

Web Accessibility Training



Jon Whiting & Jared Smith
WebAIM.org

Resources online at:

webaim.org/presentations/2020/gep

Outline

Day 1

- Overview of WCAG 2
- Brief overview of law
- WAVE
- User experience
 - Deaf & hard of hearing
 - Low vision
 - Colorblind
 - Blind (& other screen reader users)

Day 2

- User experience, contd.
 - Photosensitive epilepsy
 - Motor disabilities
 - Cognitive and learning disabilities
- Rules of ARIA use
- Advanced forms
- JavaScript
- ARIA roles, states, and properties

Is your web content currently
accessible?

WCAG 2

- Web Content Accessibility Guidelines
- Versions 1.0 (1999), 2.0 (2008), and 2.1 (2018)
- Foundation of most accessibility laws worldwide
- Principles based
 - Perceivable
 - Operable
 - Understandable
 - Robust

Principle 1 – Perceivable

Information and user interface components must be presentable to users in ways they can perceive.

Guideline 1.1 – Text Alternatives

Provide text alternatives for any non-text content so that it can be changed into other forms people need, such as large print, braille, speech, symbols or simpler language.

[» Show techniques and failures for 1.1](#)

1.1.1 Non-text Content — Level A

All non-text content that is presented to the user has a text alternative that serves the equivalent purpose, except for the situations listed below. [» Show full description](#)

 Understanding 1.1.1

[» Show techniques and failures for 1.1.1](#)

 SHARE |  BACK TO TOP

Guideline 1.2 – Time-based Media

Provide alternatives for time-based media.

1.2.1 Audio-only and Video-only (Prerecorded) — Level A

For prerecorded audio-only and prerecorded video-only media, the following are true, except when the audio or video is a media alternative for text and is clearly labeled as such: [» Show full description](#)

 Understanding 1.2.1

Principle 1: Perceivable

Web content is made available to the senses - sight, hearing, and/or touch

Guideline 1.1 Text Alternatives

Provide text alternatives for any non-text content

Success Criteria	Recommendations
1.1.1 Non-text Content (Level A)	<ul style="list-style-type: none"><input type="checkbox"/> All images, form image buttons, and image map hot spots have appropriate, equivalent alternative text.<input type="checkbox"/> Images that do not convey content, are decorative, or contain content that is already conveyed in text are given null alt text (alt="") or implemented as CSS backgrounds. All linked images have descriptive alternative text.<input type="checkbox"/> Equivalent alternatives to complex images are provided in context or on a separate linked page.<input type="checkbox"/> Form buttons have a descriptive value.<input type="checkbox"/> Form inputs have associated text labels.<input type="checkbox"/> Embedded multimedia is identified via accessible text.<input type="checkbox"/> Frames and iframes are appropriately titled.

Guideline 1.2 Time-based Media

Provide alternatives for time-based media

NOTE: If the audio or video is designated as an alternative to web content (e.g., an audio or sign language version of a web page, for example), then the web content itself serves as the alternative.

Success Criteria	Recommendations
1.2.1 Prerecorded Audio-only and Video-only (Level A)	<ul style="list-style-type: none"><input type="checkbox"/> A descriptive text transcript that includes relevant auditory content is provided for non-live audio-only (audio podcasts, MP3 files, etc.).<input type="checkbox"/> A descriptive text transcript or audio description is provided for non-live video-only (e.g., video that has no audio track), unless the video is decorative.
1.2.2 Captions (Prerecorded)	<ul style="list-style-type: none"><input type="checkbox"/> Synchronized captions are provided for non-live video (YouTube videos, etc.).

Section 508

- Applies to federal government
- Agencies may require compliance for recipients of federal funds
- To meet 508, meet WCAG 2.0 A/AA
- VPAT 2 – Voluntary Product Accessibility Template

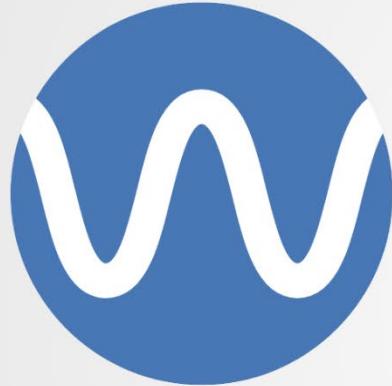
Americans with Disabilities Act

- Pre-dates the web
- 3 Important sections – Employment, State and local governments, Public and commercial facilities
- Does not define technical standards for web accessibility
- Lawsuits, complaints, and DOJ settlements require WCAG 2.0 or 2.1 A/AA

Legal Requirements – WCAG 2.1

- California GC § 11546.7
 - State agencies and entities
 - WCAG 2.0 “or a subsequent version”
 - July 2019
- European Union EN 301 549
 - New websites: September 2019
 - All websites: September 2020

Accessibility > Compliance



WAVE

web accessibility evaluation tool

wave.webaim.org

User experience

Auditory Disabilities



Captioning Methods

- Type, then sync
- Stenographer or
“Shadow Speaker”
- AI voice recognition



Audio Description

Typically a narrator that describes visual content.
Sometimes called “Descriptive Video Service” (DVS).

Avoid expensive audio descriptions by ensuring
important visual content is presented audibly.



Descriptive Transcript

- Called “Alternative for Time-based Media” in WCAG
- Text that presents all important audio and visual information
- Not synchronized
- Does not have to be text-only

Visual Disabilities

Assistive Technology

“Assistive technology promotes greater independence by enabling people to perform tasks that they were formerly unable to accomplish, or had great difficulty accomplishing.”

[Wikipedia](#)

Do you use a form of
assistive technology?

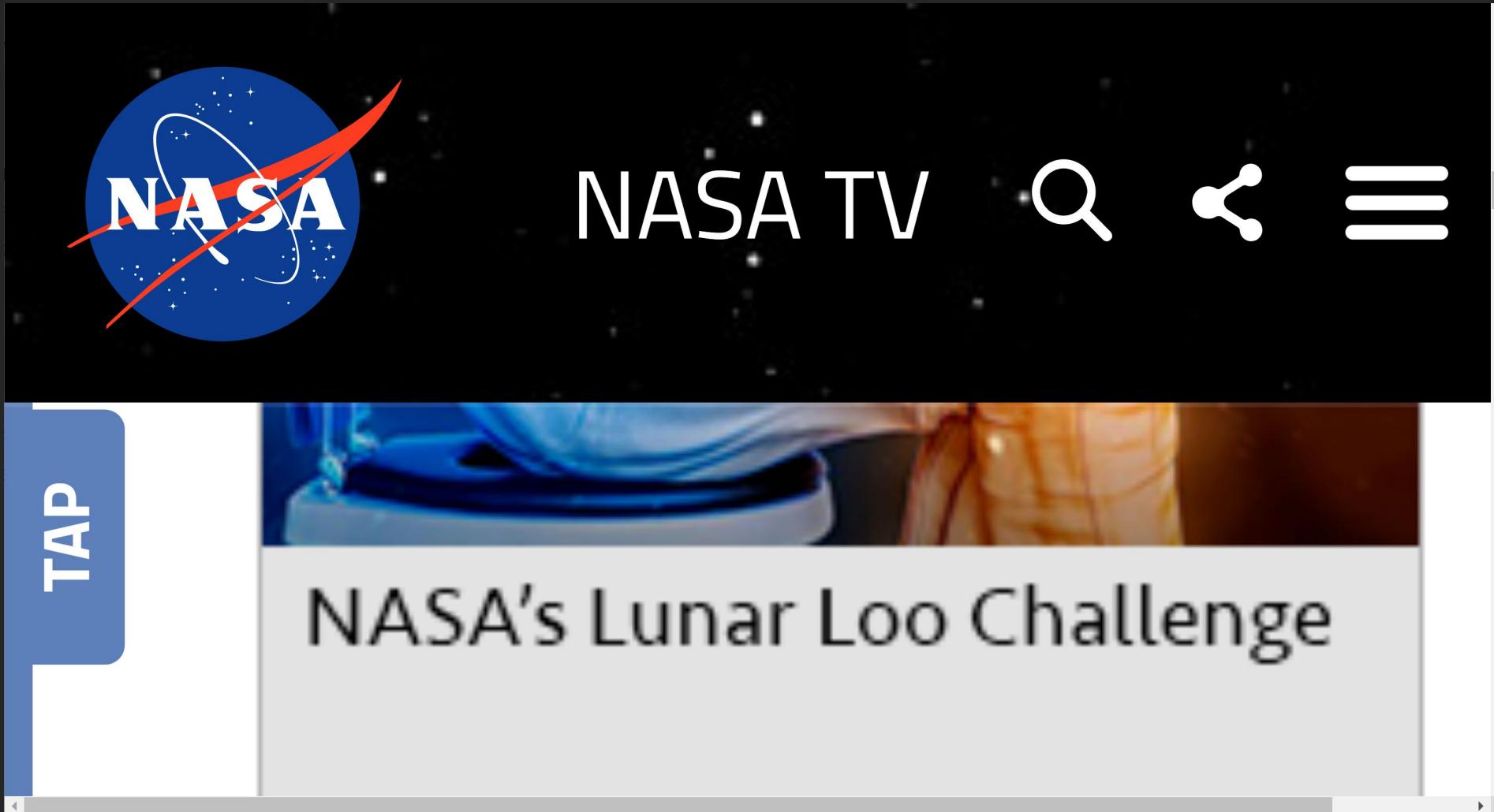


Low Vision

Browser Zoom

- Browser controls
 - Larger: Ctrl/command & +
 - Smaller: Ctrl/command & -
 - Reset : Ctrl/command & 0
- Zoomed content will trigger responsive breakpoints
 - Responsive design supports users with low vision who zoom page content

Browser Zoom on [nasa.gov](https://www.nasa.gov)



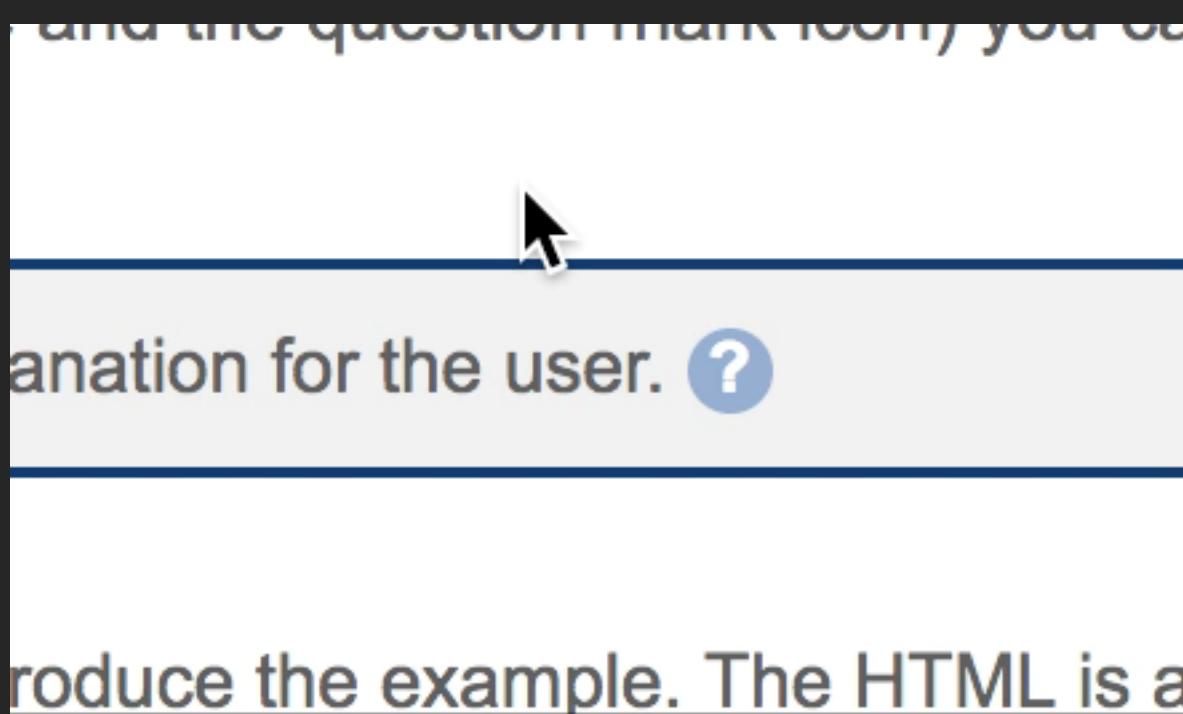
WCAG Zoom Requirements

- 1.4.4 Resize text (Level AA)
 - “Except for captions and images of text, text can be resized without assistive technology up to 200% without loss of content or functionality.”
 - Zoom to 200% is supported.
- 1.4.10 Reflow (Level AA - WCAG 2.1)
 - “Content can be presented without loss of information or functionality...” and without horizontal scrolling (except when necessary) at 400% zoom with viewport width at 1280 pixels.
- Consider text sizing
 - Perhaps to 130%-150%
 - Test in Firefox

Images of Text



Use click to trigger hover interactions



WCAG 2.1 – Content on Hover or Focus (Level AA)

Content that appears on hover and focus must be:

- “Dismissible...without moving pointer hover or keyboard focus”: Esc key dismisses
- “Hoverable”: Doesn’t disappear when moving the pointer to the new content
- “Persistent”: Visible until you move mouse away or dismiss it

Present messaging close to form controls

The image shows a user interface for a form with three fields:

- first name**: An input field containing a red exclamation mark icon. Below it is a red message box containing the text "Please enter your first name."
- where do you live?**: A dropdown menu labeled "choose" containing a red exclamation mark icon. Below it is a red message box containing the text "Please choose your country."
- birthday**: Three dropdown menus for month (jan), day (1), and year (19), followed by an input field for the year, which contains a red exclamation mark icon. Below them is a red message box containing the text "Your year of birth does not appear valid."

Evaluate page zoom

Ctrl/command & +, -, 0

Provide Sufficient Contrast

[WebAIM contrast article](#)

Inverted and High Contrast

amazon Try Prime

Your Amazon.com Today's Deals Gift Cards Sell Help

Shop by Department Search Kindle Store Go

Buy a Kindle Kindle eBooks Advanced Search Daily Deals Free Reading Apps Kindle Singles News

The World's Best-Selling E-readers

Compare Kindles

Kindle \$69 Kindle Paperwhite \$119 Kindle Paperwhite 3G \$189

kindle fire HD

Our most affordable Kindle Fire, now in HD



Kindle Fire 7" Tablet, Now in HD
7" HD Display, Wi-Fi, 8GB or 16GB
 (9,181 customer reviews)

\$139.00 FREE Shipping Details

Storage Size: 8 GB

8 GB 16 GB

Offer Type: With Special Offers

With Special Without Special

Special offers and on the Kindle Fire

WCAG Contrast Formula

$$(L1 + 0.05) / (L2 + 0.05)$$

where

$$L = 0.2126 * R + 0.7152 * G + 0.0722 * B$$

where

R, G, and B =

$$R_{sRGB} \leq 0.03928 \text{ then } R = R_{sRGB}/12.92$$

$$\text{else } R = ((R_{sRGB} + 0.055)/1.055)^{2.4}$$

where

$$R_{sRGB} \leq R_{8bit}/255$$

21 · 1

21 · 1

Level AA Contrast Examples

4.5:1

Gray (#767676) on White

Purple (#CC21CC) on White

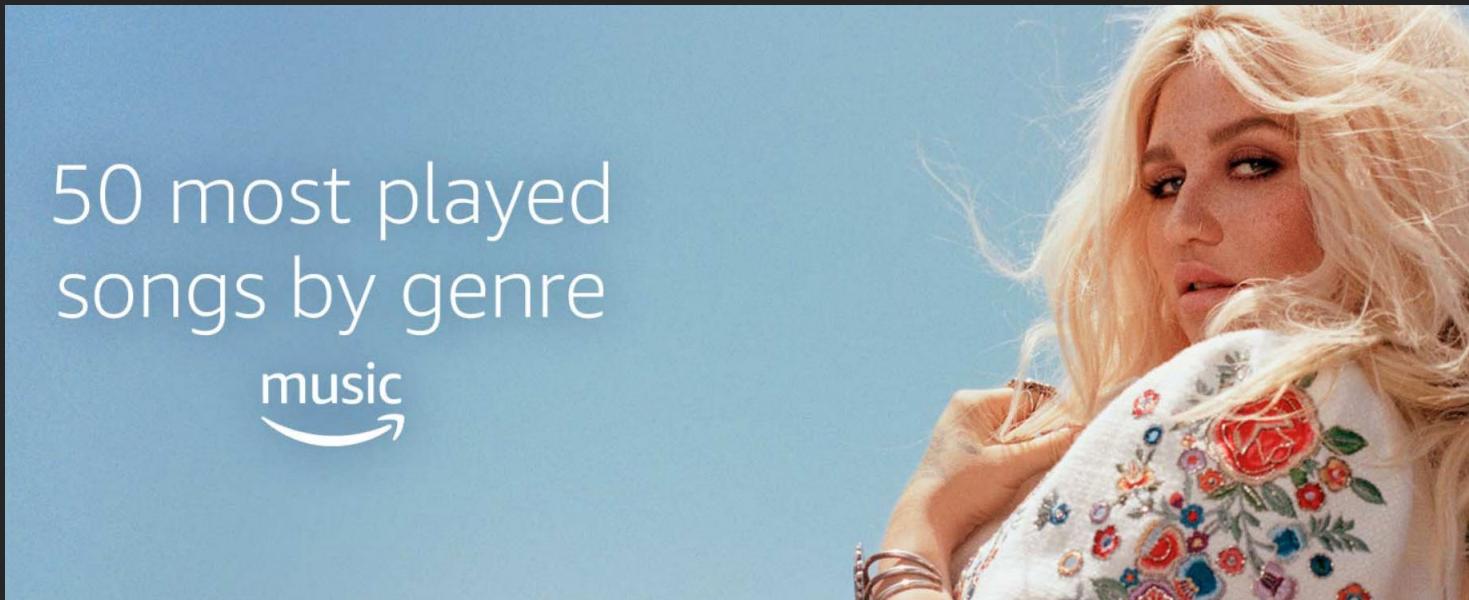
Red (#E30000) on Yellow (#FFFF00)

3:1 – “Large” text

$\geq 18\text{pt}$ (24px)

BOLD: $\geq 14\text{pt}$ (18.67px)

“Images of text”



WebAIM Contrast Checker

webaim.org/resources/contrastchecker/

I
AM
TEXT

I
AM
TEXT

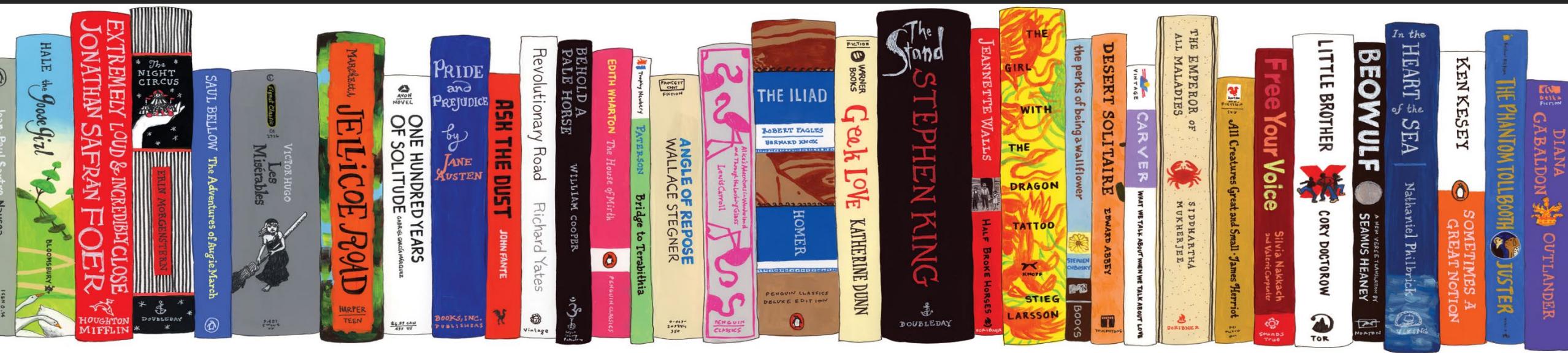


I
AM
TEXT

I
AM
TEXT

Exceptions

“Pure Decoration”

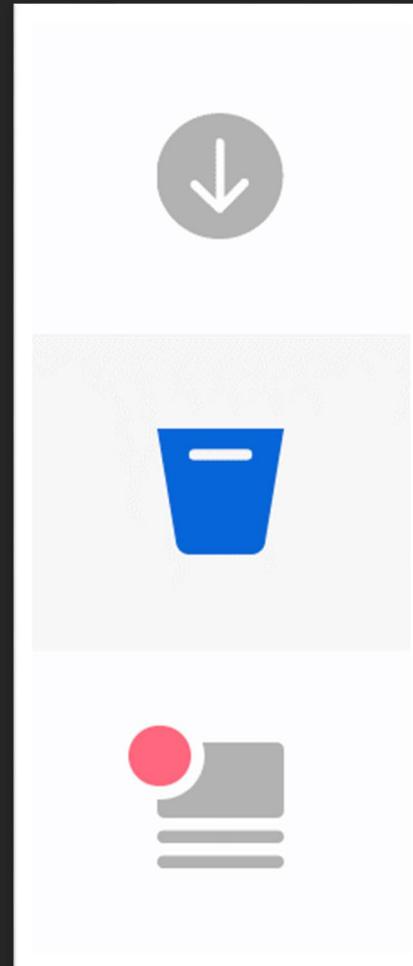


"Inactive"

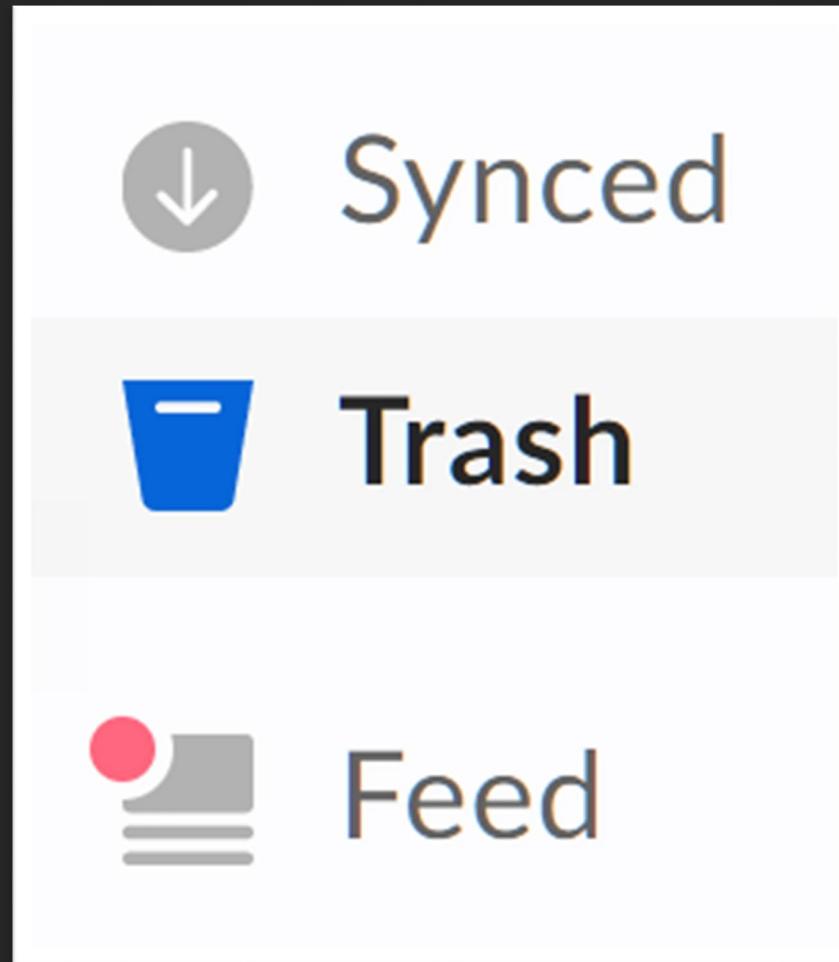
“Logotypes”



WCAG 2.1 – 3:1 Contrast of “Graphical Objects”



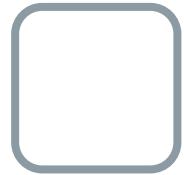
“...required to understand”



“Against adjacent color(s)”



3:1 Contrast of UI “Boundaries”



I have read and agree to the [Terms of Use](#)

Color Blindness



WCAG 1.4.1 – Use of Color

The green mushrooms listed here are okay to eat. The red mushrooms are poisonous.

- Amanita
- Chanterelle
- Porcini
- Shiitake
- Tylopilus

WCAG 1.4.1 – Use of Color

The green mushrooms listed here are okay to eat. The red mushrooms are poisonous.

- Amanita
- Chanterelle
- Porcini
- Shiitake
- Tylopilus

1. Ensure sufficient contrast
2. Do not rely on color alone

Requirements for non-underlined links

- A 3:1 contrast ratio between link text and non-link text
- Link must present a non-color cue (typically underline) on mouse hover AND keyboard focus
 - Not possible in most electronic documents (e.g., Word and PowerPoint)

You must agree to the **Terms of Use**

You must agree to the Terms of Use

You must agree to the **Terms of Use**

webaim.org/resources/linkcontrastchecker

You must agree to the Terms of Use

You must agree to the Terms of Use

and link becomes underlined on hover/focus.

You must agree to the **Terms of Use**

You must agree to the **Terms of Use**

You must agree to the Terms of Use

You must agree to the [Terms of Use](#)

Screen Reader Users

- Not all screen reader users are completely blind.
- Most have low vision.
 - 45% of respondents to our low-vision survey use a screen reader.
- Some users have a reading or cognitive disability.
- Focus on structure and semantics.

Structure & Semantics

The screenshot shows the homepage of the WebAIM website. At the top, the logo "WebAIM" is displayed with the tagline "web accessibility in mind" and a gear icon. A navigation bar with links for "services", "articles", "resources", and "community" is visible. Below the header, there is a search bar and three buttons: "Introduction to Web Accessibility" and "WebAIM Training". To the right, a woman is shown working on a laptop. The main headline reads "We have web accessibility in mind" with the subtitle "Expanding the potential of the web for people with disabilities by empowering individuals and organizations to create accessible content." Below this, there are six service cards: "Accessibility Training", "Accessible Site Certification", "Community", "Technical Assistance", "Evaluation and Reporting", and "Twitter". Each card has a small icon and a brief description.

We have web accessibility in mind

Expanding the potential of the web for people with disabilities by empowering individuals and organizations to create accessible content.

Accessibility Training

Whether here in Utah or on-site at your organization, WebAIM can provide web and document training to fit your needs.

Accessible Site Certification

As a respected third party accessibility expert, WebAIM can evaluate and certify your site to established web accessibility guidelines.

Community

[WebAIM Blog](#)

[Newsletter](#)

[E-mail Discussion List](#)

[Twitter](#)

Technical Assistance

Need assistance implementing accessibility? WebAIM's expert staff can provide the assistance you need.

Evaluation and Reporting

We can provide reports to help you know how accessible your site is and how to make it better.



Main Navigation

- [Services](#)
- [Articles](#)
- [Resources](#)
- [Community](#)

Search:

[Introduction to Web Accessibility](#)

[WebAIM Training](#)

Navigation

Header

We have web accessibility in mind

Expanding the potential of the web for people with disabilities by empowering individuals and organizations to create accessible content.

[Accessibility Training](#)

Whether here in Utah or on-site at your organization, WebAIM can provide web and document training to fit your needs.

[Accessible Site Certification](#)

As a respected third party accessibility expert, WebAIM can evaluate and certify your site to established web accessibility guidelines.

[Technical Assistance](#)

Need assistance implementing accessibility? WebAIM's expert staff can provide the assistance you need.

[Evaluation and Reporting](#)

We can provide reports to help you know how accessible your site is and how to make it better.

Community

- [WebAIM Blog](#)
- [Newsletter](#)
- [E-mail Discussion List](#)
- [Twitter](#)

Main

Screen readers for testing

Desktop

- Windows
 - NVDA (free)
 - JAWS (\$1000+)
 - JAWS + IE is still common
- Mac
 - VoiceOver

Mobile

- iOS
 - VoiceOver
 - 69% of mobile
- Android
 - Talkback

WebAIM tutorials for [JAWS](#), [NVDA](#), and [VoiceOver](#)

Screen Reader Demo

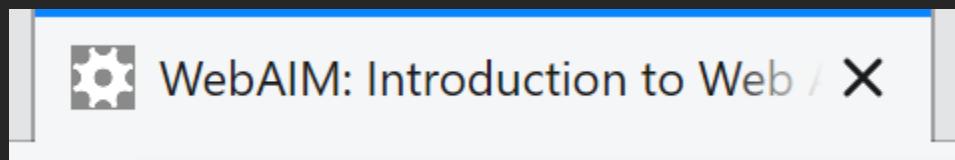
Screen reader testing

- Don't try to be an “expert”
 - 10 shortcuts to get started, 20 is about all you need
 - Screen readers read things differently and can misbehave
- Verbosity settings impact the amount and types of content that are read
- Focus on document structure, navigation, forms, and dynamic content

Headings

- The most common method for finding information on a page
 - 69% of WebAIM survey respondents
- Headings should describe a section of content
 - Should never be empty
- Usually one h1 per page
- Do not skip heading levels (e.g., h2 to h4)
 - Can skip backward (e.g., h4 to h2)

Page Title



```
<title>WebAIM: Introduction to  
Web Accessibility</title>
```

- Often the first thing read
- Should usually match, or be similar to, the first-level heading
- Must be descriptive
- Should be succinct

HTML Regions

- <header>
- <nav>
- <main>
- <aside>
- <footer>

Screen reader users can navigate by region

ARIA

Accessible Rich Internet Applications

ARIA expands the vocabulary of HTML

Rule #1 of ARIA Use

“If you can use a native HTML element or attribute...then do so.”

[Notes on Using ARIA in HTML](#)

Rule #1, simplified

If you can use HTML, then do so.

Rule #1, paraphrased

Don't use ARIA

ARIA Landmarks

- <header> - role="banner"
- <nav> - role="navigation"
- <main> - role="main"
- <aside> - role="complementary"
- <footer> - role="contentinfo"
- ??? – role="search"

Regions/Landmarks

- We recommend HTML regions (Rule #1)
 - <header role="banner"> is unnecessary
- “Region” / “Landmark” are generally synonymous
- All content in the page should be within a region
- More doesn’t always mean better

Evaluate title, headings, and regions/landmarks

WAVE “Structure” tab

Lists

- Numbers/`` for ordered or hierarchical lists
- Bullets/`` for unordered or parallel lists
- `<dl>` for definition/description lists
 - Name/value pairs in HTML5 (e.g., FAQ)

EASY SHORTBREAD

Ingredients

- 1 cup butter, softened
- 1/2 cup sugar
- 2 1/2 cups flour

Directions

1. Preheat the oven to 300 F.
2. Cream butter and sugar.
3. Gradually stir flour into the creamed mixture until blended.
4. Pat the dough in the bottom of an ungreased 9x13 baking pan.
5. Bake at 300 F for 30 to 40 minutes, until just lightly browned.
6. Remove from oven and pierce all over with a fork.



Baking Terms

Softened butter

Butter left at room temperature for approximately 20 minutes.

Cream butter and sugar

Mix butter and sugar together until it is light and fluffy.



Forms

Use the Correct Input Type

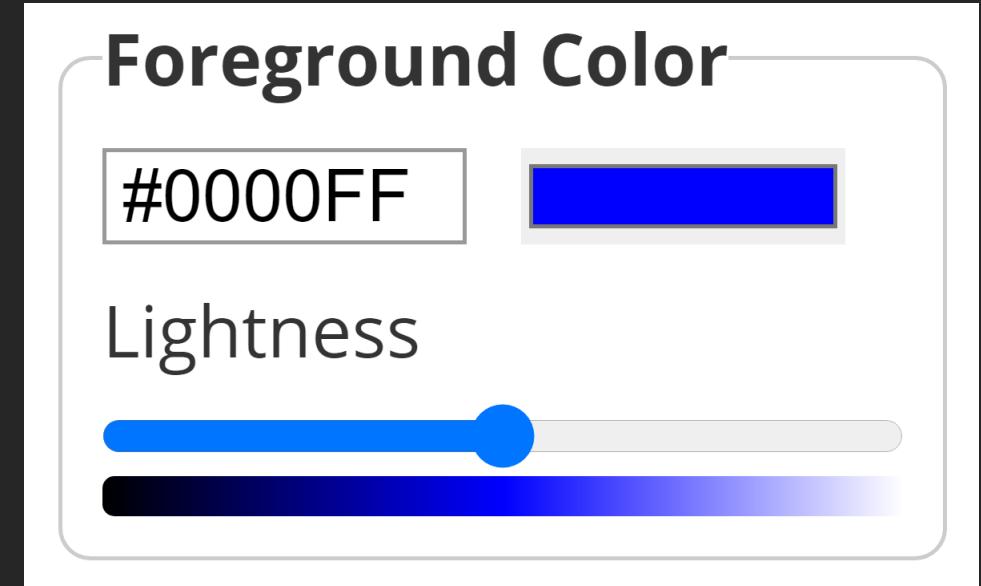
Select from the following to make group reservations.

School Group
 Other Group

Tour Operator

HTML5 Input Types

- <input type="date|email|range|color"> and many others.
- The browser can (or, in some cases, hopefully will) provide a natively accessible control/interface.
- Mobile devices often present custom keyboards
 - e.g., @ key for type="email".



[e.g., WebAIM contrast checker](#)

Form Labels

First Name:

```
<label for="fname">First Name:</label>
<input type="text" id="fname"...>
```

Text boxes, text areas, select menus, checkboxes,
radio buttons, and HTML5 input types.

Implicit Form Labels

```
<label>First Name:  
<input type="text"></label>
```

<fieldset> and <legend>

Choose a shipping method:

- Overnight
- Two day
- Ground

```
<fieldset>
  <legend>Choose a shipping method:</legend>
  <input id="overnight" type="radio"...
  <label for="overnight">Overnight</label>
  ...
</fieldset>
```

Evaluate forms and labels

Tables

Data tables

Class Schedule

Class Name	Course Number	Location
Advanced Website Development	BIS 5650	B105
Database Management	BIS 3330	B220

Data tables

```
<table>  
  <caption>Class Schedule</caption>  
  <tr>  
    <th scope="col">Class Name</th>  
    <th scope="col">Course Number</th>  
    <th scope="col">Location</th>  
  </tr>  
  <tr>  
    <th scope="row">Advanced Website Development</th>  
    <td>BIS 5650</td>  
    <td>B105</td>  
  </tr>  
  . . .
```

Structure for screen readers

Class Schedule

Class Name	Course Number	Location
Website Development	BIS 5650	B105
Database Management	BIS 3330	B220

Table example

Fall Semester		
Class Name	Course Number	Location
Advanced Website Development	BIS 5650	B105
Database Management	BIS 3330	B220

Fall Semester

Class Name	Course Number	Location
Advanced Website Development	BIS 5650	B105
Database Management	BIS 3330	B220

Fall Semester		
Class Name	Course Number	Location
Advanced Website Development	BIS 5650	B105
Database Management	BIS 3330	B220
Winter Semester		
Class Name	Course Number	Location
Advanced Website Development	BIS 5650	B105
Database Management	BIS 3330	B220

Fall Semester		
Class Name	Course Number	Location
Advanced Website Development	BIS 5650	B105
Database Management	BIS 3330	B220
Winter Semester		
Class Name	Course Number	Location
Advanced Website Development	BIS 5650	B105
Database Management	BIS 3330	B220

Fall Semester

Class Name	Course Number	Location
Advanced Website Development	BIS 5650	B105
Database Management	BIS 3330	B220

Winter Semester

Class Name	Course Number	Location
Advanced Website Development	BIS 5650	B105
Database Management	BIS 3330	B220

Layout Tables



Heat wave linked to temperatures



After three years of effort city scientists now agree that the primary cause of the 2003 heatwave was hot air from our
MORE ►►



Man Gets Nine Months in Violin Case



Mayor: These kinds of crimes need more creative, effective punishments. For example, we could require compulsory
MORE ►►



Lack of brains hinders research



Brain donations: huge drop off in brain donations due to the great 'success' of 'Slow Traffic, Safe Streets' policy
MORE ►►

Layout Tables

 Heat wave linked to temperatures	 Man Gets Nine Months in Violin Case	 Lack of brains hinders research
		
After three years of effort city scientists now agree that the primary cause of the 2003 heatwave was hot air from our MORE ►►	Mayor: These kinds of crimes need more creative, effective punishments. For example, we could require compulsory MORE ►►	Brain donations: huge drop off in brain donations due to the great 'success' of 'Slow Traffic, Safe Streets' policy MORE ►►

Layout Tables



Heat wave linked to temperatures



After three years of effort city scientists now agree that the primary cause of the 2003 heatwave was hot air from our

[MORE ►►](#)



Man Gets Nine Months in Violin Case



Mayor: These kinds of crimes need more creative, effective punishments. For example, we could require compulsory

[MORE ►►](#)



Lack of brains hinders research



Brain donations: huge drop off in brain donations due to the great 'success' of 'Slow Traffic, Safe Streets' policy

[MORE ►►](#)

Use <table role="presentation">



Heat wave linked to temperatures



After three years of effort city scientists now agree that the primary cause of the 2003 heatwave was hot air from our
MORE ►►



Man Gets Nine Months in Violin Case



Mayor: These kinds of crimes need more creative, effective punishments. For example, we could require compulsory
MORE ►►



Lack of brains hinders research



Brain donations: huge drop off in brain donations due to the great 'success' of 'Slow Traffic, Safe Streets' policy
MORE ►►

Language of Page

```
<html lang="en">
```

Language of Page

“Most people today can hardly conceive of life without the internet. Some have argued that no other single invention has been more revolutionary since Gutenberg's printing press in the 1400s. Now, at the click of a mouse, the world can be “at your fingertips”—that is, if you can use a mouse... and see the screen... and hear the audio—in other words, if you don't have a disability of any kind.”



Language of Parts

```
<div lang="fr">
```

Alternative Text

Alternative Text

- Read by screen readers
- Alternative to images if they are disabled or not supported
- Used by search engines

What is **equivalent** alternative text?

CONTENT and **FUNCTION**

VERY RARELY Description

Alternative Text

Should...

- be equivalent.
- be succinct.
- NOT be redundant.
- NOT use the phrases "image of ..." or "graphic of ..." to describe the image.





alt="WebAIM - Web
Accessibility in Mind"

Can be presented two ways

- In the **alt attribute** of the img element.
``
- In the **context** or surroundings of the image itself.



```
<a href="minivans.html">  
    <br>  
    Minivans  
</a>
```



```
<a href="minivans.html">  
    <br>  
    Minivans  
</a>
```

Empty vs. Missing alt attribute

- Empty alt (`alt=""`) definitively states:
 - The image is decorative, or
 - Alternative text is nearby
- Missing alt is ambiguous
 - Screen readers will probably treat it the same as `alt=""`.
- Modern content tools (e.g., WordPress and Word) should have a way to identify decorative images.

Images that are the only thing
within a link must **ALWAYS**
have alternative text

... and image map hot-spots and image buttons too.

Image Link

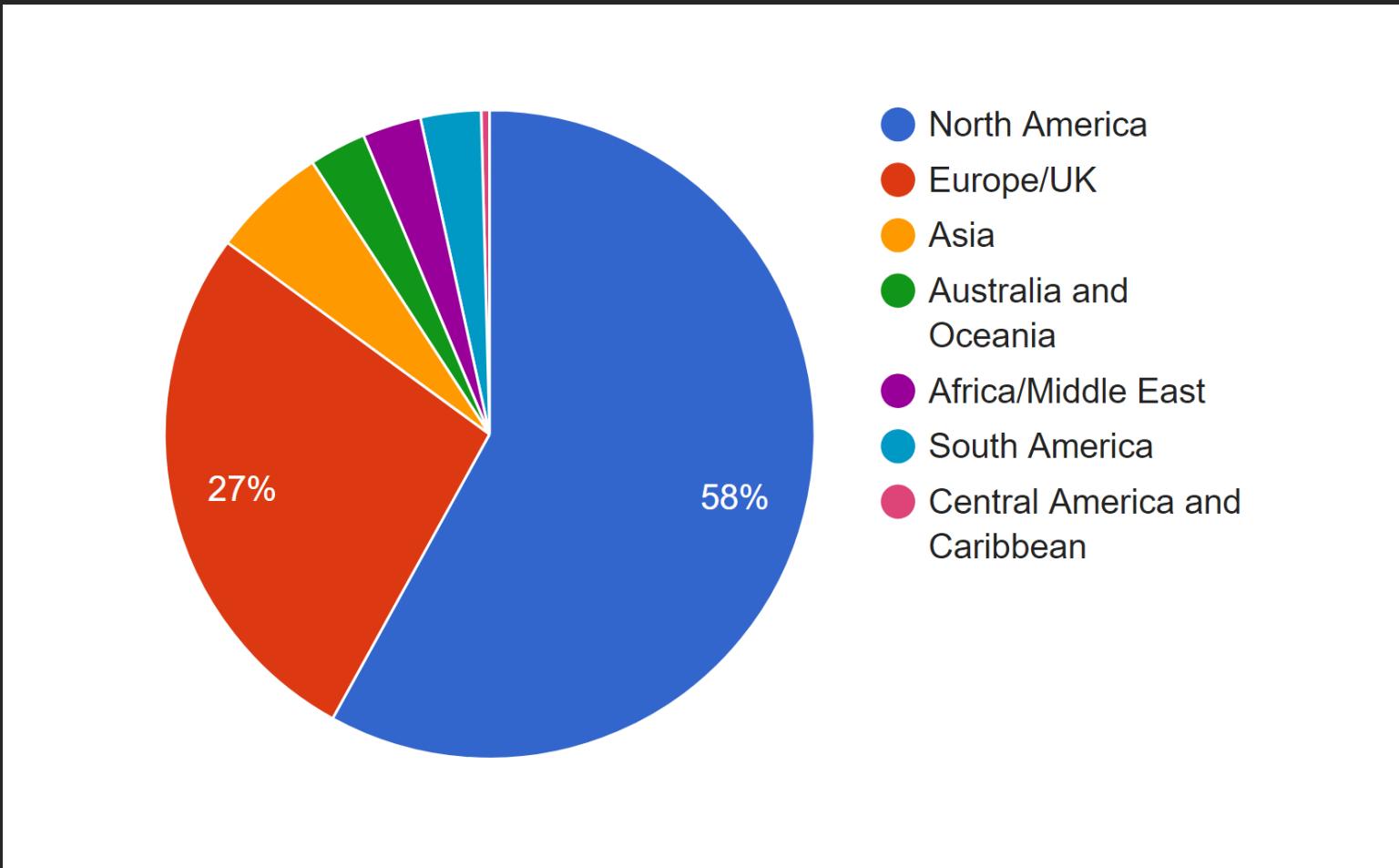


Text Link

Minivans

```
<a href="minivans.html">  
</a>  
<br>  
<a href="minivans.html">Minivans</a>
```

Complex Images



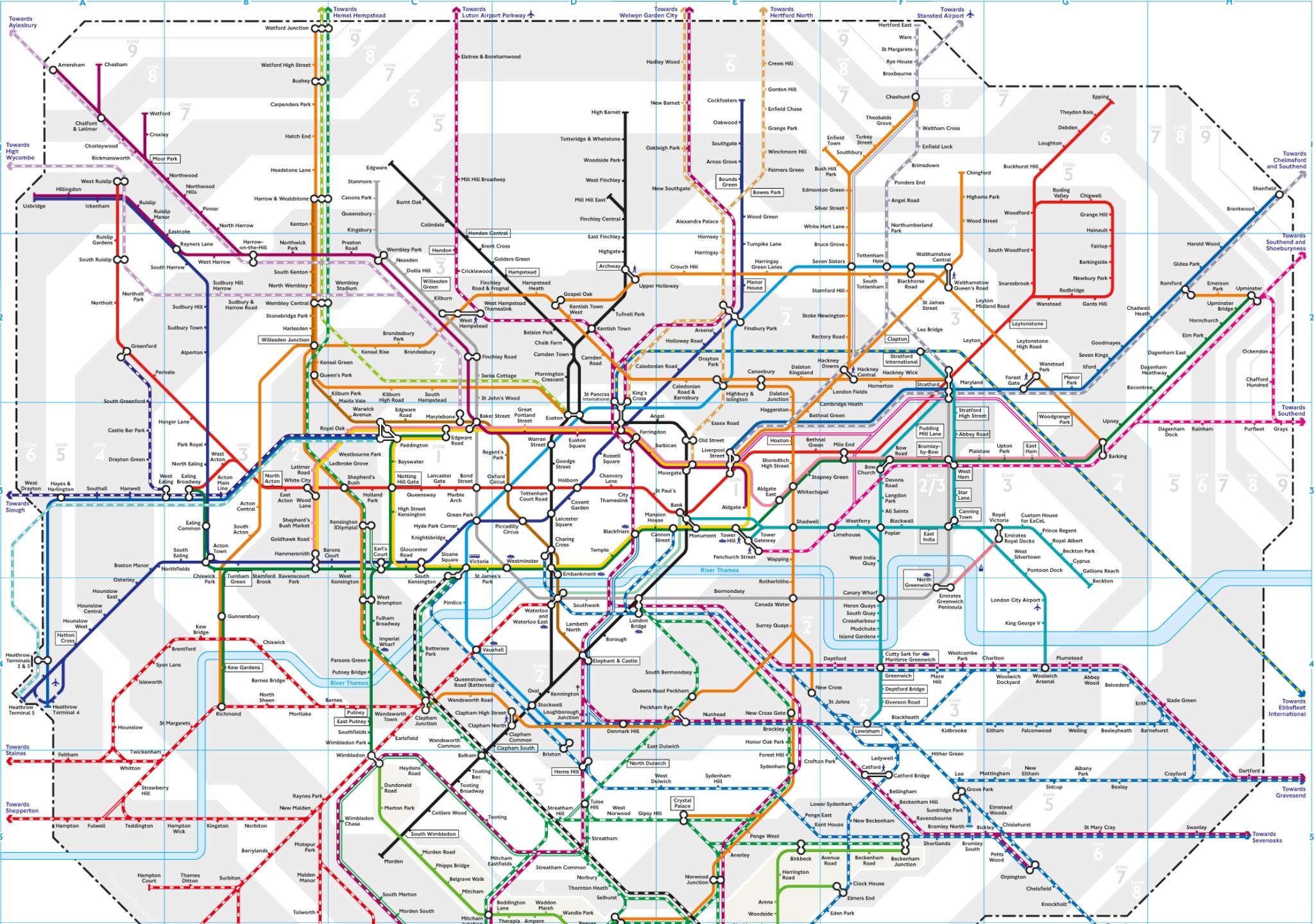
Complex Images

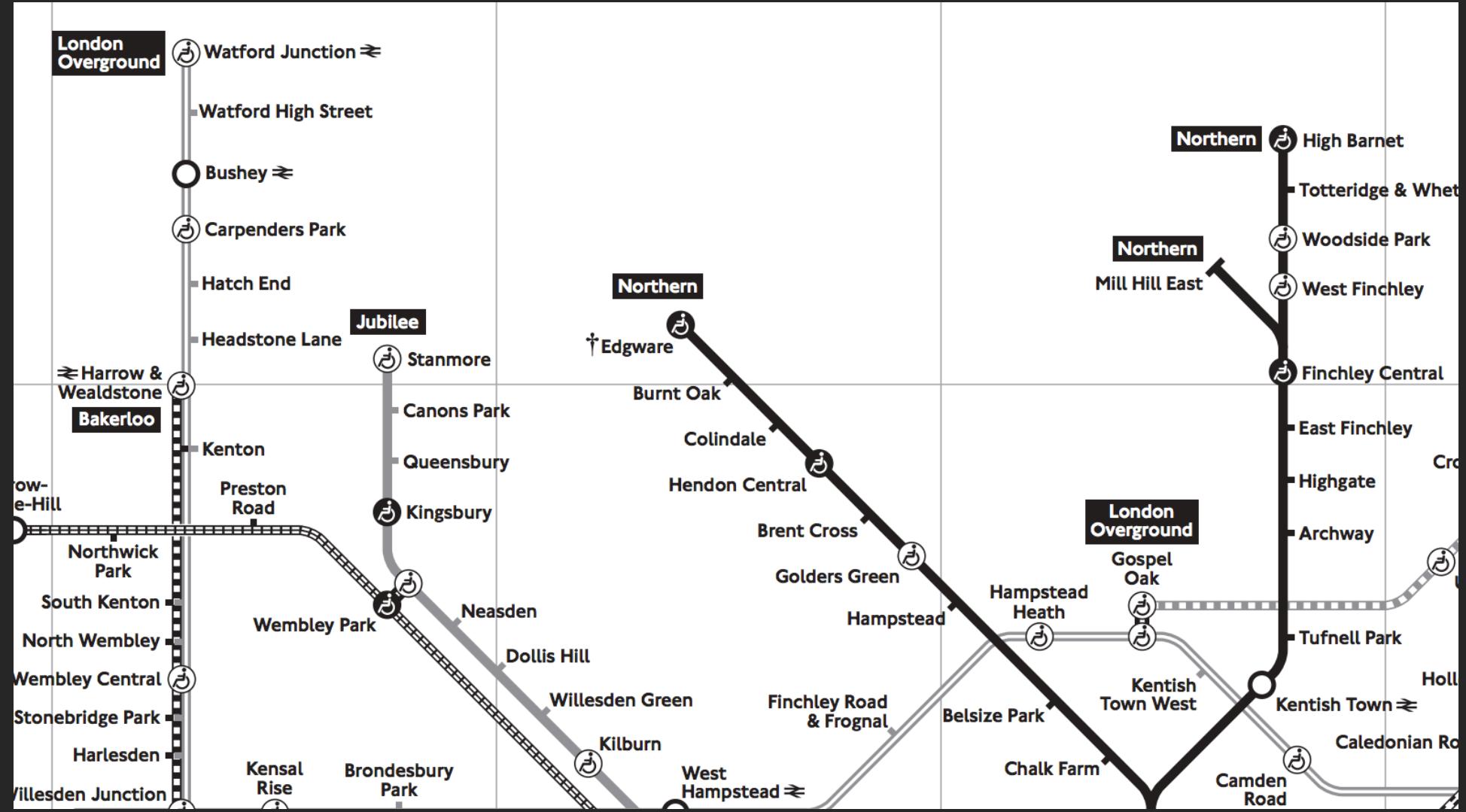
- Provide the alternative in context *OR*
- Provide a link to a page that contains the longer description
- Avoid longdesc—support is poor, and it has been dropped from HTML.
- The main image should still have some alt text!

London's Rail & Tube services

Key to lines and symbols

Bakerloo	
Central	
Circle	
District	
Hammersmith & City	
Jubilee	
Metropolitan	
Northern	
Piccadilly	
Victoria	
Waterloo & City	
DLR	
London Overground	
London Trams	
TfL Rail	
Emirates Air Line cable car	
Chiltern Railways	
c2c	
Gatwick Express	
Great Northern	
Great Western Railway	
Greater Anglia	
Heathrow Express	
Southern	
Southeastern	
Southeastern high speed	
South Western Railway	
Thameslink	
West Midlands Trains	
London Trams fare zone	
Station in both fare zones	
Interchange stations	
Street level transfer between stations	
Airport	
Riverboat services	
Victoria Coach Station	







Plan a journey

Heathrow Airport Terminal 4

Paddington

Leaving: now [change time](#) [Hide preferences](#) [Plan my journey](#)

My Journeys

Recents

Heathrow Airport Terminal 4 to
Piccadilly Circus, London, UK

Add favourites

Turn off / clear

Public transport

Cycling

Walking

Travel by

[select all](#) | [deselect all](#)

- | | | |
|--|-------------------|-------------------------------------|
| | Bus | <input checked="" type="checkbox"/> |
| | National Rail | <input checked="" type="checkbox"/> |
| | London Overground | <input checked="" type="checkbox"/> |
| | River Bus | <input checked="" type="checkbox"/> |
| | Emirates Air Line | <input checked="" type="checkbox"/> |

- | | | |
|--|----------|-------------------------------------|
| | Tube | <input checked="" type="checkbox"/> |
| | DLR | <input checked="" type="checkbox"/> |
| | TfL Rail | <input checked="" type="checkbox"/> |
| | Tram | <input checked="" type="checkbox"/> |
| | Coach | <input checked="" type="checkbox"/> |

Show me

- | | | |
|--|----------------------------|----------------------------------|
| | The fastest routes | <input checked="" type="radio"/> |
| | Routes with fewest changes | <input type="radio"/> |
| | Routes with least walking | <input type="radio"/> |

- | | |
|------------------------------|----------------------------------|
| No accessibility requirement | <input checked="" type="radio"/> |
| Use escalators, not stairs | <input type="radio"/> |
| Use stairs, not escalators | <input type="radio"/> |
| Step-free to platform only | <input type="radio"/> |
| Full step-free access | <input type="radio"/> |

Photosensitive Epilepsy



Caution with flashing/strobing content

- 3 times per second or greater
- Size, brightness, and red threshold
- Annoying rule

Cognitive/Learning Disabilities

- Largest disability group. Larger than all the others put together.
- Because users vary greatly, we will focus on general recommendations.

Make Content “Understandable”

- Be careful with movement and other distractions
- Focus on important content
- Be consistent
- Plain language
 - “readable to users with a lower secondary reading level” (WCAG AAA)
 - Write for your audience

There must be a mechanism for the user to pause, stop, or hide...

- Moving, blinking, or scrolling information that:
 1. starts automatically,
 2. lasts more than five seconds,
 3. *and* is presented in parallel with other content.
- Common failures:
 - Carousels
 - Auto-playing media
 - Animating ads

WCAG 2.1 - Text Spacing (Level AA)

“No loss of content or functionality occurs” when the user increases spacing between:

- Paragraphs: $2 \times$ font size
- Lines: $1.5 \times$ font size
- Words: $0.16 \times$ font size
- Letters: $0.12 \times$ font size

Text Spacing Applied

Earlier guidelines [\[edit\]](#)

The first web accessibility guideline was compiled by Gregg Vanderheiden and released in January 1995, just after the 1994 Second International Conference on the World-Wide Web (WWW II) in Chicago (where Tim Berners-Lee first mentioned disability access in a keynote speech after seeing a pre-conference workshop on accessibility led by Mike Paciello).^[2]

Over 38 different Web access guidelines followed from various authors and organizations over the next few years.^[3] These were brought together in the *Unified Web Site Accessibility Guidelines* compiled at the [University of Wisconsin-Madison](#).^[4] Version 8 of the *Unified Web Site Accessibility Guidelines*, published in 1998, served as the starting point for the W3C's WCAG 1.0.^[5]

WCAG 1.0 [\[edit\]](#)

The WCAG 1.0 was published and became a [W3C recommendation](#) on 5 May 1999. They have since been superseded by WCAG 2.0.

WCAG 1.0 consist of 14 guidelines—each of which describes a general principle of accessible design. Each guideline covers a basic theme of web accessibility and is associated with one or more checkpoints that describes how to apply that guideline to particular webpage features.

- Guideline 1: Provide equivalent alternatives to auditory and visual content
- Guideline 2: Don't rely on colour alone
- Guideline 3: Use markup and style sheets, and do so properly
- Guideline 4: Clarify natural language usage
- Guideline 5: Create tables that transform gracefully
- Guideline 6: Ensure that pages featuring new technologies transform gracefully
- Guideline 7: Ensure user control of time sensitive content changes
- Guideline 8: Ensure direct accessibility of embedded user interfaces
- Guideline 9: Design for device independence

Earlier guidelines [\[edit\]](#)

The first web accessibility guideline was compiled by Gregg Vanderheiden and released in January 1995, just after the 1994 Second International Conference on the World-Wide Web (WWW II) in Chicago (where Tim Berners-Lee first mentioned disability access in a keynote speech after seeing a pre-conference workshop on accessibility led by Mike Paciello).^[2]

Over 38 different Web access guidelines followed from various authors and organizations over the next few years.^[3] These were brought together in the *Unified Web Site Accessibility Guidelines* compiled at the [University of Wisconsin-Madison](#).^[4] Version 8 of the *Unified Web Site Accessibility Guidelines*, published in 1998, served as the starting point for the W3C's WCAG 1.0.^[5]

WCAG 1.0 [\[edit\]](#)

The WCAG 1.0 was published and became a [W3C recommendation](#) on 5 May 1999. They have since been superseded by WCAG 2.0.

WCAG 1.0 consist of 14 guidelines—each of which describes a general principle of accessible design. Each guideline covers a basic theme of web accessibility and is associated with one or more checkpoints that describes how to apply that guideline to particular webpage features.

- Guideline 1: Provide equivalent alternatives to auditory and visual content
- Guideline 2: Don't rely on colour alone
- Guideline 3: Use markup and style sheets, and do so properly
- Guideline 4: Clarify natural language usage

[Text spacing bookmarklet](#)

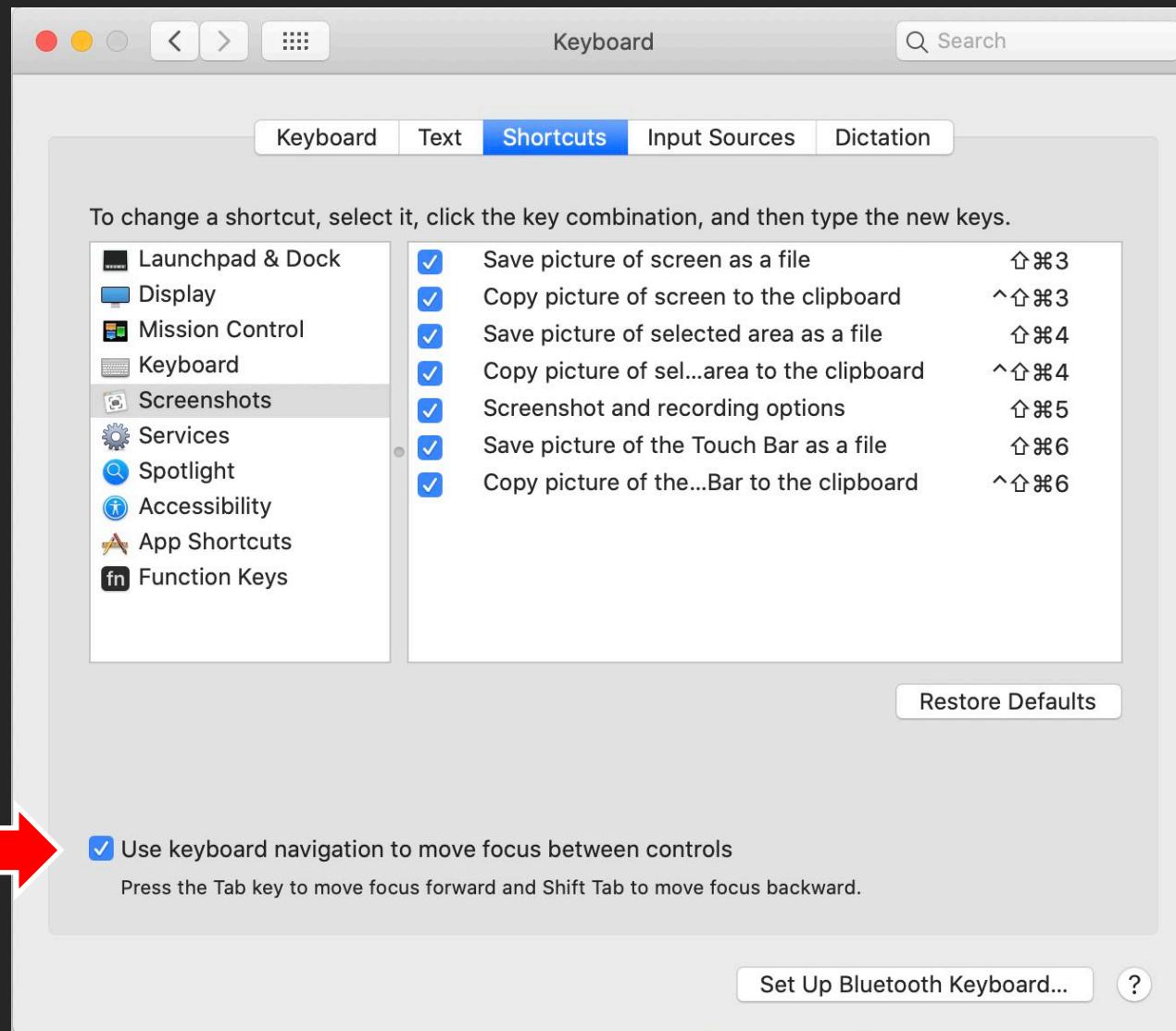
Motor disabilities

- Content and functionality should be accessible to mouse users and keyboard users
- Lack of fine motor control
- Repetition and fatigue
- Control over timing or moving elements

Standard keystrokes

- Navigate links, form controls, etc.: Tab, Shift + Tab
- Link: Enter
- Button: Enter **or** Spacebar
- Checkbox: Spacebar
- Radio buttons: ↑/↓ and ←/→
- Select menu: ↑/↓, Letters, Spacebar to expand
- Close: Esc
- Spacebar: Scrolls the page unless focused on a control
- Other widgets: use standard keyboard interaction patterns

Some tweaking required on Mac



kindle fire HD

Our most affordable Kindle Fire, now in HD



Product Tour



See 1 customer image

Share your own customer images



FreeTime



Apps & Games

K

learn more

17 of 17

^

v

x

★★★★★ (9,193 customer reviews)

\$139.00



Exclusive digital benefits with Amazon Prime

Storage Size: 8 GB

8 GB

16 GB

Offer Type: With Special Offers

With Special Offers

Without Special Offers

Special offers and sponsored screensavers display on the Kindle Fire lock screen. [Learn more](#)

In Stock.

Ships from and sold by **Amazon Digital Services**.
Important information about using outside the U.S.

Want it tomorrow, Feb. 19? Order within **7 hrs 50 mins** and choose **One-Day Shipping** at checkout. [Details](#)

We want you to know...

[Learn more](#) about design decisions and feature changes to help make an informed purchase

Our most affordable Kindle Fire—now with a stunning HD display, faster processor, and longer battery life

- Experience movies, TV, and games, and more on a stunning HD display (216 ppi / 1280x800)
- Fast 1.5GHz dual-core processor—apps launch quickly, games and videos play smoothly
- Create profiles and set time limits for children with Kindle FreeTime. Easy-to-use parental controls let everyone enjoy, worry-free
- Ultra-fast web browsing over built-in Wi-Fi, plus updated e-mail and calendar support for Gmail, Outlook, and more
- Instant access to over 100,000 apps and games in the Amazon Appstore, including a new paid app for free every day
- **Prime** In addition to Free Two-Day Shipping, Prime members can stream tens of thousands of Prime Instant Video titles at no additional cost, over half of which can be downloaded to the latest generation of Kindle Fire tablets.

[Learn more](#)

Meaningful link text

- [Click here](#)
- [Click here to log in](#)
- [Click here to log in](#)
- [Log in](#)



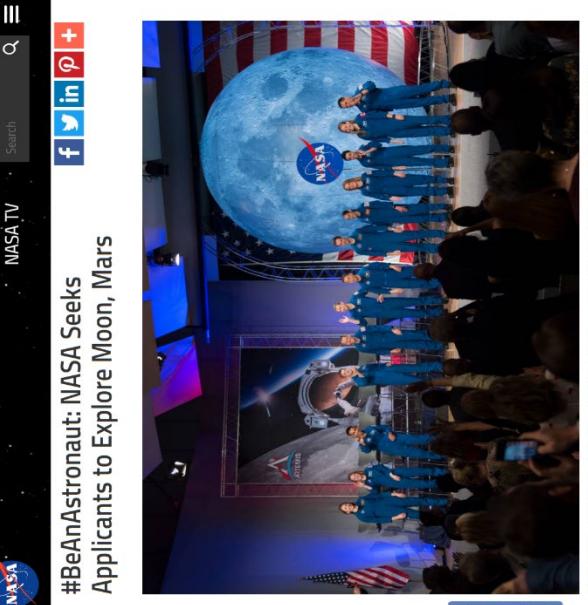
!= Touch Only

Evaluate keyboard accessibility

- Navigate the site using only the keyboard (Tab, Shift + Tab, Enter, Space, arrow keys, Esc). Is all functionality available?
- Is navigation order logical?
- Is a visible keyboard focus indicator/outline present?
- How about responsive layouts?

WCAG 2.1 - Mobile

1.3.4 (AA) – Don't lock orientation



NASA's new class of astronauts – the first to graduate since the agency announced its Artemis program – appear on stage during their graduation ceremony at the agency's Johnson Space Center in Houston on Jan. 10, 2020. The class includes 11 NASA candidates, as well as two Canadian Space Agency (CSA) candidates, selected in 2017. They will join the active astronaut corps, beginning careers in exploration that may take them to the International Space Station, or inspire to the Moon under the Artemis program, or someday Mars. Pictured from left are: Kylie Barron of NASA, Zena Cardman of NASA, Raja Chari of NASA, Bob Hines of NASA, Warren Hoburg of NASA, Jenny Kim of NASA, Jessica Kunk of CSA, Jasmin Moghbeli of NASA, Loral O'Hara of NASA, Jessica Watkins of NASA, Jennifer Snyders of CSA, and Frank Rubio of NASA.

Credits: NASA

Lee esta nota de prensa en español.

For the first time in more than four years, NASA began accepting applications Monday for future astronauts. Aspiring Moon to Mars explorers have until 11:59 p.m. EDT Tuesday, March 31, to apply.

The call for more astronauts comes at a time when the agency is preparing to send the first woman and next man to the Moon with the Artemis program. Exploring the Moon during this decade will help prepare humanity for its next giant leap – sending astronauts to Mars.

U.S. citizens may submit applications to #BeAnAstronaut at:

2.1.4 (A) – Disable or remap single key shortcuts

**Keyboard
shortcuts:**

[Learn more](#)



Keyboard shortcuts off



Keyboard shortcuts on

New to 2.1 – Guideline 2.5 Input Modalities

- 2.5.1 (A) – Don’t rely on multipoint or path-based gestures (e.g., swipe to navigate)
 - WCAG 2.2 proposes “dragging” functionality have single click/touch alternative
- 2.5.2 (A) – Pointer functionality can be canceled (e.g., no down-event to trigger)
- 2.5.3 (A) – “Label in Name”
- 2.5.4 (A) – Don’t rely on device motion (e.g., shake, tilt, or pan)
- 2.5.5 (AAA) – $\geq 44 \times 44$ pixel clickable targets
- 2.5.6 (AAA) – Don’t restrict input type (e.g., touch-only)

WCAG 2 “Label” and “Name”

WCAG requires an accessible “label” (2.4.6, 3.3.2) and “name” (1.1.1)

- The **Label** is visually presented
- The **Name** is presented to assistive technology
 - Also called “accessible name”
 - May be visually hidden

What is the WCAG “Label”?



What is the WCAG “Name”?



Next

```
<a href="page2.php">
  <img alt="Next"...
</a>
```

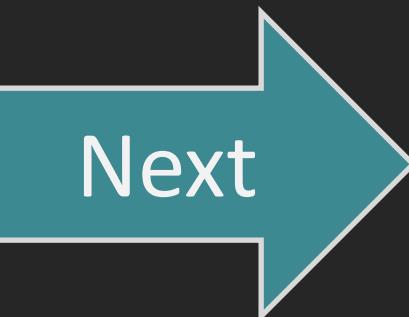
“Label” and “Name” are often the same

Code	Label	Name
<code>Log in</code>	<u>Log in</u>	“Log in”
<code><label>First Name:<input type="text"></label></code>	First Name: 	“First Name:”
<code><button>Activate</button></code>	Activate	“Activate”

2.5.3 – Label in Name (Level A)

“For user interface components with labels that include text or images of text, the name contains the text presented.”

Label not in Name – Fails 2.5.3



Next

```
<a href="page2.php">
    <img alt="Continue" ...>
</a>
```

Label in Name – Passes 2.5.3



```
<a href="page2.php">
    <img alt="Next Page" ...>
</a>
```

Better solution



```
<a href="page2.php">  
    <img alt="Next" ...>  
</a>
```

Two principles of accessible names

1. An element can only have one accessible name.
2. To be sure of the name, you must test with a screen reader or inspect the HTML.

Hiding Content

- Hide from everyone:
 - `display:none`
 - `visibility:hidden`
 - `HTML hidden attribute`
- Ensure hidden elements are not navigable
 - Hide them with `display:none` or ensure they become visible when they receive focus
- Avoid 0 pixels, same color as background, etc.

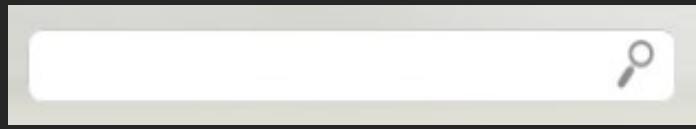
Visually Hiding Content

- Position off-screen left with CSS for screen readers

```
.sr-only {  
    position: absolute;  
    left: -10000px;  
    top: auto;  
    width: 1px;  
    height: 1px;  
    overflow: hidden;  
}
```

Can also use CSS clip and/or clip-path

Hidden <label>



```
<label class="sr-only" for="s">  
    Search Terms</label>  
<input type="text" id="s">
```

Do not remove the keyboard focus indicators from links

Avoid

`a {outline:0;}`

or

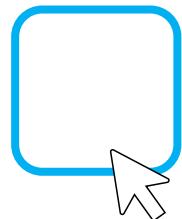
`a {outline:none;}`

WCAG AA Failure

3:1 contrast of keyboard focus indicators

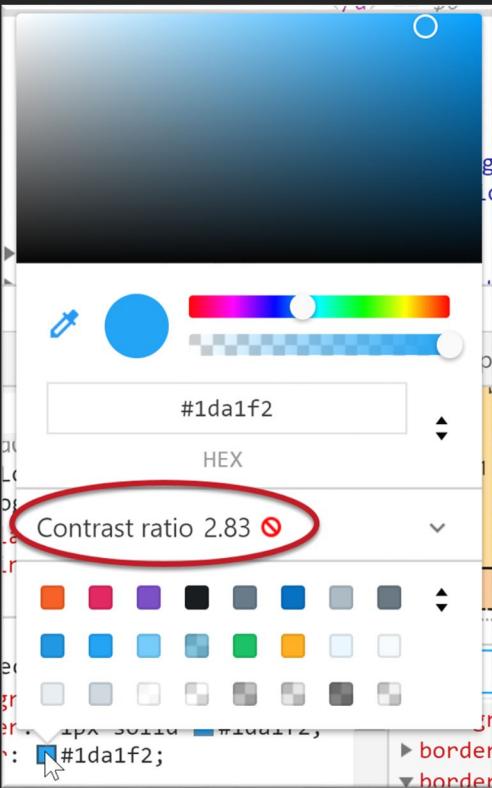


WCAG 2.1 – 3:1 contrast of UI “States”



I have read and agree to the [Terms of Use](#)

Chrome Developer Tools



A screenshot of the Chrome DevTools element info panel. It displays styling information for an element with the selector "a.js-nav.EdgeButton.EdgeButton--medium.EdgeButton-...". The panel includes fields for Color (white), Font ("14px 'Segoe UI', Ar...", color #1DA1F2), Background (#1DA1F2), Margin (0px 0px 16px), Padding (6px 16px), and Contrast (2.83 < 4.5). A preview of a blue button with the text "Sign Up" is shown at the bottom.

Evaluating contrast with Chrome DevTools

“Skip” links

```
<a href="#maincontent">Skip to  
main content</a>
```

```
...
```

```
<main id="maincontent">...
```

WCAG 2.4.1 (Level A): A mechanism is available to bypass blocks of content that are repeated on multiple Web pages.

Hidden “Skip” links

```
a#skip {  
    position: absolute;  
    left: -1000px;  
    top: auto;  
    width: 1px;  
    height: 1px;  
    overflow: hidden;  
}
```

```
a#skip:focus {  
    position: static;  
    width: auto;  
    height: auto;  
}
```

Don't use `display:none` on the link

Rule #1 of ARIA Use

If you can use HTML, then do so.

Rule #2

Do not change native semantics,
unless you really have to.

```
<div role="navigation">  
    <ul>...</ul>  
    </div>
```

instead of

```
<ul role="navigation">
```

ARIA does not change functionality; it only changes the roles/properties presented to screen reader users

Rule #3

All interactive ARIA controls must be usable
with the keyboard.

Design Patterns

WAI ARIA Authoring Practices

Rule #4

Do not use
role="presentation" or
aria-hidden="true" on visible,
focusable elements.

Rule #5

All interactive elements must have an
accessible name

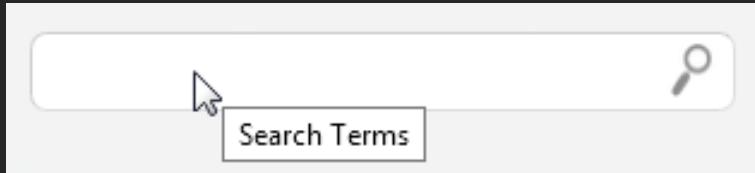
Accessible Name Computation

Form <label>

First Name:

```
<label for="fname">First Name:</label>
<input type="text" id="fname" ...>
```

title attribute (if no <label>)

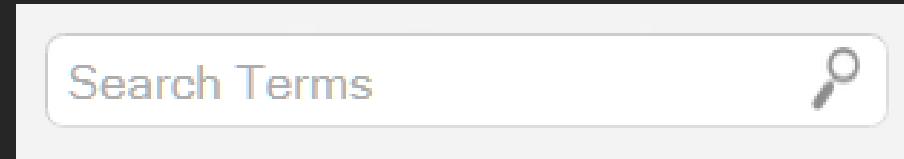


```
<input title="Search Terms"...>
```

title attribute

- Advisory information only
- Should be read for:
 - Form controls missing labels
 - Frames and iframes (iframe title is optional)
 - <abbr>/<acronym>...usually
 - Typically best to avoid <abbr> and <acronym>
- May or may not be read in other places (links, images, etc.)
- Not accessible to keyboard users, touch screen users, etc.

placeholder attribute (if no title or <label>)



```
<input placeholder="Search Terms"...>
```

Placeholder is read by a screen reader (accessible name),
but is not a suitable “label”

< Add debit or credit card

First name

Last name



Card number

Expires

MM/YY

CSC

3 digits



Billing address

Add

< Add debit or credit card

Smith

John



Card number



Expires

1/22



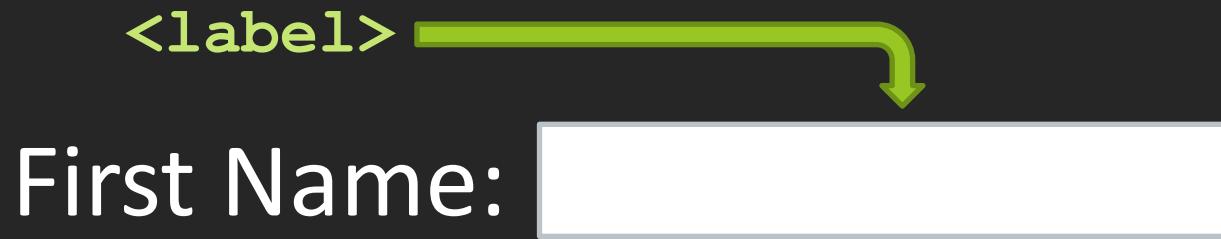
CSC

3 digits



Enter a valid expiration
date

Add



```
<label for="fname">First name:</label>
```

```
<input type="text" id="fname">
```

1:1 relationship between <label> and a form control.

First Name:



aria-labelledby

```
<span id="fnamelabel">First name:</span>
```

```
<input type="text"  
aria-labelledby="fnamelabel">
```



```
<label for="fname" id="fnamelabel">  
First name:</label>
```

```
<input type="text" id="fname"  
aria-labelledby="fnamelabel">
```

Unnecessary markup (Rule #1 of ARIA),
but label will only be read once.

One label for multiple controls

`id="namelabel"`



Name	Age	Weight

```
<input type="text" name="name1"  
aria-labelledby="namelabel">
```

Multiple labels for multiple controls

`id="officenum"`

`id="cyndi"`

Name	Office Number	Phone
Cyndi	<input aria-labelledby="cyndi officenum" name="officel" type="text"/>	<input name="officenum" type="text"/>
Jared	<input aria-labelledby="cyndi officenum" name="officel" type="text"/>	<input name="officenum" type="text"/>
Jonathan	<input aria-labelledby="cyndi officenum" name="officel" type="text"/>	<input name="officenum" type="text"/>

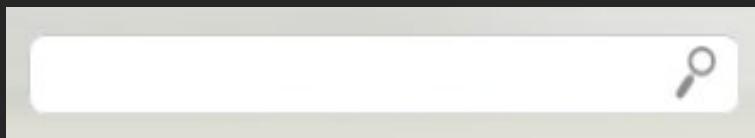
```
<input type="text" name="officel"  
aria-labelledby="cyndi officenum">
```

Form values as labels for other inputs

Name	Office Number	Phone
Cyndi		
Jared		
Jonathan		

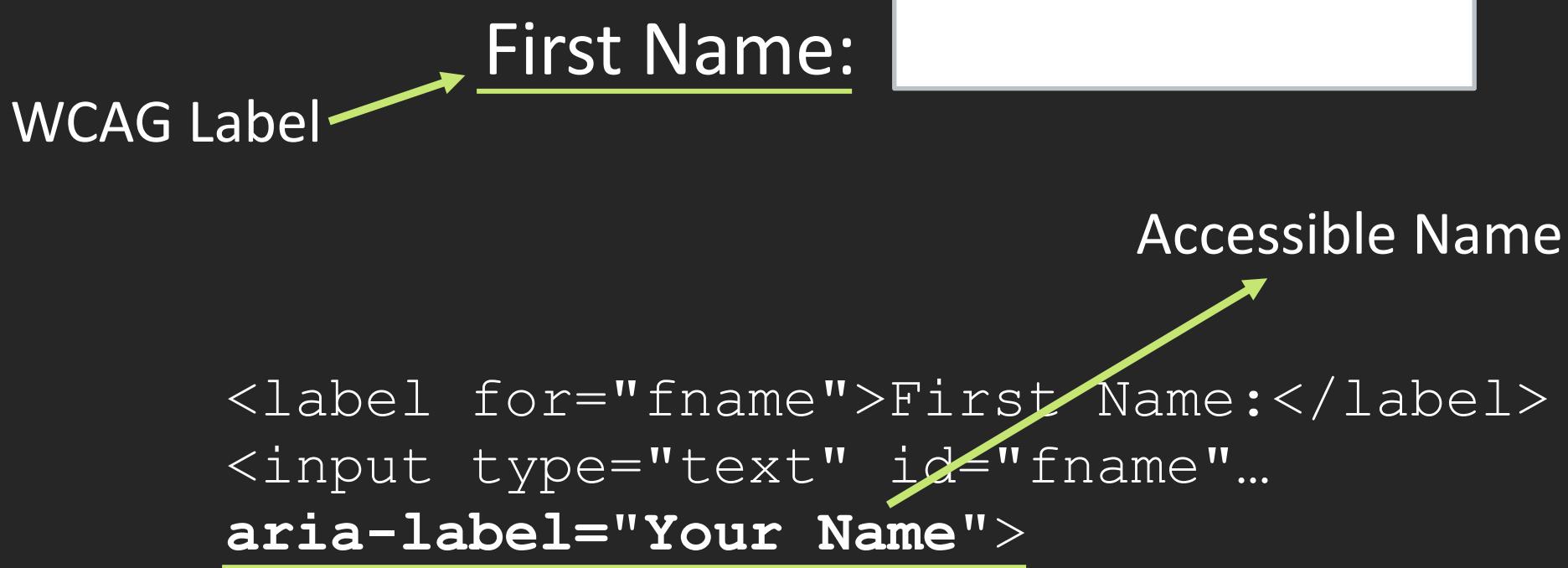
```
<input type="text" name="office1"  
aria-labelledby="name1 officenum">
```

aria-label

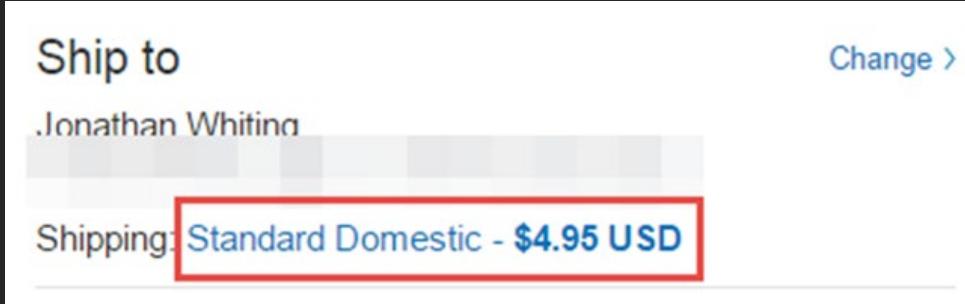


```
<input aria-label="Search Terms" ...>
```

ARIA *labels* override default accessible *names*



...and not just in forms



```
<a... aria-label="Change shipping  
method">Standard Domestic - $4.95 USD</a>
```

Also applies to alternative text, button text, etc.

 cannot have an accessible name

[Download the Employment Application](#)



```
<a>Download the Employment Application  
<span class="PDFlink" aria-label="PDF"></span>  
</a>
```

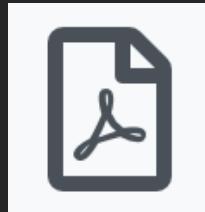
Neither can <div>, <p>, <i>, etc.

Three fixes

[Download the Employment Application](#) 

1. Download the Employment Application<**span class="sr-only"- 2. Download the Employment Application
- 3. **Download the Employment Application**

CSS generated content

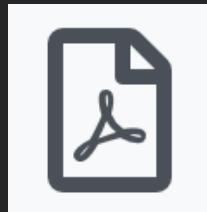


```
<i class="pdf"></i>
```

```
.pdf::after {  
    font-family: FontAwesome;  
    content: "\f1c1";  
}
```

CSS generated content will be read in modern screen readers that understand the generated content.

CSS generated content



```
<i class="pdf" role="img" aria-label="PDF"></i>
```

or

```
<button aria-label="PDF"><i class="pdf"></i></button>
```

Scalable Vector Graphics (SVG)

1.
2. <svg role="img" aria-labelledby="title">
 <title id="title">Sales increased 10%
 from 2010 to 2015</title>
 ...</svg>
3. <svg role="img" aria-label="Sales increased 10%
 from 2010 to 2015">
 ...</svg>

Use `role="presentation"` if the SVG is decorative (or `alt=""` for ``).

aria-describedby

Username:
Must be 8-15 characters

```
<label for="user">Username:</label>
<input type="text" id="user"
aria-describedby="usernameDetails"
```

Evaluate accessible name and description with
Chrome DevTools

ARIA and HTML labels and descriptions

- Are read as a stream of text. Users cannot easily navigate or explore the referenced content.
- Should be succinct.
- aria-label is not automatically translated (e.g., Google Translate)
- Semantics (lists, links, etc.) are removed from referenced content.

Username:

Must:

- *be at least 14 characters in length*
- *contain at least 4 vowels*
- *contain one medieval rune of alien origin*

Username:

Must meet these requirements

Hidden ARIA and HTML labels and descriptions

- <label> hidden with CSS display:none or hidden **will not** be read by screen readers.
- Elements hidden with display:none or hidden, but referenced by aria-labelledby or aria-describedby **WILL** be read.

```
<label for="user">Username:</label>
<input type="text" id="user"
aria-describedby="usernameerror">
<p id="usernameerror"
style="display:none"
```

aria-required

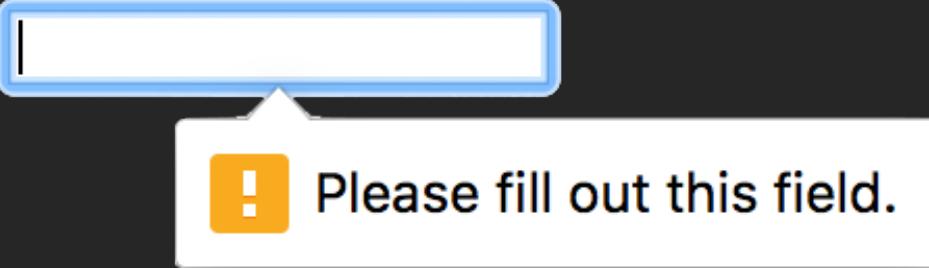
Username:*

```
<label for="username">Username:</label>*<br/><input type="text" id="username" aria-required="true">
```

A screen reader indicates the field is required...
and that's all.

HTML required

Username:*



Please fill out this field.

```
<label for="username">Username:</label>*  
<input type="text" id="username" required>
```

Semantics are the same, but validation messaging is provided

Pay with debit or credit card

We don't share your financial details with the merchant.

United States



Card number

Expires

CSC

MM



YY



3 digits



First name

Last name

Email



Mobile



Phone number

Billing address

Street address

Apt., ste., bldg., etc. (optional)

City

State



ZIP

aria-invalid

Password:



```
<label for="password">Password:</label>
<input type="password" id="password"
aria-invalid="true">
```

A screen reader indicates the field is invalid or broken...
and that's all.

Use ARIA attributes to control styling:

```
[aria-invalid=true] {
  border: 2px solid red;
  background-image: url("error.png");
}
```

Form Validation and Error Recovery

- Avoid
 - Collecting unnecessary information
 - Forcing users to provide formatting unless necessary
- Identify
 - Let the users know there is an error
 - Direct users to errors
 - Describe the error
- Suggest
 - How to fix it
- Prevent
 - Important changes (e.g., Financial) are reversible/checked/confirmed
- WCAG 4.1.3 requires “status messages” to be accessible.

Form Validation Types

- Alert, then focus
- Errors at the top
- Inline errors

Alert, then focus

First Name:

Last Name:

Email Address:

webaim.org says

First Name is required.

Errors at the top

There was a problem with the form.

- Please enter Your Name.
- Please enter Your E-mail Address.

Inline Errors

Username

spiderman

Username not available

Password

Must have more than 8 characters

Must have at least one number

Must have upper & lowercase letters

Use links/buttons appropriately

- Links open a new page or jump to another location within the existing page
- Buttons submit form data or perform an in-page function

Ensure Interactive Elements are Links or Form Controls

Tabindex of 1 or greater

- `tabindex="1+"` defines an explicit tab order
- Never use positive tabindex values!
- If the default tab order is not logical, fix your source code order.

Tabindex of 0 or -1

- `tabindex="0"` allows things besides links and form controls to receive keyboard focus.
- `tabindex="-1"` allows things besides links and form controls to receive programmatic focus (by scripting, links, etc.)

```
<div onclick="submitForm()">  
  Search</div>
```

```
<div onclick="submitForm( )"  
      tabindex="0">  
  Search</div>
```

Check for Enter (13) and Space (32) key presses

```
if(event.keyCode==13 || event.keyCode==32)
{
    doStuff();
}
```

```
function allyClick(event) {
    if(event.type === 'click') {
        return true;
    }
    else if(event.type === 'keypress') {
        var code = event.charCode || event.keyCode;
        if(code === 32) {
            event.preventDefault(); // don't scroll the page
        }
        if((code === 32) || (code === 13)) {
            return true;
        }
    }
    else{
        return false;
    }
}
$('#fake-button').on('click keypress', function(event) {
    if(allyClick(event) === true) {
        // do magic javascript stuff
    }
}) ;
```

Just Use A Button

```
<div onclick="submitForm( )"  
      tabindex="0">  
  Search</div>
```

```
<a href="#" onclick="submitForm()">  
    Search</a>
```

```
<a href="#" onclick="submitForm()"  
      role="button">  
    Search</a>
```

Just Use A Button

```
<button onclick="submitForm() ">  
  Search</button>
```

`tabindex="-1"`

- Allows non-focusable elements to receive programmatic focus (by scripting, links, etc.)
- Necessary for focusing dialog boxes, error messages, etc.
- Also used to manage keyboard focus in some widgets (e.g., a tree menu or tab panel)
- **WARNING:** This removes the element from the default tab order.

Tab Panels

The behavior of the Tab Panels on the AHA Workspace are the correct implementation for tabs

Tab Panel

SUPPLIER UAT SOURCING REPORT

Documents

SUPPLIER 362 FORM 4 SCORECARD 18 ACTION PLAN 1

All 362 Registered 154 Invited 101 Approved 96 Rejected 6 Identified 3 Disqualified 1 Waitlisted 1

Invite For Registration Additional Forms Request F

Legal Company Name	DUNS Number	Organization entity	Created By	Primary Contact Name	Primary Contact Email
CHRISTUS St.Vincent St. Michaels F		Corporate Services + (2) more	Greg Setser	Tiffany Taylor	ahageptest+tiffstreet@
AirCon Savers Inc.	0	Corporate Services + (2) more	Greg Setser	Steve Cooley	ahageptest+airconsaver
Technology Partners LLC	890678456	Business Technology + (2) more	Sally Smart	Gary Jones	ahageptest+garyjones@
Mr Clean Inc	0	Corporate Services + (2) more	Greg Setser	Daniel Cleaner	ahageptest+mrclean@g
Technology Partners Corp	879056768	Corporate Services + (2) more	Greg Setser	Sally Jones	ahageptest+techpartner
Prospective Inc	0	Corporate Services + (2) more	Greg Setser	Pete Prospect	ahageptest+prospective
Full Board Inc	0	Corporate Services + (2) more	Greg Setser	Phillip Board	ahageptest+fullboard@
Will's Lawn Care	0	Corporate Services + (2) more	Greg Setser	Will Weeder	ahageptest+willlawncar
WeWork Hard Corp	0	Corporate Services + (2) more	Greg Setser	Jen Brown	ahageptest+weworkhan
test	123123123	Advocacy + (2) more	AHA Admin	abhinandan patil	abhinandan.patil@gep.c

Rows Per Page: 10 1 - 10 Of 362 < >

Additional resources:

- <https://www.youtube.com/watch?v=OOAsU9SUooA>
- <https://www.w3.org/wiki/TabPanel>

`role="dialog"`
w/ `aria-labelledby`

`tabindex="-1"` then set focus
to dialog (or focus a control or
element inside the dialog)

Button with
`aria-haspopup="dialog"`



Returns
focus when
dismissed



Maintains keyboard
focus if modal

Closes with
ESC key

Dialog focus management

- Currently requires scripted focus detection and re-focus, or `tabindex="-1"` and `aria-hidden`
- Future:

```
<div inert>
  page content
</div>
<dialog modal>
  dialog content
</dialog>
```

ARIA Roles

- Avoid duplicating default roles:
`<button role="button">`
- ARIA roles override HTML native roles, but do NOT change functionality.
- Be very careful! You can destroy accessibility by adding one attribute.



`<input type="checkbox" role="radio">`

ARIA states and properties

- Elements can have more than one aria- attribute.
- ARIA states and properties should not override HTML properties if there's a conflict.
- Be very careful!



```
<input type="checkbox" aria-checked="false">
```

Differentiating regions/landmarks

```
<div role="navigation" aria-label="Main navigation">  
  <nav aria-label="Page navigation">
```

Generic regions/landmarks

```
<div role="region" aria-labelledby="filterheading">  
  <h2 id="filterheading">Filters</h2>  
  ...  
</div>
```

Generic regions must have an accessible name.

More ARIA

```
<button aria-expanded="false">Details</button>  
webaim.org/presentations/2020/examples/disclosure.htm
```

```
<button aria-pressed="true">Toggle Highlights</button>  
webaim.org/presentations/2020/examples/ariapressed.htm
```

```
<a aria-current="page | step | location | date | time | true">
```

Windows screen reader modes

- Reading / Virtual Cursor / Document
- Forms / Application

The current mode determines whether the screen reader or the browser handles most keyboard commands

Some ARIA roles (tree, slider, grid, tabpanel, menu, etc.) trigger forms/application mode.

You must ensure the user is aware and that the proper keyboard interactions are implemented.

Test with AND without a Windows screen reader.

Navigation menus are not application menus!

Navigation tabs (links) are not application tabs!

Data tables are not grids!

etc.

These trigger application mode and thus disable standard keyboard navigation.

Live Regions / Alerts

aria-live=assertive - read now

aria-live=polite - read at a pause

aria-live=off - read when the user encounters it

aria-busy

aria-atomic - read the entire region (true) or only what has changed (false)

aria-relevant - If aria-atomic=false, read additions, removals, text, or all

aria-controls

Special live regions: alert (important), status (not important), timer (always changing), marquee (same as aria-live="polite"), and log (updates added to the bottom)

Live Regions / Alerts

- Element must be present in the DOM when the page loads, then updated dynamically.
- Some highly dynamic content updates simply cannot be made accessible using ARIA
- Often best to have one messaging element, rather than multiple live regions
- Give users control over content updates

Single Page Applications

- Ensure good document structure
 - Use structural elements (<main> or role="region")
 - Update page titles to reflect content/state
 - Test keyboard navigation
 - Ensure only visible elements are navigable
 - Set focus () when necessary
 - \$('main').attr("tabIndex", -1) .focus () ;
 - Use live regions for messaging, if necessary

ARIA

With great power comes great responsibility!

Thank You!

webaim.org

- E-mail discussion list
- Monthly newsletter
- Tutorials, articles, and resources
- Blog