowing them to always expand to enclose their content, and allowing containers to themselves wrap as needed.

The Recommended Code is using a minimum of `44px` for the settings (following https://www.w3.org/TR/WCAG21/#target-size), meaning designers may want to reconsider how they want to show some buttons as larger than others (visual importance), as well as dealing with horizontal button rows wrapping as necessary (such as in the "Image Resize" section of the Block Panel).

The checkboxes are made larger so that the growing checkmark icon remains visible inside the checkboxes (by allowing them to grow with the checkmark). These do not need to meet target size minimums as they are inline with their (clickable) labels.

Recommended Code

```css
/* selects */
.wp-admin select {
    ...
    min-height: 44px;
    ...
}

/* checkboxes/radios */
input[type=checkbox], input[type=radio] {
    ...
    height: 1.5em;
    ...
    width: 1.5em;
    min-width: 16px;
}

/* buttons */
.components-button.is-large {
    min-height: 54px;
    ...
}

.components-button.is-small {
    min-height: 44px;
    ...
}

/* pressable buttons */
.components-toolbar__control.components-button {
    ...
    min-width: 44px;
    min-height: 44px;
}

/* number inputs */
input[type=number] {
    min-height: 44px;
}
```

Relevant standards

- 1.4.4 Resize text (Level AA)

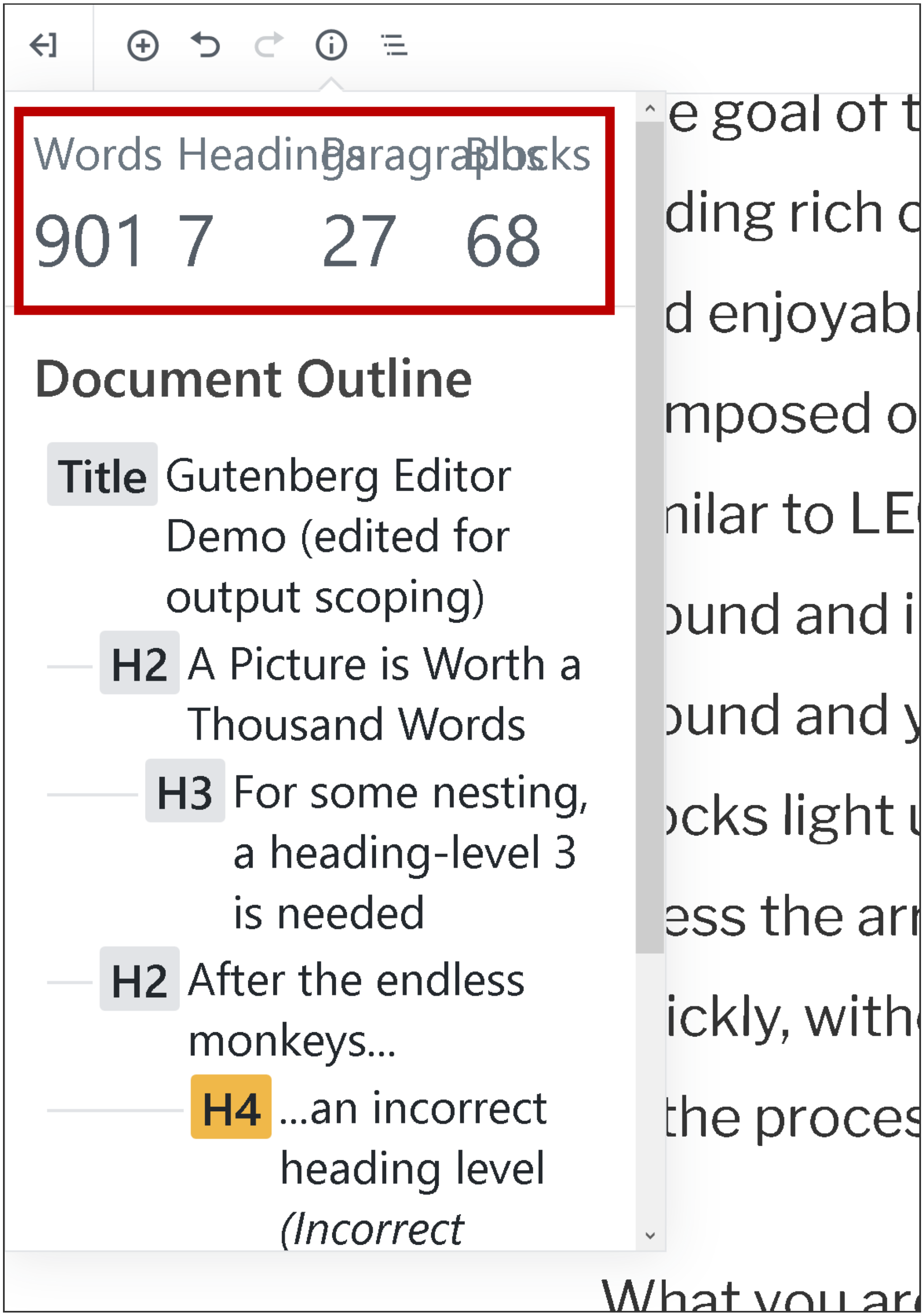
GUT-21 - Content overlaps at 200% text enlarge

- **Severity:** Medium
- **Affected Populations:** Low-Vision, Cognitively Impaired
- **Platform(s):** All/ Universal
- **Components affected:** Structure and Navigation

Issue description

The summary text at the top of the Content Structure popup is unreadable at 200% text enlarge, due to overlapping text. The containers have set widths and cannot adjust to changes in font-size, which causes individual words (which cannot wrap without hard word-breaks) to overflow their containers.

The ability to resize text is essential for users with low-vision, and may be helpful for users who have a cognitive disability. Catering to zoom alone is not sufficient because the browser's font-size may be increased independently of zoom level.



Words	Headings	Paragraphs	Blocks
901	7	27	68

Document Outline

- **Title** Gutenberg Editor Demo (edited for output scoping)
  - **H2** A Picture is Worth a Thousand Words
    - **H3** For some nesting, a heading-level 3 is needed
  - **H2** After the endless monkeys…
    - **H4** …an incorrect heading level *(Incorrect*

e goal of t
ding rich c
d enjoyabl
mposed o
nilar to LE
ound and i
ound and y
ocks light u
ess the arr
ickly, with
the proces

What you are

Issue Code

```
.table-of-contents__count {
    width: 25%;
    display: flex;
    flex-direction: column;
    ...
}
```

Remediation Guidance

Avoid fixed widths on containers (even percentage-based widths) when the content inside cannot wrap.

Instead, set min-widths on containers, and allow them to expand with increases in content size, allowing the containers themselves to wrap as needed.

Relevant standards

- 1.4.4 Resize text (Level AA)

Related Issues

- Related to "Content lacks semantic markup"

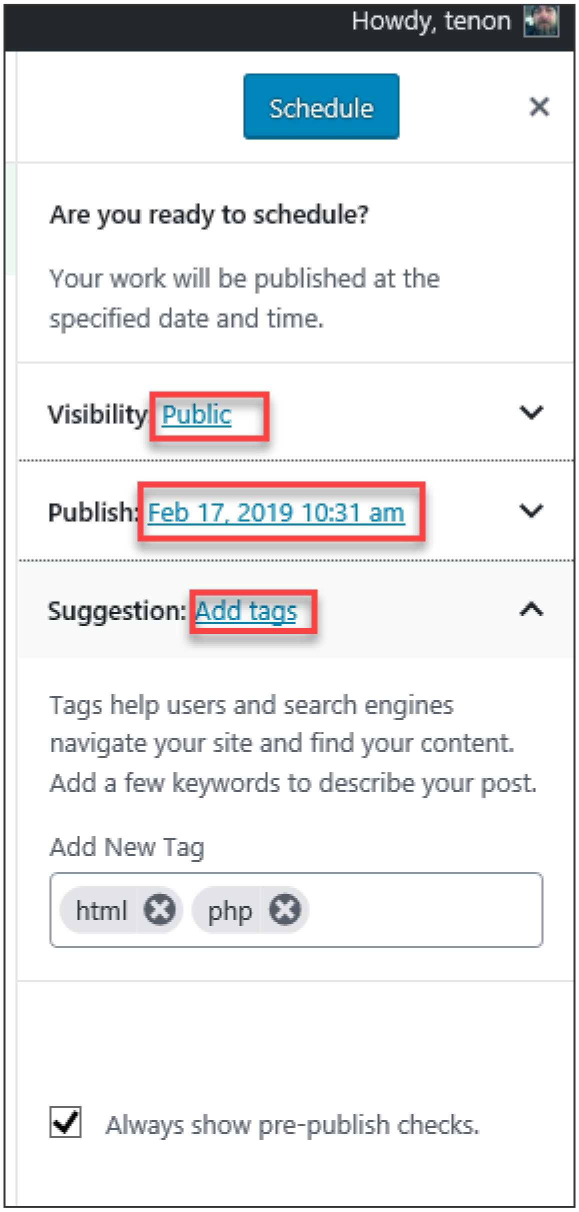
GUT-13 - Plain text appears to be a link

- **Severity:** Medium
- **Affected Populations:** Low-Vision, Motor Impaired, Cognitively Impaired
- **Platform(s):** All/ Universal
- **Components affected:** Publish and Unpublish

Issue description

There are three text elements within the Publish panel (the values for "Visibility", "Publish", and "Suggestion") which are styled to appear as links, but do not function as links. The entire containing element is a button to open or close the expanding region, while the text elements themselves have no direct function.

This may be confusing for users, since the expectation conveyed by their appearance is that these elements are links.



Are you ready to schedule?

Your work will be published at the specified date and time.

Visibility: Public

Publish: Feb 17, 2019 10:31 am

Suggestion: Add tags

Tags help users and search engines navigate your site and find your content. Add a few keywords to describe your post.

Add New Tag

html ⊗ php ⊗

☑ Always show pre-publish checks.

Issue Code

```
<button class="components-button components-panel__body-toggle" aria-expanded="false" type="button">
   ...
   Visibility:
   <span class="editor-post-publish-panel__link">Public</span>
</button>

<button class="components-button components-panel__body-toggle" aria-expanded="false" type="button">
    ...
    Publish:
    <span class="editor-post-publish-panel__link">Jan 18, 2019 4:40 pm</span>
</button>

<button class="components-button components-panel__body-toggle" aria-expanded="false" type="button">
   ...
    Suggestion:
    <span class="editor-post-publish-panel__link">Add tags</span></button>

/* CSS */
.editor-post-publish-panel__link {
    color: #007fac;
    text-decoration: underline;
    ...
}
```

Remediation Guidance

Remove the `text-decoration` and `color` from these elements so that they don't look like links.

Recommended Code

```
<button class="components-button components-panel__body-toggle" aria-expanded="false" type="button">
   ...
   Visibility:
   <span class="editor-post-publish-panel__link">Public</span>
</button>

<button class="components-button components-panel__body-toggle" aria-expanded="false" type="button">
    ...
    Publish:
    <span class="editor-post-publish-panel__link">Jan 18, 2019 4:40 pm</span>
</button>

<button class="components-button components-panel__body-toggle" aria-expanded="false" type="button">
   ...
    Suggestion:
    <span class="editor-post-publish-panel__link">Add tags</span></button>

/* CSS */
.editor-post-publish-panel__link {
    ...
}
```

Relevant standards

- 1.3.3 Sensory Characteristics (Level A)

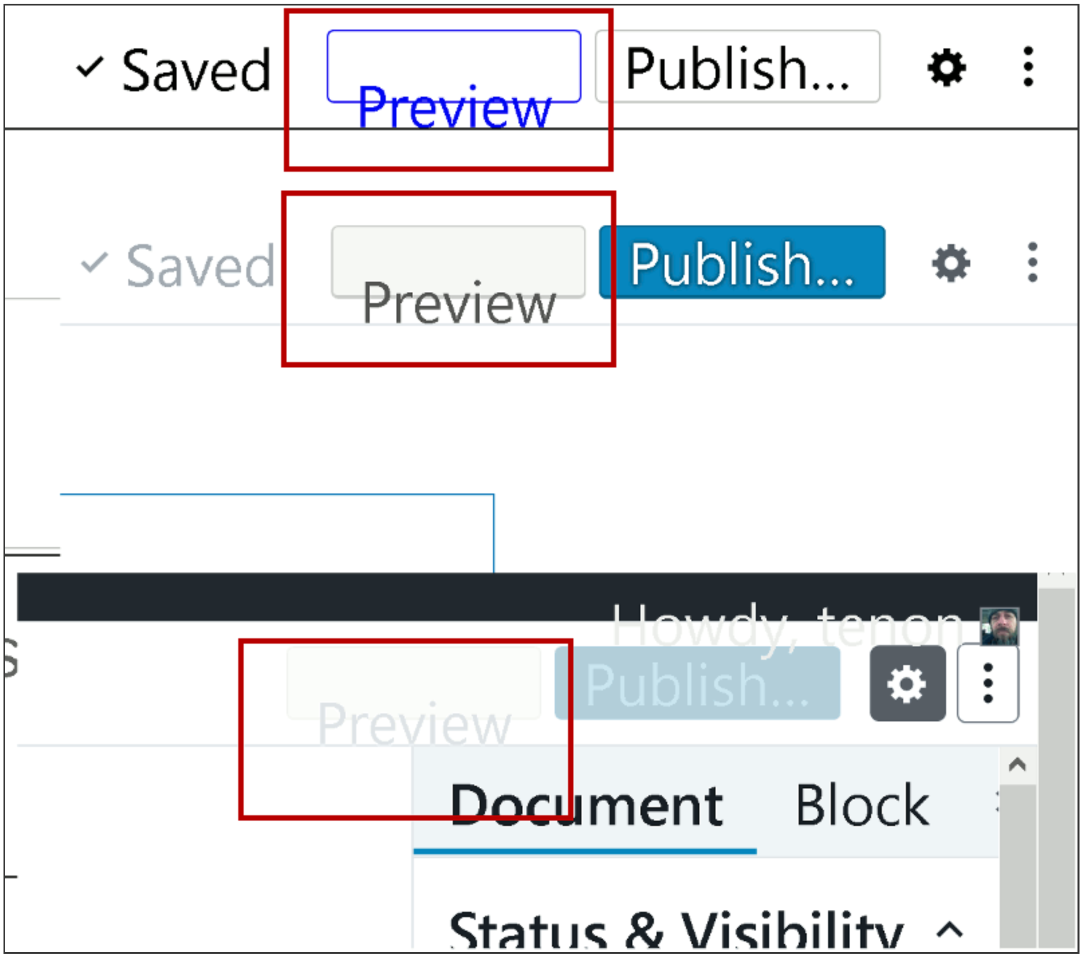
GUT-2 - Button text overflows button borders at 200% text enlarge

- **Severity:** Medium
- **Affected Populations:** Low-Vision, Cognitively Impaired
- **Platform(s):** All/ Universal
- **Components affected:** Editor Top Bar, Media Dialog

Issue description

Users who enlarge text alone to 200% will see the "Preview" button misaligned with its borders in the Editor Top Bar. This may make reading it more difficult for users with cognitive disabilities such as dyslexia.

The ability to resize text is essential for users with low-vision, and may be helpful for users who have a cognitive disability. Catering to zoom alone is not sufficient because the browser's font-size may be increased independently of zoom level.



Issue Code

```
b         i  b        {
```

```
.components-button.is-button {
    ...
    line-height: 26px;
}
.components-button.is-large {
    ...
    line-height: 28px;
}
.edit-post-header .components-button.editor-post-preview {
    height: 33px;
    line-height: 32px;
}

.media-frame-title {
    position: absolute;
    top: 0;
    ...
    height: 50px;
    ...
}
.media-frame-router {
    position: absolute;
    top: 50px;
    ...
}
...
.wp-core-ui .button.button-large {
    height: 30px;
    line-height: 28px;
    padding: 0 12px 2px;
}
...
.wp-core-ui .button.button-hero {
    ...
    height: 46px;
    line-height: 44px;
    ...
}
```

Remediation Guidance

Avoid setting fixed-pixel `height` for containers of text, since text can expand separately from browser zoom.

Allow line-heights and container heights to grow naturally with text, by using `min-height` and vertical `padding` instead of `height`.

Recommended Code

```
.media-frame-title {
    ...
    min-height: 50px;
    ...
}
.media-frame-router {
    ...
}
...
.wp-core-ui .button.button-large {
    min-height: 30px;
}
...
.wp-core-ui .button.button-hero {
    ...
    min-height: 46px;
    ...
}
```

Relevant standards

- 1.4.4 Resize text (Level AA)

Appendices

These appendices allow you to browse issues in different ways, helping you in the process of reviewing and managing remediation of each issue.

Issues By severity

Each issue is assigned an **severity**, which indicates the predicted impact of the issue on users with disabilities.

- **HIGH** – affected users will be prevented from completing a task
- **MEDIUM** – affected users will face significant problems completing a task
- **LOW** – affected users will experience minor difficulties completing a task

Note: The severity of an issue provides a guide to remediation priority only, and is not necessarily the same as the priority level of WCAG Success Criteria related to the issue. Unless indicated otherwise, **all** issues listed in the report require remediation to conform with WCAG 2.0.

Low

- GUT-114 - Users are not warned against autoplaying videos
- GUT-104 - Citation field is only indicated by placeholder text
- GUT-102 - Heading block uses a counter-intuitive icon
- GUT-97 - Blocks use custom selection color
- GUT-87 - Permalink and its Edit button could be clearer
- GUT-84 - Undo and Redo icons are not consistent throughout the interface
- GUT-80 - Drag and drop functionality unavailable to keyboard or touchscreen
- GUT-76 - Field boundaries have insufficient color contrast
- GUT-75 - Button has insufficient color contrast
- GUT-74 - Radio and checkbox labels rely on implicit association
- GUT-71 - Reading order does not match visual order
- GUT-67 - Missing role on menu item
- GUT-66 - Icons alone are visible for controls unless they are hovered or focussed
- GUT-64 - Visible labelling only done via placeholder attribute
- GUT-54 - Form controls are not grouped
- GUT-50 - Insufficient form semantics

- GUT-49 - Insufficient labelling and inappropriate nesting of form controls
- GUT-44 - Insufficient contrast on empty field borders
- GUT-40 - Useful content is unnecessarily hidden
- GUT-39 - Instructions are unclear
- GUT-35 - Text styled as headings are not headings
- GUT-25 - Content lacks semantic markup
- GUT-18 - Unnecessary use of aria-label
- GUT-14 - Suggested tags should be a list
- GUT-11 - SVGs are focusable in IE
- GUT-9 - Incorrect cursor used on buttons
- GUT-6 - Tooltips cannot be dismissed without moving the pointer or focus
- GUT-5 - Buttons are smaller than the minimum target size

Medium

- GUT-113 - Users cannot add video metadata in the visual editor
- GUT-112 - Users can duplicate post titles
- GUT-111 - Language and text direction are not editable in the visual editor
- GUT-109 - Blocks are difficult to edit with speech recognition
- GUT-108 - Users are able to directly link images
- GUT-105 - Current-page link is not programmatically labelled as such
- GUT-103 - Users cannot add table metadata in the visual editor
- GUT-101 - Menu remains open after actions which moves focus
- GUT-100 - Selected block outline has insufficient color contrast
- GUT-99 - Keyboard focus order doesn't match visual order at high browser zoom
- GUT-96 - Some content becomes unreadable with text enlarge
- GUT-95 - Keyboard operation is confusing and inconsistent
- GUT-94 - Changes and updates are not announced to assistive technologies
- GUT-89 - Insufficient color contrast on block type buttons
- GUT-88 - Image cropping requires mouse and keyboard
- GUT-85 - Visible labels are not available on editing buttons
- GUT-83 - Page content is an invisible Tab stop

- GUT-82 - Reading and focus order do not match visual order
- GUT-81 - Disabled fieldset does not convey the disabled state of controls
- GUT-79 - Options vanish at higher zoom levels
- GUT-78 - Cursor suggests functionality even when functionality is not present
- GUT-77 - Text overlaps or is clipped when text spacing or size is enlarged
- GUT-73 - Status message not exposed to assistive technologies
- GUT-68 - Custom interactions in control styled as a native control
- GUT-63 - Live search does not announce results to assistive technologies
- GUT-58 - Tab structure does not use tab semantics
- GUT-55 - Insufficient color contrast in the options menu
- GUT-53 - Keyboard focus is not managed when dialogs close
- GUT-51 - Shortcut keys conflict with browser shortcuts
- GUT-43 - Insufficient contrast on link text
- GUT-41 - Text inadvertently rendered by assistive technologies
- GUT-36 - Form controls are neither grouped nor named
- GUT-32 - Content overflows and is cut off at 200% text enlarge
- GUT-31 - Focus state has low contrast
- GUT-28 - Page content can be accessed while dropdowns are open
- GUT-26 - Focus not visible on Content Structure pane
- GUT-21 - Content overlaps at 200% text enlarge
- GUT-20 - Buttons are low contrast and focus not visible
- GUT-19 - Insufficient color contrast for calendar selected date and Publish button
- GUT-17 - Focus not visible on block types in High Contrast modes
- GUT-15 - Keyboard focus is not managed when dynamic content disappears
- GUT-13 - Plain text appears to be a link
- GUT-10 - Undo and Redo functions do not communicate changes
- GUT-8 - Toolbar elements are tabbable
- GUT-2 - Button text overflows button borders at 200% text enlarge
- GUT-1 - Insufficient color contrast on Editor Top Bar

High

- GUT-110 - Language cannot be set on posts
- GUT-106 - Media players receive focus but show no visible focus state
- GUT-70 - Single visible menu is made up of multiple menus
- GUT-65 - Custom field check box automatically changes the context on user focus
- GUT-62 - Control is invisible
- GUT-60 - Errors and field information are not associated with their control
- GUT-45 - Focus is invisible in high contrast views
- GUT-42 - Feature Image dialog does not follow the dialog pattern
- GUT-33 - Keyboard focus cannot easily reach Settings panel after opening
- GUT-29 - States are not visible in High Contrast views
- GUT-27 - Content Structure dropdown remains open after focus moves to page
- GUT-16 - Radio button automatically changes the context on user focus
- GUT-12 - Date values in the datepicker are not fully parsed
- GUT-7 - Focus not visible on Preview button, and low contrast on various buttons
- GUT-4 - Mobile resolution and page zoom will cause content to be hidden or a horizontal scroll to show
- GUT-3 - Buttons vanish at larger zoom levels and on mobile

Issues By Content Category

This section allows you to browse issues organised by the type of page content they apply to.

Color

- GUT-100 - Selected block outline has insufficient color contrast
- GUT-97 - Blocks use custom selection color
- GUT-89 - Insufficient color contrast on block type buttons
- GUT-76 - Field boundaries have insufficient color contrast
- GUT-75 - Button has insufficient color contrast
- GUT-55 - Insufficient color contrast in the options menu
- GUT-44 - Insufficient contrast on empty field borders
- GUT-43 - Insufficient contrast on link text
- GUT-31 - Focus state has low contrast
- GUT-19 - Insufficient color contrast for calendar selected date and Publish button
- GUT-7 - Focus not visible on Preview button, and low contrast on various buttons
- GUT-1 - Insufficient color contrast on Editor Top Bar

Document Structure

- GUT-113 - Users cannot add video metadata in the visual editor
- GUT-112 - Users can duplicate post titles
- GUT-103 - Users cannot add table metadata in the visual editor
- GUT-99 - Keyboard focus order doesn't match visual order at high browser zoom
- GUT-83 - Page content is an invisible Tab stop
- GUT-79 - Options vanish at higher zoom levels

- GUT-58 - Tab structure does not use tab semantics
- GUT-49 - Insufficient labelling and inappropriate nesting of form controls
- GUT-35 - Text styled as headings are not headings
- GUT-25 - Content lacks semantic markup
- GUT-14 - Suggested tags should be a list

Dynamic Content

- GUT-94 - Changes and updates are not announced to assistive technologies
- GUT-73 - Status message not exposed to assistive technologies
- GUT-63 - Live search does not announce results to assistive technologies
- GUT-42 - Feature Image dialog does not follow the dialog pattern
- GUT-41 - Text inadvertently rendered by assistive technologies
- GUT-10 - Undo and Redo functions do not communicate changes

Forms

- GUT-87 - Permalink and its Edit button could be clearer
- GUT-81 - Disabled fieldset does not convey the disabled state of controls
- GUT-74 - Radio and checkbox labels rely on implicit association
- GUT-64 - Visible labelling only done via placeholder attribute
- GUT-60 - Errors and field information are not associated with their control
- GUT-54 - Form controls are not grouped
- GUT-50 - Insufficient form semantics
- GUT-39 - Instructions are unclear
- GUT-36 - Form controls are neither grouped nor named

Frames

- GUT-104 - Citation field is only indicated by placeholder text

Images and Non-text Content

- GUT-108 - Users are able to directly link images
- GUT-84 - Undo and Redo icons are not consistent throughout the interface
- GUT-66 - Icons alone are visible for controls unless they are hovered or focussed

Keyboard Accessibility and Focus Control

- GUT-106 - Media players receive focus but show no visible focus state
- GUT-101 - Menu remains open after actions which moves focus
- GUT-95 - Keyboard operation is confusing and inconsistent
- GUT-82 - Reading and focus order do not match visual order
- GUT-80 - Drag and drop functionality unavailable to keyboard or touchscreen
- GUT-71 - Reading order does not match visual order
- GUT-70 - Single visible menu is made up of multiple menus
- GUT-68 - Custom interactions in control styled as a native control
- GUT-65 - Custom field check box automatically changes the context on user focus
- GUT-53 - Keyboard focus is not managed when dialogs close
- GUT-51 - Shortcut keys conflict with browser shortcuts
- GUT-45 - Focus is invisible in high contrast views
- GUT-33 - Keyboard focus cannot easily reach Settings panel after opening
- GUT-29 - States are not visible in High Contrast views
- GUT-27 - Content Structure dropdown remains open after focus moves to page

- GUT-26 - Focus not visible on Content Structure pane
- GUT-20 - Buttons are low contrast and focus not visible
- GUT-17 - Focus not visible on block types in High Contrast modes
- GUT-16 - Radio button automatically changes the context on user focus
- GUT-15 - Keyboard focus is not managed when dynamic content disappears
- GUT-11 - SVGs are focusable in IE
- GUT-8 - Toolbar elements are tabbable

Links and Navigation

- GUT-28 - Page content can be accessed while dropdowns are open

Name, State, Role, and Value

- GUT-105 - Current-page link is not programmatically labelled as such
- GUT-67 - Missing role on menu item
- GUT-18 - Unnecessary use of aria-label

Other

- GUT-114 - Users are not warned against autoplaying videos
- GUT-109 - Blocks are difficult to edit with speech recognition
- GUT-102 - Heading block uses a counter-intuitive icon
- GUT-88 - Image cropping requires mouse and keyboard
- GUT-78 - Cursor suggests functionality even when functionality is not present

- GUT-9 - Incorrect cursor used on buttons
- GUT-6 - Tooltips cannot be dismissed without moving the pointer or focus
- GUT-5 - Buttons are smaller than the minimum target size
- GUT-4 - Mobile resolution and page zoom will cause content to be hidden or a horizontal scroll to show
- GUT-3 - Buttons vanish at larger zoom levels and on mobile

Parsing

- GUT-12 - Date values in the datepicker are not fully parsed

Text Content and Typography

- GUT-111 - Language and text direction are not editable in the visual editor
- GUT-110 - Language cannot be set on posts
- GUT-96 - Some content becomes unreadable with text enlarge
- GUT-85 - Visible labels are not available on editing buttons
- GUT-77 - Text overlaps or is clipped when text spacing or size is enlarged
- GUT-62 - Control is invisible
- GUT-40 - Useful content is unnecessarily hidden
- GUT-32 - Content overflows and is cut off at 200% text enlarge
- GUT-21 - Content overlaps at 200% text enlarge
- GUT-13 - Plain text appears to be a link
- GUT-2 - Button text overflows button borders at 200% text enlarge

Issues By Component

This section allows you to browse issues by component. By component, we mean a sample item listed in the agreed test plan. A component can represent an entire screen, a region of a screen, such as a common header, or a complex user interface widget, such as a date picker.

Add Block

- GUT-18 - Unnecessary use of aria-label
- GUT-17 - Focus not visible on block types in High Contrast modes
- GUT-6 - Tooltips cannot be dismissed without moving the pointer or focus

Block Editing

- GUT-111 - Language and text direction are not editable in the visual editor
- GUT-109 - Blocks are difficult to edit with speech recognition
- GUT-108 - Users are able to directly link images
- GUT-104 - Citation field is only indicated by placeholder text
- GUT-103 - Users cannot add table metadata in the visual editor
- GUT-102 - Heading block uses a counter-intuitive icon
- GUT-100 - Selected block outline has insufficient color contrast
- GUT-99 - Keyboard focus order doesn't match visual order at high browser zoom
- GUT-97 - Blocks use custom selection color
- GUT-96 - Some content becomes unreadable with text enlarge
- GUT-95 - Keyboard operation is confusing and inconsistent
- GUT-94 - Changes and updates are not announced to assistive technologies
- GUT-29 - States are not visible in High Contrast views

- GUT-6 - Tooltips cannot be dismissed without moving the pointer or focus

Block Options

- GUT-101 - Menu remains open after actions which moves focus
- GUT-99 - Keyboard focus order doesn't match visual order at high browser zoom
- GUT-97 - Blocks use custom selection color
- GUT-94 - Changes and updates are not announced to assistive technologies
- GUT-6 - Tooltips cannot be dismissed without moving the pointer or focus

Block Panel

- GUT-68 - Custom interactions in control styled as a native control
- GUT-41 - Text inadvertently rendered by assistive technologies
- GUT-40 - Useful content is unnecessarily hidden
- GUT-39 - Instructions are unclear
- GUT-36 - Form controls are neither grouped nor named
- GUT-35 - Text styled as headings are not headings
- GUT-33 - Keyboard focus cannot easily reach Settings panel after opening
- GUT-32 - Content overflows and is cut off at 200% text enlarge
- GUT-31 - Focus state has low contrast
- GUT-29 - States are not visible in High Contrast views
- GUT-6 - Tooltips cannot be dismissed without moving the pointer or focus

Block Types

- GUT-114 - Users are not warned against autoplaying videos
- GUT-113 - Users cannot add video metadata in the visual editor
- GUT-43 - Insufficient contrast on link text
- GUT-6 - Tooltips cannot be dismissed without moving the pointer or focus

Document Panel

- GUT-41 - Text inadvertently rendered by assistive technologies
- GUT-36 - Form controls are neither grouped nor named
- GUT-33 - Keyboard focus cannot easily reach Settings panel after opening
- GUT-32 - Content overflows and is cut off at 200% text enlarge
- GUT-29 - States are not visible in High Contrast views
- GUT-20 - Buttons are low contrast and focus not visible
- GUT-19 - Insufficient color contrast for calendar selected date and Publish button
- GUT-16 - Radio button automatically changes the context on user focus
- GUT-12 - Date values in the datepicker are not fully parsed

Edit Media

- GUT-88 - Image cropping requires mouse and keyboard
- GUT-85 - Visible labels are not available on editing buttons
- GUT-84 - Undo and Redo icons are not consistent throughout the interface
- GUT-83 - Page content is an invisible Tab stop
- GUT-82 - Reading and focus order do not match visual order
- GUT-81 - Disabled fieldset does not convey the disabled state of controls
- GUT-80 - Drag and drop functionality unavailable to keyboard or touchscreen
- GUT-79 - Options vanish at higher zoom levels
- GUT-78 - Cursor suggests functionality even when functionality is not present

- GUT-77 - Text overlaps or is clipped when text spacing or size is enlarged
- GUT-76 - Field boundaries have insufficient color contrast
- GUT-75 - Button has insufficient color contrast
- GUT-74 - Radio and checkbox labels rely on implicit association
- GUT-73 - Status message not exposed to assistive technologies
- GUT-45 - Focus is invisible in high contrast views

Editor Top Bar

- GUT-10 - Undo and Redo functions do not communicate changes
- GUT-8 - Toolbar elements are tabbable
- GUT-7 - Focus not visible on Preview button, and low contrast on various buttons
- GUT-6 - Tooltips cannot be dismissed without moving the pointer or focus
- GUT-3 - Buttons vanish at larger zoom levels and on mobile
- GUT-2 - Button text overflows button borders at 200% text enlarge
- GUT-1 - Insufficient color contrast on Editor Top Bar

Empty Block

- GUT-89 - Insufficient color contrast on block type buttons

Global

- GUT-66 - Icons alone are visible for controls unless they are hovered or focussed
- GUT-9 - Incorrect cursor used on buttons

- GUT-5 - Buttons are smaller than the minimum target size
- GUT-4 - Mobile resolution and page zoom will cause content to be hidden or a horizontal scroll to show

Media Dialog

- GUT-74 - Radio and checkbox labels rely on implicit association
- GUT-64 - Visible labelling only done via placeholder attribute
- GUT-63 - Live search does not announce results to assistive technologies
- GUT-62 - Control is invisible
- GUT-60 - Errors and field information are not associated with their control
- GUT-58 - Tab structure does not use tab semantics
- GUT-50 - Insufficient form semantics
- GUT-49 - Insufficient labelling and inappropriate nesting of form controls
- GUT-45 - Focus is invisible in high contrast views
- GUT-44 - Insufficient contrast on empty field borders
- GUT-43 - Insufficient contrast on link text
- GUT-42 - Feature Image dialog does not follow the dialog pattern
- GUT-41 - Text inadvertently rendered by assistive technologies
- GUT-2 - Button text overflows button borders at 200% text enlarge

Output Markup

- GUT-114 - Users are not warned against autoplaying videos
- GUT-113 - Users cannot add video metadata in the visual editor
- GUT-112 - Users can duplicate post titles
- GUT-110 - Language cannot be set on posts
- GUT-106 - Media players receive focus but show no visible focus state

- GUT-105 - Current-page link is not programmatically labelled as such
- GUT-103 - Users cannot add table metadata in the visual editor
- GUT-11 - SVGs are focusable in IE

Publish and Unpublish

- GUT-29 - States are not visible in High Contrast views
- GUT-20 - Buttons are low contrast and focus not visible
- GUT-19 - Insufficient color contrast for calendar selected date and Publish button
- GUT-16 - Radio button automatically changes the context on user focus
- GUT-15 - Keyboard focus is not managed when dynamic content disappears
- GUT-14 - Suggested tags should be a list
- GUT-13 - Plain text appears to be a link
- GUT-12 - Date values in the datepicker are not fully parsed
- GUT-11 - SVGs are focusable in IE

Structure and Navigation

- GUT-28 - Page content can be accessed while dropdowns are open
- GUT-27 - Content Structure dropdown remains open after focus moves to page
- GUT-26 - Focus not visible on Content Structure pane
- GUT-25 - Content lacks semantic markup
- GUT-21 - Content overlaps at 200% text enlarge
- GUT-6 - Tooltips cannot be dismissed without moving the pointer or focus
- GUT-3 - Buttons vanish at larger zoom levels and on mobile

Title Block

- GUT-111 - Language and text direction are not editable in the visual editor
- GUT-109 - Blocks are difficult to edit with speech recognition
- GUT-87 - Permalink and its Edit button could be clearer
- GUT-35 - Text styled as headings are not headings

Tools and Options

- GUT-71 - Reading order does not match visual order
- GUT-70 - Single visible menu is made up of multiple menus
- GUT-67 - Missing role on menu item
- GUT-65 - Custom field check box automatically changes the context on user focus
- GUT-55 - Insufficient color contrast in the options menu
- GUT-54 - Form controls are not grouped
- GUT-53 - Keyboard focus is not managed when dialogs close
- GUT-51 - Shortcut keys conflict with browser shortcuts

Issues By Affected Population

This section allows you to browse issues by population of people with disabilities affected. Some issues will affect more than one disability group.

Blind

Blindness represents complete or almost complete vision loss. Causes of blindness include cataracts, glaucoma, and age related macular degeneration. Although many blind people retain some degree of vision, it is often not mature enough to complete everyday tasks such as interacting with the Web. Therefore, people who are blind often navigate the Web using screen reader software, as well as a keyboard rather than a mouse.

- GUT-114 - Users are not warned against autoplaying videos
- GUT-113 - Users cannot add video metadata in the visual editor
- GUT-112 - Users can duplicate post titles
- GUT-111 - Language and text direction are not editable in the visual editor
- GUT-110 - Language cannot be set on posts
- GUT-108 - Users are able to directly link images
- GUT-103 - Users cannot add table metadata in the visual editor
- GUT-95 - Keyboard operation is confusing and inconsistent
- GUT-94 - Changes and updates are not announced to assistive technologies
- GUT-88 - Image cropping requires mouse and keyboard
- GUT-87 - Permalink and its Edit button could be clearer
- GUT-83 - Page content is an invisible Tab stop
- GUT-82 - Reading and focus order do not match visual order
- GUT-81 - Disabled fieldset does not convey the disabled state of controls
- GUT-80 - Drag and drop functionality unavailable to keyboard or touchscreen
- GUT-73 - Status message not exposed to assistive technologies
- GUT-70 - Single visible menu is made up of multiple menus
- GUT-67 - Missing role on menu item

- GUT-65 - Custom field check box automatically changes the context on user focus
- GUT-63 - Live search does not announce results to assistive technologies
- GUT-60 - Errors and field information are not associated with their control
- GUT-58 - Tab structure does not use tab semantics
- GUT-54 - Form controls are not grouped
- GUT-53 - Keyboard focus is not managed when dialogs close
- GUT-50 - Insufficient form semantics
- GUT-49 - Insufficient labelling and inappropriate nesting of form controls
- GUT-41 - Text inadvertently rendered by assistive technologies
- GUT-36 - Form controls are neither grouped nor named
- GUT-35 - Text styled as headings are not headings
- GUT-33 - Keyboard focus cannot easily reach Settings panel after opening
- GUT-28 - Page content can be accessed while dropdowns are open
- GUT-25 - Content lacks semantic markup
- GUT-18 - Unnecessary use of aria-label
- GUT-16 - Radio button automatically changes the context on user focus
- GUT-15 - Keyboard focus is not managed when dynamic content disappears
- GUT-14 - Suggested tags should be a list
- GUT-12 - Date values in the datepicker are not fully parsed
- GUT-11 - SVGs are focusable in IE
- GUT-10 - Undo and Redo functions do not communicate changes
- GUT-8 - Toolbar elements are tabbable

Cognitively Impaired

Cognitive impairment is a very broad area, but typically a person with a cognitive impairment may have difficulties with reading, attention, problem solving, or the ability to retain information over time.

- GUT-114 - Users are not warned against autoplaying videos
- GUT-113 - Users cannot add video metadata in the visual editor
- GUT-112 - Users can duplicate post titles
- GUT-111 - Language and text direction are not editable in the visual editor

- GUT-110 - Language cannot be set on posts
- GUT-108 - Users are able to directly link images
- GUT-106 - Media players receive focus but show no visible focus state
- GUT-105 - Current-page link is not programmatically labelled as such
- GUT-104 - Citation field is only indicated by placeholder text
- GUT-103 - Users cannot add table metadata in the visual editor
- GUT-102 - Heading block uses a counter-intuitive icon
- GUT-101 - Menu remains open after actions which moves focus
- GUT-100 - Selected block outline has insufficient color contrast
- GUT-99 - Keyboard focus order doesn't match visual order at high browser zoom
- GUT-95 - Keyboard operation is confusing and inconsistent
- GUT-94 - Changes and updates are not announced to assistive technologies
- GUT-89 - Insufficient color contrast on block type buttons
- GUT-88 - Image cropping requires mouse and keyboard
- GUT-87 - Permalink and its Edit button could be clearer
- GUT-85 - Visible labels are not available on editing buttons
- GUT-84 - Undo and Redo icons are not consistent throughout the interface
- GUT-83 - Page content is an invisible Tab stop
- GUT-82 - Reading and focus order do not match visual order
- GUT-81 - Disabled fieldset does not convey the disabled state of controls
- GUT-79 - Options vanish at higher zoom levels
- GUT-78 - Cursor suggests functionality even when functionality is not present
- GUT-77 - Text overlaps or is clipped when text spacing or size is enlarged
- GUT-76 - Field boundaries have insufficient color contrast
- GUT-74 - Radio and checkbox labels rely on implicit association
- GUT-73 - Status message not exposed to assistive technologies
- GUT-71 - Reading order does not match visual order
- GUT-70 - Single visible menu is made up of multiple menus
- GUT-68 - Custom interactions in control styled as a native control
- GUT-67 - Missing role on menu item
- GUT-66 - Icons alone are visible for controls unless they are hovered or focussed
- GUT-65 - Custom field check box automatically changes the context on user focus
- GUT-64 - Visible labelling only done via placeholder attribute
- GUT-63 - Live search does not announce results to assistive technologies
- GUT-62 - Control is invisible
- GUT-60 - Errors and field information are not associated with their control

- GUT-58 - Tab structure does not use tab semantics
- GUT-55 - Insufficient color contrast in the options menu
- GUT-54 - Form controls are not grouped
- GUT-53 - Keyboard focus is not managed when dialogs close
- GUT-51 - Shortcut keys conflict with browser shortcuts
- GUT-50 - Insufficient form semantics
- GUT-49 - Insufficient labelling and inappropriate nesting of form controls
- GUT-45 - Focus is invisible in high contrast views
- GUT-44 - Insufficient contrast on empty field borders
- GUT-43 - Insufficient contrast on link text
- GUT-41 - Text inadvertently rendered by assistive technologies
- GUT-40 - Useful content is unnecessarily hidden
- GUT-39 - Instructions are unclear
- GUT-36 - Form controls are neither grouped nor named
- GUT-35 - Text styled as headings are not headings
- GUT-33 - Keyboard focus cannot easily reach Settings panel after opening
- GUT-32 - Content overflows and is cut off at 200% text enlarge
- GUT-31 - Focus state has low contrast
- GUT-29 - States are not visible in High Contrast views
- GUT-28 - Page content can be accessed while dropdowns are open
- GUT-27 - Content Structure dropdown remains open after focus moves to page
- GUT-26 - Focus not visible on Content Structure pane
- GUT-25 - Content lacks semantic markup
- GUT-21 - Content overlaps at 200% text enlarge
- GUT-20 - Buttons are low contrast and focus not visible
- GUT-19 - Insufficient color contrast for calendar selected date and Publish button
- GUT-18 - Unnecessary use of aria-label
- GUT-16 - Radio button automatically changes the context on user focus
- GUT-15 - Keyboard focus is not managed when dynamic content disappears
- GUT-14 - Suggested tags should be a list
- GUT-13 - Plain text appears to be a link
- GUT-12 - Date values in the datepicker are not fully parsed
- GUT-11 - SVGs are focusable in IE
- GUT-10 - Undo and Redo functions do not communicate changes
- GUT-9 - Incorrect cursor used on buttons
- GUT-8 - Toolbar elements are tabbable

- GUT-7 - Focus not visible on Preview button, and low contrast on various buttons
- GUT-6 - Tooltips cannot be dismissed without moving the pointer or focus
- GUT-4 - Mobile resolution and page zoom will cause content to be hidden or a horizontal scroll to show
- GUT-3 - Buttons vanish at larger zoom levels and on mobile
- GUT-2 - Button text overflows button borders at 200% text enlarge
- GUT-1 - Insufficient color contrast on Editor Top Bar

Low-Vision

- GUT-114 - Users are not warned against autoplaying videos
- GUT-113 - Users cannot add video metadata in the visual editor
- GUT-112 - Users can duplicate post titles
- GUT-111 - Language and text direction are not editable in the visual editor
- GUT-110 - Language cannot be set on posts
- GUT-108 - Users are able to directly link images
- GUT-106 - Media players receive focus but show no visible focus state
- GUT-105 - Current-page link is not programmatically labelled as such
- GUT-104 - Citation field is only indicated by placeholder text
- GUT-103 - Users cannot add table metadata in the visual editor
- GUT-102 - Heading block uses a counter-intuitive icon
- GUT-101 - Menu remains open after actions which moves focus
- GUT-100 - Selected block outline has insufficient color contrast
- GUT-99 - Keyboard focus order doesn't match visual order at high browser zoom
- GUT-97 - Blocks use custom selection color
- GUT-96 - Some content becomes unreadable with text enlarge
- GUT-95 - Keyboard operation is confusing and inconsistent
- GUT-94 - Changes and updates are not announced to assistive technologies
- GUT-89 - Insufficient color contrast on block type buttons
- GUT-88 - Image cropping requires mouse and keyboard
- GUT-85 - Visible labels are not available on editing buttons
- GUT-83 - Page content is an invisible Tab stop

- GUT-82 - Reading and focus order do not match visual order
- GUT-81 - Disabled fieldset does not convey the disabled state of controls
- GUT-79 - Options vanish at higher zoom levels
- GUT-78 - Cursor suggests functionality even when functionality is not present
- GUT-77 - Text overlaps or is clipped when text spacing or size is enlarged
- GUT-76 - Field boundaries have insufficient color contrast
- GUT-75 - Button has insufficient color contrast
- GUT-73 - Status message not exposed to assistive technologies
- GUT-71 - Reading order does not match visual order
- GUT-70 - Single visible menu is made up of multiple menus
- GUT-68 - Custom interactions in control styled as a native control
- GUT-67 - Missing role on menu item
- GUT-65 - Custom field check box automatically changes the context on user focus
- GUT-64 - Visible labelling only done via placeholder attribute
- GUT-63 - Live search does not announce results to assistive technologies
- GUT-62 - Control is invisible
- GUT-60 - Errors and field information are not associated with their control
- GUT-58 - Tab structure does not use tab semantics
- GUT-55 - Insufficient color contrast in the options menu
- GUT-54 - Form controls are not grouped
- GUT-53 - Keyboard focus is not managed when dialogs close
- GUT-51 - Shortcut keys conflict with browser shortcuts
- GUT-49 - Insufficient labelling and inappropriate nesting of form controls
- GUT-45 - Focus is invisible in high contrast views
- GUT-44 - Insufficient contrast on empty field borders
- GUT-43 - Insufficient contrast on link text
- GUT-41 - Text inadvertently rendered by assistive technologies
- GUT-33 - Keyboard focus cannot easily reach Settings panel after opening
- GUT-32 - Content overflows and is cut off at 200% text enlarge
- GUT-31 - Focus state has low contrast
- GUT-29 - States are not visible in High Contrast views
- GUT-27 - Content Structure dropdown remains open after focus moves to page
- GUT-26 - Focus not visible on Content Structure pane
- GUT-25 - Content lacks semantic markup
- GUT-21 - Content overlaps at 200% text enlarge
- GUT-20 - Buttons are low contrast and focus not visible

- GUT-19 - Insufficient color contrast for calendar selected date and Publish button
- GUT-18 - Unnecessary use of aria-label
- GUT-17 - Focus not visible on block types in High Contrast modes
- GUT-16 - Radio button automatically changes the context on user focus
- GUT-15 - Keyboard focus is not managed when dynamic content disappears
- GUT-14 - Suggested tags should be a list
- GUT-13 - Plain text appears to be a link
- GUT-12 - Date values in the datepicker are not fully parsed
- GUT-11 - SVGs are focusable in IE
- GUT-10 - Undo and Redo functions do not communicate changes
- GUT-9 - Incorrect cursor used on buttons
- GUT-7 - Focus not visible on Preview button, and low contrast on various buttons
- GUT-6 - Tooltips cannot be dismissed without moving the pointer or focus
- GUT-5 - Buttons are smaller than the minimum target size
- GUT-4 - Mobile resolution and page zoom will cause content to be hidden or a horizontal scroll to show
- GUT-3 - Buttons vanish at larger zoom levels and on mobile
- GUT-2 - Button text overflows button borders at 200% text enlarge
- GUT-1 - Insufficient color contrast on Editor Top Bar

Motor Impaired

Motor impairment relates to the partial or total loss of function of a person's body part or parts. Examples include the loss of limbs, a spinal cord injury, arthritis, Parkinson's disease, or a congenital condition such as cerebral palsy. This can result in muscle weakness, lack of muscle control, or complete paralysis. Assistive technologies for people with motor impairments include voice recognition software, eye tracking devices, switches and head wands.

- GUT-114 - Users are not warned against autoplaying videos
- GUT-109 - Blocks are difficult to edit with speech recognition
- GUT-106 - Media players receive focus but show no visible focus state
- GUT-101 - Menu remains open after actions which moves focus
- GUT-99 - Keyboard focus order doesn't match visual order at high browser zoom

- GUT-95 - Keyboard operation is confusing and inconsistent
- GUT-88 - Image cropping requires mouse and keyboard
- GUT-83 - Page content is an invisible Tab stop
- GUT-81 - Disabled fieldset does not convey the disabled state of controls
- GUT-80 - Drag and drop functionality unavailable to keyboard or touchscreen
- GUT-74 - Radio and checkbox labels rely on implicit association
- GUT-68 - Custom interactions in control styled as a native control
- GUT-66 - Icons alone are visible for controls unless they are hovered or focussed
- GUT-65 - Custom field check box automatically changes the context on user focus
- GUT-64 - Visible labelling only done via placeholder attribute
- GUT-62 - Control is invisible
- GUT-58 - Tab structure does not use tab semantics
- GUT-53 - Keyboard focus is not managed when dialogs close
- GUT-51 - Shortcut keys conflict with browser shortcuts
- GUT-45 - Focus is invisible in high contrast views
- GUT-40 - Useful content is unnecessarily hidden
- GUT-33 - Keyboard focus cannot easily reach Settings panel after opening
- GUT-31 - Focus state has low contrast
- GUT-29 - States are not visible in High Contrast views
- GUT-28 - Page content can be accessed while dropdowns are open
- GUT-27 - Content Structure dropdown remains open after focus moves to page
- GUT-26 - Focus not visible on Content Structure pane
- GUT-20 - Buttons are low contrast and focus not visible
- GUT-17 - Focus not visible on block types in High Contrast modes
- GUT-16 - Radio button automatically changes the context on user focus
- GUT-15 - Keyboard focus is not managed when dynamic content disappears
- GUT-13 - Plain text appears to be a link
- GUT-11 - SVGs are focusable in IE
- GUT-8 - Toolbar elements are tabbable
- GUT-7 - Focus not visible on Preview button, and low contrast on various buttons
- GUT-6 - Tooltips cannot be dismissed without moving the pointer or focus
- GUT-5 - Buttons are smaller than the minimum target size
- GUT-4 - Mobile resolution and page zoom will cause content to be hidden or a horizontal scroll to show

Issues By WCAG Success Criteria

This section allows you to view issues organized by relevant WCAG Success Criteria. Success criteria are presented in numerical order, and each success criterion has an indication of its conformance level (A or AA). **Note:** A success criterion may have more than one associated issue, and an issue may have more than one relevant success criterion.

1.2.2 Captions (Prerecorded) (Level A)

- GUT-113 - Users cannot add video metadata in the visual editor

1.3.1 Info and Relationships (Level A)

- GUT-87 - Permalink and its Edit button could be clearer
- GUT-70 - Single visible menu is made up of multiple menus
- GUT-60 - Errors and field information are not associated with their control
- GUT-58 - Tab structure does not use tab semantics
- GUT-50 - Insufficient form semantics
- GUT-49 - Insufficient labelling and inappropriate nesting of form controls
- GUT-42 - Feature Image dialog does not follow the dialog pattern
- GUT-36 - Form controls are neither grouped nor named
- GUT-35 - Text styled as headings are not headings
- GUT-25 - Content lacks semantic markup
- GUT-14 - Suggested tags should be a list
- GUT-10 - Undo and Redo functions do not communicate changes

1.3.2 Meaningful Sequence (Level A)

- GUT-82 - Reading and focus order do not match visual order
- GUT-71 - Reading order does not match visual order

1.3.3 Sensory Characteristics (Level A)

- GUT-102 - Heading block uses a counter-intuitive icon
- GUT-97 - Blocks use custom selection color
- GUT-81 - Disabled fieldset does not convey the disabled state of controls
- GUT-62 - Control is invisible
- GUT-60 - Errors and field information are not associated with their control
- GUT-50 - Insufficient form semantics
- GUT-29 - States are not visible in High Contrast views
- GUT-28 - Page content can be accessed while dropdowns are open
- GUT-13 - Plain text appears to be a link
- GUT-10 - Undo and Redo functions do not communicate changes

1.4.1 Use of Color (Level A)

- GUT-20 - Buttons are low contrast and focus not visible
- GUT-7 - Focus not visible on Preview button, and low contrast on various buttons

1.4.10 Reflow (Level AA)

- GUT-99 - Keyboard focus order doesn't match visual order at high browser zoom
- GUT-79 - Options vanish at higher zoom levels
- GUT-33 - Keyboard focus cannot easily reach Settings panel after opening
- GUT-4 - Mobile resolution and page zoom will cause content to be hidden or a horizontal scroll to show
- GUT-3 - Buttons vanish at larger zoom levels and on mobile

1.4.11 Non-text Contrast (Level AA)

- GUT-100 - Selected block outline has insufficient color contrast
- GUT-76 - Field boundaries have insufficient color contrast
- GUT-44 - Insufficient contrast on empty field borders
- GUT-31 - Focus state has low contrast
- GUT-20 - Buttons are low contrast and focus not visible
- GUT-7 - Focus not visible on Preview button, and low contrast on various buttons

1.4.12 Text Spacing (Level AA)

- GUT-77 - Text overlaps or is clipped when text spacing or size is enlarged

1.4.13 Content on Hover or Focus (Level AA)

- GUT-6 - Tooltips cannot be dismissed without moving the pointer or focus

1.4.3 Contrast (Minimum) (Level AA)

- GUT-89 - Insufficient color contrast on block type buttons
- GUT-75 - Button has insufficient color contrast
- GUT-55 - Insufficient color contrast in the options menu
- GUT-43 - Insufficient contrast on link text
- GUT-19 - Insufficient color contrast for calendar selected date and Publish button
- GUT-1 - Insufficient color contrast on Editor Top Bar

1.4.4 Resize text (Level AA)

- GUT-96 - Some content becomes unreadable with text enlarge
- GUT-77 - Text overlaps or is clipped when text spacing or size is enlarged
- GUT-32 - Content overflows and is cut off at 200% text enlarge
- GUT-21 - Content overlaps at 200% text enlarge
- GUT-2 - Button text overflows button borders at 200% text enlarge

2.1.1 Keyboard (Level A)

- GUT-106 - Media players receive focus but show no visible focus state
- GUT-101 - Menu remains open after actions which moves focus
- GUT-99 - Keyboard focus order doesn't match visual order at high browser zoom
- GUT-95 - Keyboard operation is confusing and inconsistent

- GUT-88 - Image cropping requires mouse and keyboard
- GUT-83 - Page content is an invisible Tab stop
- GUT-80 - Drag and drop functionality unavailable to keyboard or touchscreen
- GUT-73 - Status message not exposed to assistive technologies
- GUT-70 - Single visible menu is made up of multiple menus
- GUT-68 - Custom interactions in control styled as a native control
- GUT-53 - Keyboard focus is not managed when dialogs close
- GUT-15 - Keyboard focus is not managed when dynamic content disappears
- GUT-8 - Toolbar elements are tabbable

2.4.1 Bypass Blocks (Level A)

- GUT-95 - Keyboard operation is confusing and inconsistent

2.4.2 Page Titled (Level A)

- GUT-112 - Users can duplicate post titles

2.4.3 Focus Order (Level A)

- GUT-99 - Keyboard focus order doesn't match visual order at high browser zoom
- GUT-95 - Keyboard operation is confusing and inconsistent
- GUT-82 - Reading and focus order do not match visual order
- GUT-33 - Keyboard focus cannot easily reach Settings panel after opening

2.4.4 Link Purpose (In Context) (Level A)

- GUT-108 - Users are able to directly link images
- GUT-105 - Current-page link is not programmatically labelled as such

2.4.6 Headings and Labels (Level AA)

- GUT-35 - Text styled as headings are not headings
- GUT-25 - Content lacks semantic markup

2.4.7 Focus Visible (Level AA)

- GUT-106 - Media players receive focus but show no visible focus state
- GUT-101 - Menu remains open after actions which moves focus
- GUT-83 - Page content is an invisible Tab stop
- GUT-62 - Control is invisible
- GUT-45 - Focus is invisible in high contrast views
- GUT-33 - Keyboard focus cannot easily reach Settings panel after opening
- GUT-31 - Focus state has low contrast
- GUT-29 - States are not visible in High Contrast views
- GUT-27 - Content Structure dropdown remains open after focus moves to page
- GUT-26 - Focus not visible on Content Structure pane
- GUT-20 - Buttons are low contrast and focus not visible
- GUT-17 - Focus not visible on block types in High Contrast modes
- GUT-7 - Focus not visible on Preview button, and low contrast on various buttons

2.4.8 Location (Level AAA)

- GUT-105 - Current-page link is not programmatically labelled as such

2.5.5 Target Size (Level AAA)

- GUT-62 - Control is invisible
- GUT-5 - Buttons are smaller than the minimum target size

2.5.6 Concurrent Input Mechanisms (Level AAA)

- GUT-109 - Blocks are difficult to edit with speech recognition
- GUT-88 - Image cropping requires mouse and keyboard
- GUT-80 - Drag and drop functionality unavailable to keyboard or touchscreen

3.1.1 Language of Page (Level A)

- GUT-110 - Language cannot be set on posts

3.1.2 Language of Parts (Level AA)

- GUT-111 - Language and text direction are not editable in the visual editor
- GUT-110 - Language cannot be set on posts

3.2.2 On Input (Level A)

- GUT-16 - Radio button automatically changes the context on user focus

3.2.4 Consistent Identification (Level AA)

- GUT-84 - Undo and Redo icons are not consistent throughout the interface

3.2.5 Change on Request (Level AAA)

- GUT-94 - Changes and updates are not announced to assistive technologies
- GUT-65 - Custom field check box automatically changes the context on user focus
- GUT-16 - Radio button automatically changes the context on user focus

3.3.1 Error Identification (Level A)

- GUT-60 - Errors and field information are not associated with their control

3.3.2 Labels or Instructions (Level A)

- GUT-87 - Permalink and its Edit button could be clearer
- GUT-60 - Errors and field information are not associated with their control
- GUT-54 - Form controls are not grouped
- GUT-50 - Insufficient form semantics
- GUT-39 - Instructions are unclear
- GUT-36 - Form controls are neither grouped nor named

4.1.2 Name, Role, Value (Level A)

- GUT-70 - Single visible menu is made up of multiple menus
- GUT-67 - Missing role on menu item
- GUT-58 - Tab structure does not use tab semantics
- GUT-50 - Insufficient form semantics
- GUT-42 - Feature Image dialog does not follow the dialog pattern
- GUT-36 - Form controls are neither grouped nor named
- GUT-35 - Text styled as headings are not headings
- GUT-25 - Content lacks semantic markup

4.1.3 Status Messages (Level AA)

- GUT-94 - Changes and updates are not announced to assistive technologies

- GUT-73 - Status message not exposed to assistive technologies
- GUT-63 - Live search does not announce results to assistive technologies
- GUT-60 - Errors and field information are not associated with their control

Tenon, LLC TalkToUs@tenon.io

309 Ferndale Road
Glen Burnie, MD 21061 United States

+1 443.875.7343