

<dev> + design

IMPROVING WEB ACCESSIBILITY IN 5 EASY STEPS

"KEEN" KUEHN



defining disability

Who do we think about when we talk about disabled users and website accessibility?

Accessibility is for those who...



C A N



USE A MOUSE

SEE THE SCREEN







• More than 1 billion people in the world have some form of disability. This corresponds to about 15% of the world's population.

But this statistic refers to permanent disabilities...

disability in numbers

- Wikipedia, World report on disability

defining disability

What comes to mind when we talk about disabled users and website accessibility?

Accessibility applies to those who... are disabled permanently... temporarily... or situationally!



C A





USEA MOUSE



Arm injury







Bright day



Public space





Getting older





Ear infection



accessibility for all!

it's about WHEN.

We're not making accessible websites for the **15%**.

Accessibility isn't about WHO...

We're making accessible websites for the **100%** of

people that will have a permanent, temporary, or

situational disability at some point in their lives.

good question

How do we know if a website has poor accessibility?

TEST IT!

It's easy to do a quick user test to see where your site may be falling short.

Keen Eye Design LLC



der notes

Accessibility testing doesn't have to be allencompassing or complicated. Everyone designers, devs, and QA — should all contribute to accessibility testing.

You may not have the time or resources this sprint to fix anything, but being *aware* of areas your site is failing accessibility allows you to target those areas and improve them later.

If you find an accessibility issue, put a story in the backlog to work on later when you have time. Just make sure you don't keep putting it off for too long!

esing accessibility

- 1. Look for proper color contrast*
 - Text, links, buttons, & backgrounds
- 2. Run a WCAG evaluation tool* on the page
- 3. Put your mouse cursor in the URL bar, and start pressing the TAB key
 - Can you tell where you are on the page?
 - Can you navigate through the menu and to all the links and buttons?
- 4. Make sure your code uses semantic html

Keen Eye Designuc

^{*}Links to suggested tools provided at the end of this presentation

good question

So...what exactly is "semantic" html?

Col-md-6 col-lg-8"> <!--"nav" role="navigation">

<a href="tall-but <a href="image-lo <a

pes practices

"Semantic" refers to tags with inherent meaning. They say exactly what they are.

Non-semantic tags have no meaning by themselves.

<header> <nav> <main> <section> <article> <footer>

S E M A N T I C

<h#>, , <button>, <input>, <a>, , are also semantic!



NOT SEMANTIC

<div></div>
<div></div>
<div></div>
<div></div>

Don't let and <div> define your page!



der notes

When you use <div> tags for everything, it can be difficult to keep track of what each <div> is doing.

This not only makes it impossible for screen readers to parse the page, it also makes the code harder to maintain and debug.





code check

What does bad accessibility look like?

Good design ≠ good accessibility!

Just because you have a well-designed website, doesn't mean it's accessible. Most accessibility is in how a page is coded behind the scenes. Check out these examples:

https://www.iflysouthern.com/

https://sessions.minnestar.org/events/43

code review

The 5 most common places where sites fail accessibility

90% of accessibility lies in the code



1. <h#> tags must be in order, and can't be skipped

WHY DOES THIS FAIL ACCESSIBILITY?

```
<body>
  <h3>My Website</h3>
  • • •
  <h5>Services</h5>
  • • •
  <h1><a href="">Book Now!</a></h1>
  • • •
  <h5>Add-on Services</h5>
  • • •
<body>
```

PASSES ACCESSIBILITY!

<body>

• • •

• • •

<body>

• • •

• • •

Don't use <h#> tags for their inherent SIZES. (No matter what your UX designer designs!) Always use <h#> tags in their hierarchal order.

Ideally, there should only be only **ONE <h1> on a page**. (Exceptions to this rule could be sites that use one long scroll, rather than separate pages, for content.)





- <h1 class="mediumfont">My Website</h1>
- <h2 class"smallfont">Services</h2>
- Book Now!
- <h2 class="smallfont">Add-on Services</h2>

Use CSS to style the type scale.



der notes

Ensuring proper <h#> order is one of the easiest places to fix accessibility. It can also be one of the most confusing!

This is where designers and developers end up on different pages—many designers don't comprehend how their designs get translated into code.

A designer may use an H3 because that's the SIZE they want, but it may not be the right h-tag

This is where you, as developers, need to enforce proper h-tag hierarchy, and use CSS to style the <h#> to the correct size the designer is looking for.

1. <h#> tags must be in order, and can't be skipped

Think of headlines like a term paper outline.

Web pages should be designed with a hierarchy in mind:

- Page title <h1>
- Headline <h2>
- Sub-head <h3>
- Content

Buttons, paragraphs, links, and calls-to-action (CTAs) should never be coded as <h#>, even if they're the same large size as a headline.

Don't use <h#> as a shortcut for size!

Use CSS to style the font size of your <button> or <a>, and keep <h#> for true headlines and sub-heads.





Textbook Outline Example

Mr. Vilain

I. Chapter Section Title

A. Bold Title

- a. Detail #1
- > b. Detail #2

Sub-Section Title

- a. Detail #1 Summarize the 1st Paragraph (1 sentence)
- cp> b. Detail #2 Summarize the 2nd Paragraph (1 sentence)
- c. Detail #3 Summarize the 3rd Paragraph (1 sentence)

2. Sub-Section Title

- <P> a. Detail #1 Summarize the 1st Paragraph (1 sentence)
- <b. Detail #2 Summarize the 2nd Paragraph (1 sentence)
- c. Detail #3 Summarize the 3rd Paragraph (1 sentence)

B. Bold Title

- a, Detail #1
- b. Detail #2

Sub-Section Title

- a. Detail #1 Summarize the 1st Paragraph (1 sentence)
- <b. Detail #2 Summarize the 2nd Paragraph (1 sentence)
- c. Detail #3 Summarize the 3rd Paragraph (1 sentence)

1. <h#> tags must be in order, and can't be skipped

HOW WOULD YOU CODE THIS HEADLINE?

WHO WE ARE

Health tech for a better future

From AI to connected care and beyond, our technology is building a bridge to better health for more people.

See how

WHO WE ARE <h1>Health tech for a better future</h1>

From AI to connected care and beyond, our technology is building a bridge to better health for more people.

"Who we are" is not a paragraph that's separate from the main headline, it's part of it. Nest it inside the <h1> and style it differently with a .

DO THIS INSTEAD!

</h1>

. . .





WHY IS THIS NOT IDEAL?

See how

<h1>WHO WE ARE Health tech for a better future

2. Color contrast is too low

ALWAYS CHECK YOUR COLOR CONTRAST

Acceptable color contrast depends on the font size and thickness. Decorative non-text items like borders and buttons also need to pass accessibility.

BEWARE! Just because a color passes on a white background, doesn't mean it'll pass on light gray.

Foreground Color		Background Color		Contrast Ratio	Share Results
#2F78DE	● #2F78DE ① <i>⋛</i>		Ð	3.66 : 1	Fail
WCAG Compliance	Results		Sample Large Text		
Element Type		АА	ААА	Sample normal text. on this page	Experiment with the settings to see the result here.
Aa Normal Text		X Fail	X Fail	Enter Your Text He	re
Aa Large Text		✓ Pass	X Fail		Button
Non Text Eleme	nts	✓ Pass	X Fail		



This medium blue passes on white, but FAILS on light gray

Keen Eye Design

lor		Background Color		Contrast Ratio	Share Results
F	0 ₹	#2EB7D3	Û	2.38 : 1	Fail
pliance Re	sults			Sample Lar	rge Text
pe		АА	ААА	on this page to see	the result here.
I Text		X Fail	× Fail	Enter Your Text Here	checkbox
Гext		X Fail	× Fail	Butto	n
ext Elements		X Fail	X Fail		
				OFF See Color	s In Grayscale

White on turquoise = FAIL

olor			Background Color		Contrast Ratio	Share Results
000	ŋ	≵	#2EB7D3	ð	8.83 : 1	Good
npliance	Results	5			Sample	Large Text
уре			АА	ААА	Sample normal text. E on this page to	xperiment with the settings o see the result here.
al Text			✓ Pass	✓ Pass	Enter Your Text Here	; is a checkbox
Text			✓ Pass	✓ Pass		Button
ext Element	S		✓ Pass	✓ Pass		
					OFF See 0	Colors In Grayscale

Black on turquoise = PASS

2. Color contrast is too low

WHY DOES THIS FAIL ACCESSIBILITY?

Improving lives through music education and performing arts



Chops, Inc. is a nonprofit 501(c)(3) organization based in Minneapolis, Minnesota. We offer year-round music performance and educational programs, including steel-drum bands, after-school drumlines, summer music camps, and artistic residencies. We also support Minnesota performing arts through fiscal sponsorship, providing technical expertise to help community-based performing groups reach their full potential. So come on over and join the fun!

JOIN NOW!

DONATE

Chops, Inc. is a nonprofit 501(c)(3) organization based in Minneapolis, Minnesota.

BEHIND THE IMAGE IS WHITE BACKGROUND

If the image doesn't load, or the user has images turned off, it's white-on-white text.

Accessibility checkers can't tell what colors are in your image; they can only look at the hex or rgba color behind your image (often defaulted as white).

Even though the background image provides enough contrast with the white text, the accessibility tool only sees white text on a white background, and will fail.

the image.



Fix: Make sure background images have a specified backup background-color in the same contrasting color family as

3. Use <a> to contain links, NOT , <div>, or onClick

WHY DOES THIS FAIL ACCESSIBILITY?

Click here to learn more

<div class="link" onClick="functionToExecute()"> Click here to learn more </div>

Too often I see links or buttons being coded with <div> and some javascript onClick functions to make them clickable.

This is really bad for accessibility!

Keyboard users won't be able to tab to or focus on that link, and screen readers won't read it as an interactive element.

Read our latest blog post on accessibility

There are different ways you can still use onClick functions, but they should be contained inside an anchor tag <a> or a <button>.

Always double-check to make sure keyboard users can navigate to the link and activate it without "clicking" a mouse.

And don't forget to make the anchor text relevant and descriptive! Never use "click here" for link text (some users can't "click!")





THIS PASSES!

```
Read our <a href="blog.html" class="link"</p>
onClick="functionToExecute()">latest blog post</a>
on accessibility.
```

3. Use <a> to contain links, NOT , <div>, or onClick

```
WHY DOES THIS FAIL?
```

```
<script type="text/javascript" src="jquery.js">
$(document).ready(function(){
  $(".tab").click(function(event){
      window.location='http://example.com';
  });
});
</script>
• • •
<div class="tab">tab text</div>
```

This tab isn't coded as semantic navigation; how are screen readers and keyboard navigators going to focus on this tabbed content/link?

OPTION 1:

```
<style>
</style>
• • •
```

OPTION 2:

Use links or buttons that are styled as tabs, so they're inherently focusable and work with keyboard navigation.

Be sure to wrap your tabbed navigation inside <nav> tags!



.tab {display: block;}

tab text

<button onclick="openPage(page, 'Home')"</pre> class="tablink">tab text</button>

4. Form <inputs> MUST have a <label>, and the id= must match for=

WHY DOES THIS FAIL ACCESSIBILITY?

<input id="news" type="checkbox" name="news" /> <label>Subscribe to our newsletter</label>

Subscribe to our newsletter

AND THIS....

<input id="search" title="search" type="text" /> <button type="submit">Search</button>

Search

EVERY <input> needs a corresponding <label>.

Each <input> MUST have an id=

Each <label> MUST have a for= that matches the input's id.

Don't use "title" to define your input. Tooltips aren't read by screen-readers! Instead, use CSS to visually hide the label on the screen, so screen-readers will still announce it.

THIS PASSES!

AND THIS, TOO....

* For more info on labels and inputs check out <u>https://www.w3.org/WAI/tutorials/forms/labels/</u>



<input id="news" type="checkbox" name="newletter" > <label for="news">Subscribe to newsletter</label>

<label for="search" class="notVisible">Search</label> <input id="search" type="text"> <button type="submit">Search</button>

4. Form <inputs> MUST have a <label>, and the id= must match for=

WHY DOES THIS FAIL ACCESSIBILITY?

<label>Address</label>

<input id="addressLine1" type="text" name="a1" /> <input id="addressLine2" type="text" name="a2" />

AND THIS....

```
<label>Preference A</label>
<label>Preference B</label>
<input id="preference" type="text" name="prefs" />
```

You cannot use an input without a label, and you cannot use a label without an input. Each input must have it's own matching label. No exceptions!

* For more info on labels and inputs check out <u>https://www.w3.org/WAI/tutorials/forms/labels/</u>

THIS PASSES ACCESSIBILITY!

Whenever possible, use <label> to associate text with form elements *explicitly*, using the for= and matching id=.

If you cannot control the id on the input, you can alternatively associate the label *implicitly* by nesting the input inside the label. But each input still needs its own label.

```
<label>Address Line 1
  <input id="addressLine1" type="text" name="a1" />
</label>
<label>Address Line 2
  <input id="addressLine2" type="text" name="a2" />
</label>
```



```
<label for="addressLine1">Address</label>
<input id="addressLine1" type="text" name="a1" />
<label for="addressLine2">Address</label>
<input id="addressLine2" type="text" name="a2" />
```

5. All images need alt tags

WHY DOES THIS FAIL ACCESSIBILITY?

WHY IS THIS BAD?

<img src="bkdg-texture.jpg" alt="Red and black</pre> textured background pattern" />

Sometimes images are simply decorative. You *could* provide alt text, but how useful is "Red and black textured background pattern" to a blind person?

THIS PASSES ACCESSIBILITY!

alt text.

If no alt tag is provided, some screen readers may read the file name, which could be very long and not useful to the reader at all!

THIS ALSO PASSES!

could do more harm than good!



If an image adds value to a web page or app, it is considered functional, and needs

-
- In these cases, a null (empty) alt text should be used (alt="") so the image can be ignored by assistive technologies.
- You can also use null alt text if the information provided by the image is already given in any adjacent text, like a caption.
- Don't add alt text if it doesn't provide value, or repeats content. Repetitive content

5. All images need alt tags

WHY IS THIS BAD?

<img src="logo.png" alt="Keen</pre> Eye Design logo" />

HINT: If this image wasn't a link, the alt text would be fine.

<img src="media/mini-banner.png" alt="Image of a</pre> robot" />

What image comes to mind when you read this alt text?

<img src="button.png" alt="Button that says Buy</pre> Now!" />

Don't use images for text, such as buttons or CTAs.

<img src="logo.png"</pre> alt="Home of Keen Eye Design" />

If an image is also a link, define where the link will take the user in the alt text.

<img src="media/mini-banner.png" alt="Robotic arm</pre> holding a scalpel in an hospital operating room" />

Alt text should be as descriptive as possible; imagine you're describing the image in detail to a blind person. Avoid using words like "image of" or "picture of" — the assistive technology will announce it's an image before reading the alt text.

<button class="cta"> Buy Now! </button>

Use CSS to style semantic code for inherent accessibility.



THIS IS MUCH BETTER

5. All images need alt tags

WHY IS THIS BAD?

<i class="search-icon" title="Search"></i></i>

The title attribute on this icon provides hoverable tooltips, but tooltips aren't read by screen-readers.

```
<img src="email.png" alt="Contact Us" />Contact Us
```

Here, you might think you've done the right thing by adding alt text to this email image, but the screen reader will end up reading "Contact Us" twice!

Site Search

the icon code.

Contact Us

A null/empty alt text (alt="") should be used in this case so the image can be ignored by assistive technologies.



THIS IS MUCH BETTER

If your icon isn't just decorative, and provides function (like a magnifying glass that opens a search box when clicked), use a visually hidden description before or after

TIP: Use HTML5 landmark tags to define areas of your page

WHY IS THIS BAD?

```
<body>
 <div class="header">
    <div class="nav">...</div>
 </div>
 <div class="content-container">
    <div class="main-content">
      <div class="content">...</div>
      <div class="content">...</div>
   </div>
 </div>
 <div class="footer">...</div>
</body>
```

While this will render fine on the screen, having everything in $\langle div \rangle$ s makes it impossible for assistive technologies to convey to the user how elements are grouped on the page, or where one section ends and another begins.

Plus, as a developer, it makes it harder to scan the code to find the section you're looking for!

THIS IS MUCH BETTER!

<body> <header> </header> <main> </main> </div> </body>

Use HTML5 semantic landmark tags in place of selective <div>s for better accessibility AND cleaner code that's easier to scan and debug.

* For a full list of HTML semantic tags, visit: https://www.w3schools.com/html/html5_semantic_elements.asp



```
<nav>...</nav>
<div class="content-container">
    <section>...</section>
    <section>...</section>
```

<footer>...</footer>

TIP: Use HTML5 landmark tags to define areas of your page

WHY IS THIS ALSO BAD?

```
<body>
 <main>
    <header>
     <nav>...</nav>
    </header>
    <div class="content-container">
      <section>...</section>
      <section>...</section>
    </div>
    <footer>...</footer>
 </main>
</body>
```

Don't nest everything inside the <main> tag — this defeats the purpose of having landmarks! <main> is for the main body of content. Headers, footers, and sidepanels are NOT main content.

THIS IS BETTER....

<body> <header> </header> <main> </main> </div> <footer>...</footer> </body>

<div>s still have their place as containers for grouping elements, or for wrapping styles around multiple elements.

* For a full list of HTML semantic tags, visit: https://www.w3schools.com/html/html5_semantic_elements.asp



<nav>...</nav> <div class="content-container"> <section>...</section> <section>...</section>



der noes

Remember these 5 landmark tags:

- <header> top section of your page
- <nav> any primary or secondary navigation
- <main> the main body/content area of your site
- <section> sections within the main content area
- <footer> bottom section of your page

Start incorporating these 5 tags into all your pages, and once they become second nature, look to replace other divs with landmark tags to further increase accessibility.

But don't get carried away, either — too many landmarks is just as bad not having any!

TIP: Use HTML5 landmark tags to define areas of your page

WHAT'S WRONG THIS THIS?

```
<body>
 <header>
    <nav>...</nav>
 </header>
 <div class="content-container">
    <main>
      <div aria-label="Tabbed Navigation">...</div>
      <section>...</section>
      <section>...</section>
    </main>
 </div>
 <footer>...</footer>
</body>
```

If you have more than 1 navigation on your page, you can use multiple <nav> tags as needed, just be sure to differentiate them by using "aria-label" or "arialabelledby".

THIS IS BETTER!

```
<body>
  <header>
    <nav aria-label="Main Navigation">...</nav>
  </header>
  <div class="content-container">
    <main>
      <nav aria-label="Tabbed Navigation">...</nav>
      <section>...</section>
      <section>...</section>
    </main>
  </div>
  <footer>...</footer>
</body>
```

* For a full list of HTML semantic tags, visit: https://www.w3schools.com/html/html5_semantic_elements.asp





code check

LET'S REVIEW....

- **1.** <h#> tags in the correct order, no skipping
- 2. <a> or <button> for keyboard accessible links
- 3. Semantic HTML tags instead of <div>
- 4. Every <input> needs a <label> (1:1 ratio)
 - The input's id= must match the label's for=
- 5. All images need alt tags
 - if purely decorative, leave null (alt="")

Bonus: Use HTML5 landmark tags to inherently improve accessibility without any extra effort!

"When UX doesn't consider ALL users, shouldn't it be known as "SOME User Experience" or... SUX?"

Billy Gregory Senior Accessibility Engineer









USERWAY COLOR CONTRAST CHECKER

Sample normal text sizes, large text sizes, and non-text elements to see your WCAG compliance results with <u>UserWay's Contrast Checker</u>.



WAVE EVALUATION

<u>Chrome extension</u> for visually evaluating web accessibility. Developed by WebAlM.org.





<u>Chrome browser extension</u> by Deque that integrates with your browser's Developer Tools shows you where the issues are and allows you to target the code.

*For today's purposes, we'll focus on issues we can target without any special assistive technologies.



AXE DEVTOOLS





corporate raining

Like this content?

contact Jodi at or visit us online at

Want more in-depth training? Let's bring it to your team!



- For on-site or virtual workshop sessions
- that are engaging and interactive,
- training@KeenEyeDesign.com
- KeenEyeDesign.com/workshop







- WEB



- \bigoplus



Jodi Kuenn

DESIGN SPECIALIST

user interface (ui)

user experience (ux)

design systems

web accessibility

ın/Jорі-Киенп

KeenEyeDesign.com

Hello@KeenEyeDesign.com