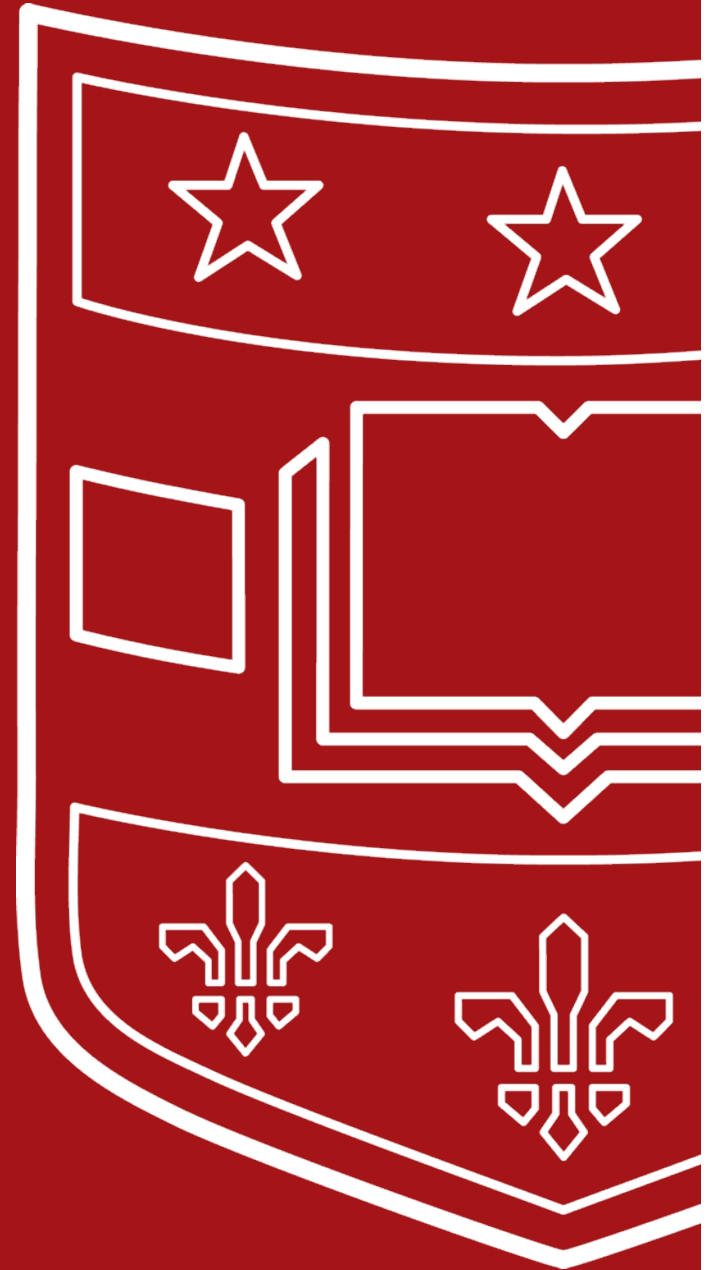


# What's New in WCAG 2.1

An overview



# WCAG Introduction



## Web Content Accessibility Guidelines

- Guidelines to help make web content more accessible to people with disabilities.
- Developed by the Website Accessibility Initiative (WAI)
- The WAI is part of the World Wide Web Consortium (W3C). W3C develops web standards including HTML and CSS.





# WCAG Version History

- **WCAG 1.0** published in May 1999
  - Focused on HTML
- Apple iPhone 1 Released June 2007
- **WCAG 2.0** published December 2008
  - Technology agnostic
    - HTML, PDF, Word Doc, etc.
- **WCAG 2.1** finalized June 5<sup>th</sup>, 2018



# WCAG Conformance Levels

- A (minimum)
- **AA**
- AAA





# Conformance Levels Examples (WCAG 2.0)

- Level A

- All functionality of the content is operable through a keyboard.
- Web pages have titles that describe topic of purpose.
- All images have appropriate alternative text.

# Conformance Levels Examples (WCAG 2.0) part 2



- Level AA

- Headings and labels describe topic or section purpose.
- Text has a contrast ratio of at least 4.5:1.
- Any keyboard operable interface (link) has a mode where focus indicator is visible.

# Conformance Levels Examples (WCAG 2.0) part 3



- **Level AAA**

- Text contrast ratio increased to 7:1 (AA 4.5:1).
- Provide a mechanism for identifying expanded forms of abbreviations.
- Images of text are used only for pure decoration.

# WCAG 2.1



- After 4 years of work, WCAG 2.1 was released on June 5<sup>th</sup>, 2018.
- All 63 success criteria from WCAG 2.0 were grandfathered into 2.1.
- Updates focuses on:
  - Mobility
  - Cognitive
  - Low Vision



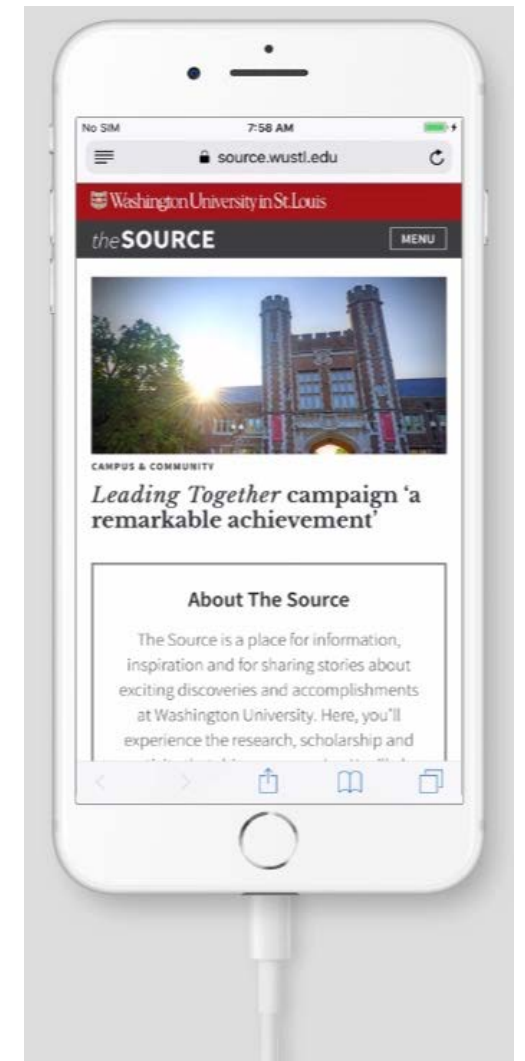
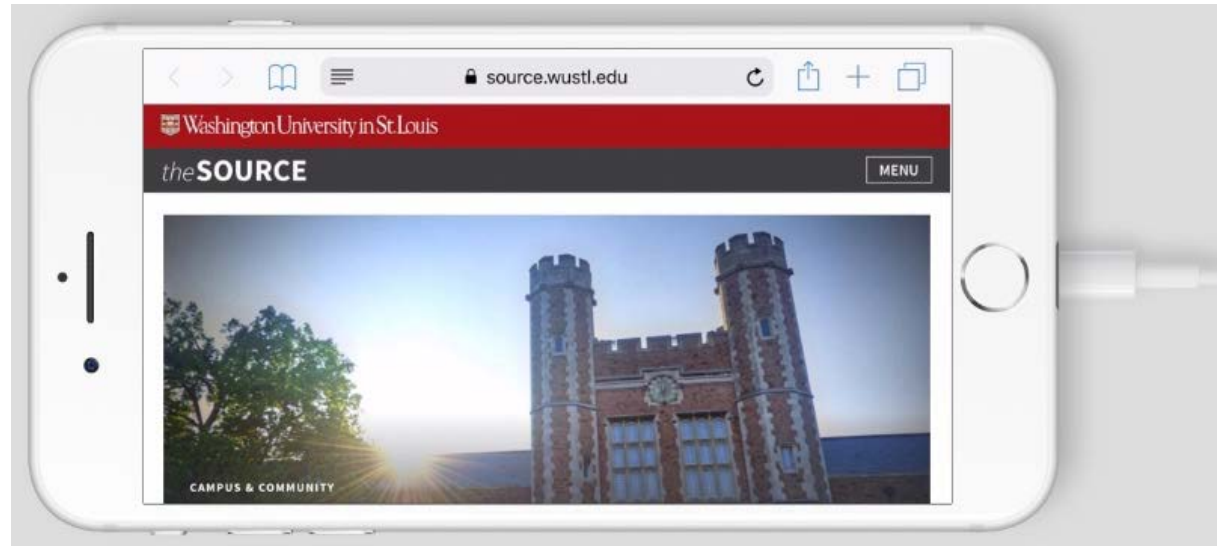
# New Level A and AA Success Criteria





## 1.3.4 Orientation (AA)

- Access and viewing of content should not be restricted to a particular orientation (landscape or portrait).





## 1.3.5 Identify Input Purpose (AA)

- Form input fields can be ‘determined programmatically’. Software, such as a browser, can tell what is expected to be entered by the user or the meaning of the data requested.
  - Example: AutoComplete



```
<form>
<label for="input-email">Email address</label>
<input id="input-email" autocomplete="email" type="email">
<label for="input-password">Password</label>
<input id="input-password" autocomplete="current-password"
type="password">
<button name="button-sign-in">Sign in</button>
</form>
```



## 1.4.10 Reflow (AA)

- Content can be enlarged up to 400% without enabling two axis scrolling.
  - Responsively designed sites should have only one scroll bar.
    - When possible avoid designs requiring horizontal scrolling.
  - Exceptions could be maps, images, presentations, etc.



## 1.4.11 Non-Text Contrast (AA)

The following elements should have a color contrast ratio of 3:1 against neighboring color:

- User interface controls
  - Form fields
  - Links
  - Buttons
- Placeholder text
- Section of graphics required to understand content



## 1.4.12 Text Spacing

No loss of content or functionality occurs by setting all of the following and by changing no other style property:

- Line height (line spacing) to at least 1.5 times the font size;
- Spacing following paragraphs to at least 2 times the font size;
- Letter spacing (tracking) to at least 0.12 times the font size;
- Word spacing to at least 0.16 times the font size.



## 1.4.12 Text Spacing – part 2

Distances between paragraphs, rows, words and characters must be able to be increased to certain values without impacting functionality or loss or content.



## 1.4.13 Content on Hover or Focus (AA)

Where receiving and then removing pointer hover or keyboard focus triggers additional content to become visible and then hidden, the following are true:

- Dismissible
- Hoverable
- Persistent





## 1.4.13 Content on Hover or Focus (AA) – Part 2

Additional Content can include:

- Custom Tooltips
- Sub-menus
- Pop-ups

The diagram illustrates two scenarios for content on hover or focus, labeled 'Fail' and 'Pass'.

**Fail:** Shows a vertical stack of three buttons. The top button is white with the text 'Option 1'. The middle button is grey with the text 'Option 2'. The bottom button is white with the text 'Option 3'. A yellow hand cursor is hovering over the 'Option 2' button, and a tooltip box containing the text 'Tooltip' is positioned over the bottom button.

**Pass:** Shows a vertical stack of three buttons. The top button is white with the text 'Option 1'. The middle button is grey with the text 'Option 2'. The bottom button is white with the text 'Option 3'. A yellow hand cursor is hovering over the 'Option 2' button, and a tooltip box containing the text 'Tooltip X' is positioned to the right of the button.

© Siteimprove



## 2.1.4 Character Key Short Cuts (A)

If a keyboard shortcut is implemented using a single key then at least one of the following is true:

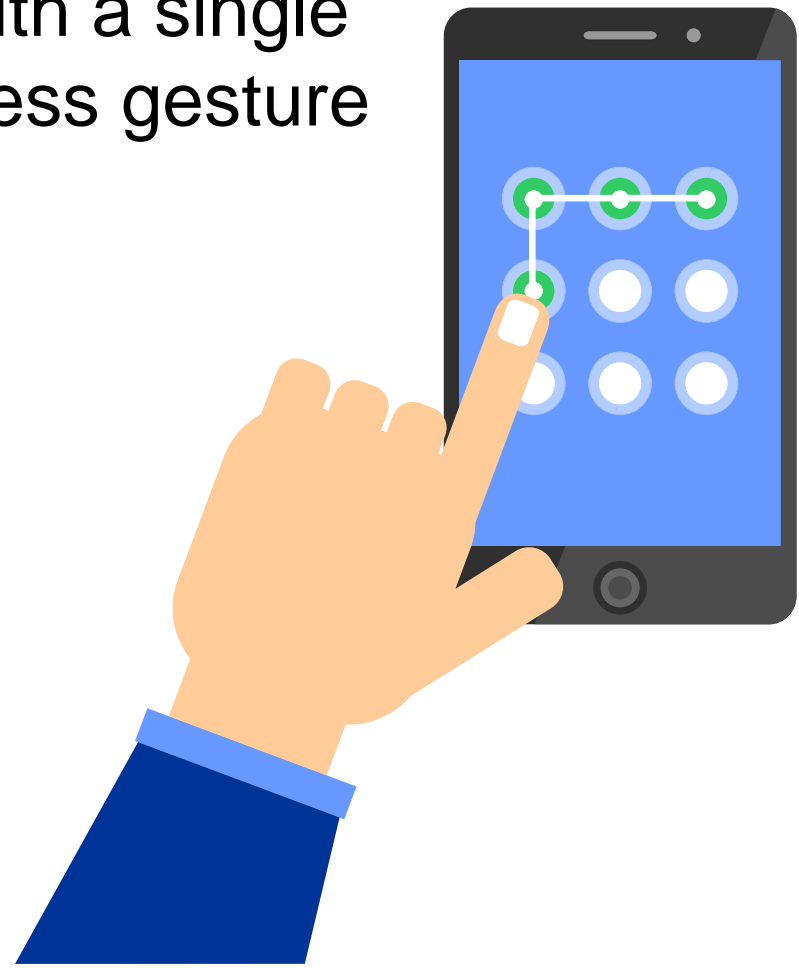
- Turn off: A mechanism is available to disable the short cut.
- Remap: A mechanism is available to remap the shortcut to use one or more non-printable keyboard characters (ex: Ctrl, Alt, etc.)
- Active only on focus: Shortcut is only active when the component has focus.





## 2.5.1 Pointer Gestures (AA)

Functionality that uses multipoint or path-based gestures for operation can be operated with a single pointer without a path-based gesture, unless gesture is essential.





## 2.5.2 Pointer Cancellation (A)

Requires the use of the standard *up-event* to trigger an interactive component.

- *Point*
- *Touch (down)*
- *Hold*
- *Release (up)*

Allows user to move away from a wrong target to prevent activating.



## 2.5.3 Label in Name (A)

On screen visible label matches the programmatic version of the label.

### *Bad Example*

```
<button class = "btn" aria-label = "Get Assistance" >  
    Help  
</button>
```

### *Good Example*

```
<button class = "btn" >  
    Help  
</button>
```



## 2.5.4 Motion Actuation

Any functionality activated by actions, such as shaking or tilting, can also be usable with interface components and be disabled.





## 4.1.3 Status Messages (AA)

For live updates (ex: status messages), use aria-live roles or attributes to notify Assistive Technology users when something on a page changes.

```
<div role="status" aria-live="off">  
...  
</div>  
<div role="status" aria-live="polite">  
...  
</div>  
<div role="status" aria-live="assertive">  
...  
</div>
```



# Level AAA New Success Criteria

- 2.5.3 Target Size - pointer minimum of 44 x 44 pixels
- 2.3.3 Animations from Interactions – refrain from using animations (motions) resulting from a user activating something.
- 2.2.6 Timeouts – users warned of any inactivity which could result in loss of data (unless over 20 hours).
- 2.5.6 Concurrent input mechanisms - don't disallow users from using concurrent inputs (mouse, keyboard, stylus, touch inputs).
- 1.3.6 Identify Purpose – interface components, icons, and sections are able to be determined programmatically.



# Questions?



[Accessibility Resources](#)