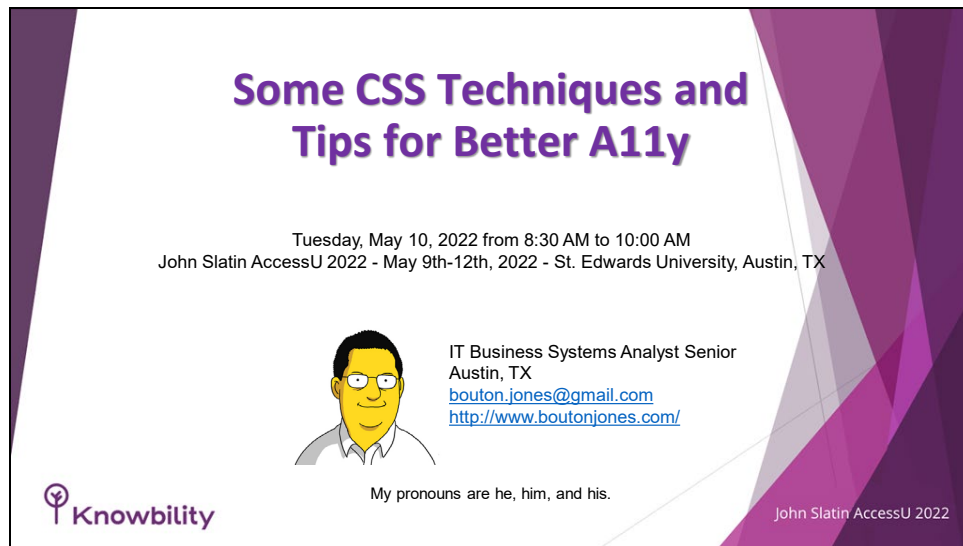



Slide 1 - **Presentation Title:** Some CSS Techniques and Tips for Better A11y




The slide features a white background with purple geometric shapes on the left and right sides. The title is centered at the top in a large, bold, purple font. Below the title, the date and time are listed, followed by the event name and location. A cartoon avatar of a man with glasses is positioned to the left of his contact information. The Knowbility logo is in the bottom left, and the event name is in the bottom right.

Some CSS Techniques and Tips for Better A11y

Tuesday, May 10, 2022 from 8:30 AM to 10:00 AM
John Slatin AccessU 2022 - May 9th-12th, 2022 - St. Edwards University, Austin, TX

 IT Business Systems Analyst Senior
Austin, TX
bouton.jones@gmail.com
<http://www.boutonjones.com/>

 My pronouns are he, him, and his.

John Slatin AccessU 2022

Tuesday, May 10, 2022 from 8:30 AM to 10:00 AM

John Slatin AccessU 2022 - May 9th-12th, 2022 - St. Edwards University, Austin, TX

Author: Bouton Jones

IT Business Systems Analyst Senior

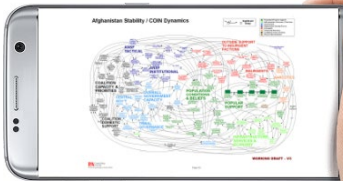
bouton.jones@gmail.com

My pronouns are he and him


Slide 2 - Life is too short to be spent photographing PowerPoint slides.

Life is too short to be spent photographing slides.

A handout containing all the slides and speaker notes can be found at <http://bit.ly/2y0p5kj> .



NEXT SLIDE



John Slatin AccessU 2022 Powered by Knowbility

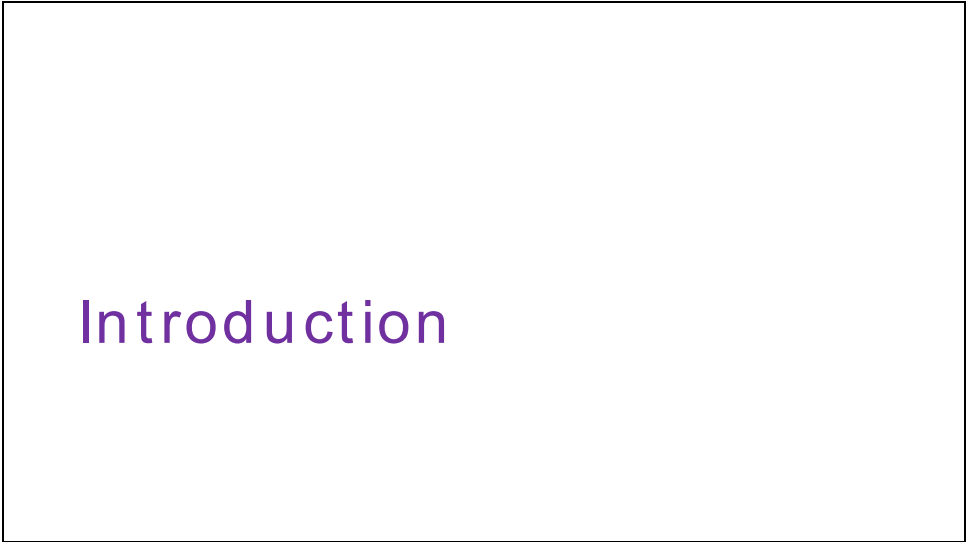
A handout containing all the slides and speaker notes can be found at <http://bit.ly/2y0p5kj> . (or <http://www.boutonjones.com/pub/a11y.html> .)

Slide 3 - QR Code for the web page where the handout can be downloaded.



Short URL: <http://bit.ly/2y0p5kI>

Long URL: <http://www.boutonjones.com/pub/a11y.html>



Slide 5 - This Session

This Session

This is primarily to show web designers and front-end or full-stack web developers how to apply HTML and CSS for better Accessibility. (Except for one mention, we won't cover JavaScript.)

This presentation will be an overview of HTML and CSS. It not be a deep dive.

Using both HTML and CSS in dedicated files allows for the separation of content and presentation which contributes to greater Accessibility.

John Slatin AccessU 2022 Powered by Knowbility

This is primarily to show web designers and front-end or full-stack web developers how to apply HTML and CSS for better Accessibility. (Except for one mention, we won't cover JavaScript.)

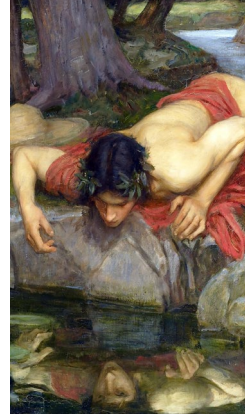
This presentation will be an overview of HTML and CSS. It not be a deep dive.

Using both HTML and CSS in dedicated files allows for the separation of content and presentation which contributes to greater Accessibility.

Slide 6 - About Bouton Jones

About Bouton Jones

- I am *not* a lawyer (IANAL.) I do *not* have a degree or certification in ADA or Accessibility.
- IT Business Systems Analyst since 2014.
- Previously a Web Designer and Web Developer. (Started web designing by “hand coding” HTML in the mid 1990’s.)
- Co-presenter of Accessibility sessions at
 - The Texas Association of Municipal Information Officers (TAMIO) Conference 2017.
 - E-Records Conference 2020 in Austin
- Presented a total of three sessions on Accessibility at the Adobe ColdFusion Summit in 2017 and 2018.
- Variously a participant, mentor, and advisor in Knowbility’s Accessibility Internet Rally



(The image is a detail of "Echo And Narcissus" by John William Waterhouse. Narcissus is in love with his reflection.)

- I am *not* a lawyer (IANAL.) I do *not* have a degree or certification in ADA or Accessibility.
- IT Business Systems Analyst since 2014.
- Previously a Web Designer and Web Developer. (Started web designing by “hand coding” HTML in the mid 1990’s.)
- Co-authored the QA Accessibility Guidelines adopted by the Commonwealth Office of Technology in 2004.
- Co-presenter of Accessibility Session at the Texas Association of Municipal Information Officers (TAMIO) Conference 2017.
- Presented a total of three sessions on Accessibility at the Adobe ColdFusion Summit in 2017 and 2018.
- Variously a participant, mentor, and advisor in Knowbility’s Accessibility Internet Rally

Slide 7 - *When* should we learn Accessibility?

When should we teach Accessibility?

Accessibility is not a new technology for us to learn. It's not a new skill, a new programming language, a new methodology, or new framework which may or may not benefit you.

It's building upon your existing skills and knowledge base so as to do a better job communicating more information to a larger audience.

We have to “shift left” in how we teach technology. We need to teach Accessibility as part of the technical stack. It should be taught *along* with programming languages, Microsoft Office, web design, business writing, etc., Until then we'll be playing catch up. We will by necessity add Accessibility at the end.

Accessibility is not a new technology for us to learn. It's not a new skill, a new programming language, a new methodology, or new framework which may or may not benefit you.

It's building upon your existing skills and knowledge base so as to do a better job communicating more information to a larger audience.

We have to “shift left” in how we teach technology. We need to teach Accessibility as part of the technical stack. It should be taught along with programming languages, Microsoft Office, web design, business writing, etc., Until then we'll be playing catch up. We will by necessity adding Accessibility at the end

Slide 8 - Type HTML the way God and Tim Berners-Lee intended --- with Semantic markup.

Type HTML the way God and Tim Berners-Lee intended ---
with Semantic markup.

*"The power of the Web is in its universality. Access by everyone
regardless of disability is an essential aspect."*

- Tim Berners-Lee, W3C Director and inventor of the World Wide
Web


John Slatin AccessU 2022 Powered by Knowbility

"The power of the Web is in its universality. Access by everyone regardless of disability is an essential aspect."

- Tim Berners-Lee, W3C Director and inventor of the World Wide Web

Slide 9 - A Brief History of HTML

A Brief History of HTML (1989 - 1994)



The site
is best
viewed in
Internet
Explorer

1989: At Cern, Tim Berners-Lee invents the Web and HTML

1992: Mosaic browser implements the IMG tag.

1994: HTML 2 is released; Netscape is formed; The World Wide Web Consortium (W3C) forms;

CITE:

- https://www.washington.edu/accesscomputing/webd2/student/unit1/module3/html_history.html
- <https://www.w3.org/People/Raggett/book4/ch02.html>

John Slatin AccessU 2022 Powered by Knowbility

Year	Event(s)
1989	At Cern, Tim Berners-Lee invents the Web and HTML.
1992	Mosaic browser implements the IMG tag.
1994	HTML 2 is released; Netscape forms; The World Wide Web Consortium forms;


IMAGE: An icon stating “This site is best viewed in Internet Explorer.”

CITE:

- https://www.washington.edu/accesscomputing/webd2/student/unit1/module3/html_history.html
- <https://www.w3.org/People/Raggett/book4/ch02.html>

Slide 10 - A Brief History of HTML (1995 – 1999)

A Brief History of HTML (1995 - 1999)



1995: HTML 2.0 is published; Microsoft releases the Internet Explorer browser; Netscape submits a proposal for frames and then quickly implements them; W3C advocates Cascading Style Sheets (CSS.)

1999: HTML 4.01 became the official standard

CITE:

- https://www.washington.edu/accesscomputing/webd2/student/unit1/module3/html_history.html
- <https://www.w3.org/People/Raggett/book4/ch02.html>

John Slatin AccessU 2022 Powered by Knowbility


IMAGE: An icon stating “This site is best viewed in Internet Explorer.”

Year	Event(s)
1995	<ul style="list-style-type: none">• HTML 2.0, published• Microsoft releases the Internet Explorer browser• Netscape submits a proposal for frames and then quickly implements them• W3C advocates Cascading Style Sheets (CSS.)
1999	HTML 4.01 became the official standard

CITE:

- https://www.washington.edu/accesscomputing/webd2/student/unit1/module3/html_history.html
- <https://www.w3.org/People/Raggett/book4/ch02.html>

A Brief History of HTML (2000 - 2012)



The site is best viewed in Internet Explorer

2000: HTML was rewritten as an XML language and called XHTML. It was similar to HTML but with stricter rules. It became an official standard. It was going to replace HTML 4.01.

2012: W3C and the WHATWG released HTML5 as a continuation and replacement of HTML 4.01.

CITE:

- https://www.washington.edu/accesscomputing/webd2/student/unit1/module3/html_history.html
- <https://www.w3.org/People/Raggett/book4/ch02.html>

John Slatin AccessU 2022 Powered by Knowbility

IMAGE: An icon stating “This site is best viewed in Internet Explorer.”

Year	Event(s)
2000	HTML was rewritten as an XML language and called XHTML. It was similar to HTML but with stricter rules. It became an official standard. It was going to replace HTML 4.01.
2012	W3C and the WHATWG released HTML5 as a continuation and replacement of HTML 4.01.

CITE:

- https://www.washington.edu/accesscomputing/webd2/student/unit1/module3/html_history.html
- <https://www.w3.org/People/Raggett/book4/ch02.html>

Slide 12 - Web Accessibility Initiative - Accessible Rich Internet Applications (ARIA)

Web Accessibility Initiative - Accessible Rich Internet Applications (WAI-ARIA):

A technical specification published by the World Wide Web Consortium (W3C) to increase the accessibility of web pages.

ARIA especially helps with *dynamic content* and advanced user interface controls developed with Ajax, HTML, JavaScript, and related technologies.

For example, with WAI-ARIA it is possible to identify a list of links as a navigation menu and to state whether it is expanded or collapsed.

ARIA Implements custom UI elements so that they are

1. usable by keyboard only users
2. understandable to users of assistive technology.

Two things that ARIA does:

1. Landmarks
2. States

John Slatin AccessU 2022 Powered by Knowbility

A technical specification published by the World Wide Web Consortium (W3C) to increase the accessibility of web pages.

ARIA especially helps with dynamic content and advanced user interface controls developed with Ajax, HTML, JavaScript, and related technologies.

For example, with WAI-ARIA it is possible to identify a list of links as a navigation menu and to state whether it is expanded or collapsed.

ARIA Implements custom UI elements so that they are

- *usable by keyboard only users*
- *understandable to users of assistive technology.*

Two things that ARIA does well:

1. Landmarks
2. States

HTML5 and WAI-ARIA

The first rule of WAI-ARIA is “If you can use a native HTML element [HTML5] or an attribute with the semantics and behavior you require already built in, instead of re-purposing an element and adding an ARIA role, state or property to make it accessible, then do so.”

HTML5 Elements	WAI-ARIA Landmark Roles	HTML5 Elements	WAI-ARIA Landmark Roles
<header>	role="banner"	<form>	role="form"
<nav>	role="navigation"	<section>	role="region"
<main>	role="main"		role="search"
<aside>	role="complementary"	<footer>	role="contentinfo"

John Slatin AccessU 2022 Powered by Knowbility

WAI-ARIA is “a technical specification published by the World Wide Web Consortium (W3C) that specifies how to increase the accessibility of web pages, in particular, dynamic content, and user interface components developed with Ajax, HTML, JavaScript, and related technologies.” (CITE: <https://en.wikipedia.org/wiki/WAI-ARIA>)

See <https://www.w3.org/WAI/standards-guidelines/aria/>

The first rule of WAI-ARIA is “If you can use a native HTML element [HTML5] or attribute with the semantics and behavior you require already built in, instead of re-purposing an element and adding an ARIA role, state or property to make it accessible, then do so.”

HTML5 Elements	WAI-ARIA Landmark Roles
<header>	role="banner"
<nav>	role="navigation"
<main>	role="main"
<aside>	Role="complementary"
<form>	role="form"
<section>	role="region"
	role="search"
<footer>	Role="contentinfo"

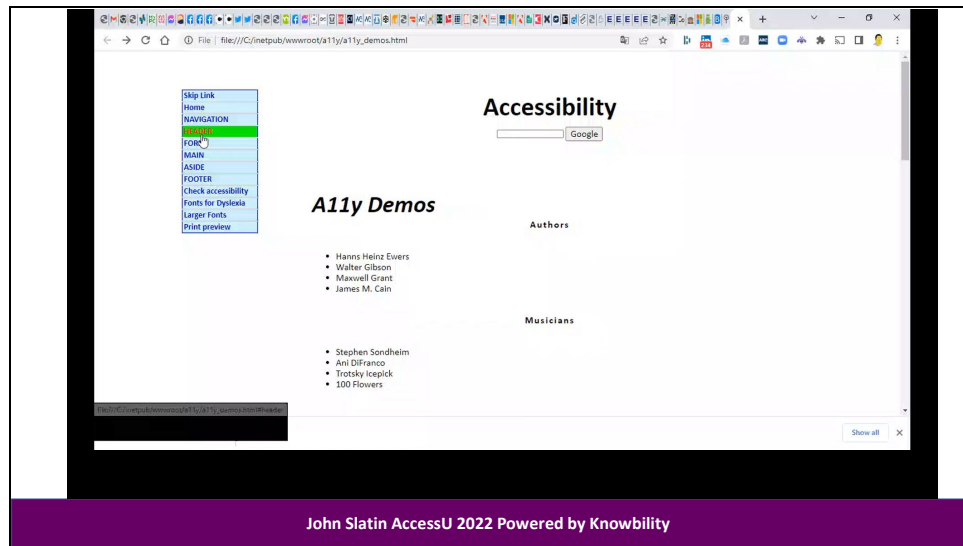
HTML Outlines

<p style="text-align: center;">HTML5 Elements</p> <pre> <BODY> <NAV id="navigation"> ... </NAV> <HEADER id="header"> ... </HEADER> <FORM id="form"> ... </FORM> <MAIN id="main"> <ASIDE id="aside"> ... </ASIDE> </MAIN> <FOOTER id="footer"> ... </FOOTER> </BODY> </pre>	<p style="text-align: center;">WAI-ARIA Landmark roles</p> <pre> <BODY> <DIV id="navigation" role="navigation"> ... </DIV> <DIV id="header" role="banner"> ... </HEADER> <FORM role="form" id="form"> ... </FORM> <DIV role="main" id="main"> <DIV id="aside" role="complementary"> ... </DIV> </DIV> <DIV id="footer" role="contentinfo"> ... </DIV> </BODY> </pre>
---	---

John Slatin AccessU 2022 Powered by Knowbility

HTML5	WAI-ARIA
<pre> <BODY> <NAV id="navigation"> ... </NAV> <HEADER id="header"> ... </HEADER> <FORM id="form"> ... </FORM> <MAIN id="main"> <ASIDE id="aside"> ... </ASIDE> </MAIN> <FOOTER id="footer"> ... </FOOTER> </BODY> </pre>	<pre> <BODY> <DIV id="navigation" role="navigation"> ... </DIV> <DIV id="header" role="banner"> ... </HEADER> <FORM role="form" id="form"> ... </FORM> <DIV role="main" id="main"> <DIV id="aside" role="complementary"> ... </DIV> </DIV> <DIV id="footer" role="contentinfo"> ... </DIV> </BODY> </pre>

Slide 15 - HTML5 Landmark Demo



This slide shows a video demonstrating the concept of navigating a web page by HTML5 elements or WAI-ARIA landmarks. It shows the focus moving through the HTML5 Elements: Navigation, Header, Form, Aside, and Footer

Loose vs Strict HTML

Related Buzz Words

- "Extra" Markup
- Structural Markup
- Standards Compliant
- Web Standards
- Semantic HTML
- Properly structured HTML
- Strict HTML
- Valid HTML
- Well-formed HTML
- Plain Old Semantic HTML (POSH)

John Slatin AccessU 2022 Powered by Knowbility

- "Extra" Markup
- Structural Markup
- Standards Compliant
- Web Standards
- Semantic HTML
- Properly structured HTML
- Strict HTML
- Valid HTML
- Well-formed HTML
- Plain Old Semantic HTML (POSH)

Loose HTML Table

```
<table class="standard">
  <tr>
    <td class="th">Signifer</td>
    <td class="th">Meaning</td>
  </tr>
  <tr>
    <td>First Row</td>
    <td>Yes</td>
  </tr>
  <tr>
    <td>Second Row</td>
    <td>No</td>
  </tr>
  <tr>
    <td>Third Row</td>
    <td>Maybe</td>
  </tr>
  <tr>
    <td>Forth Row</td>
    <td>It depends</td>
  </tr>
</tbody>
</table>
```

John Slatin AccessU 2022 Powered by Knowbility

```
<table class="standard">
  <tr>
    <td class="th">Signifer</td>
    <td class="th">Meaning</td>
  </tr>
  <tr class="OddRow">
    <td headers="signifer">First Row</td>
    <td headers="meaning">Yes</td>
  </tr>
  <tr class="EvenRow">
    <td headers="signifer">Second Row</td>
    <td headers="meaning">No</td>
  </tr>
  <tr class="OddRow">
    <td headers="signifer">Third Row</td>
    <td headers="meaning">Maybe</td>
  </tr>
  <tr class="EvenRow">
    <td headers="signifer">Forth Row</td>
    <td headers="meaning">It depends</td>
  </tr>
</tbody>
</table>
```

Slide 19 - Strict HTML Table

Strict HTML Table

```
<table class="standard">
  <caption>A Table</caption>
  <thead>
    <tr>
      <th scope="col" id="signifer"
abbr="signifer">Signifer</th>
      <th scope="col" id="meaning"
abbr="meaning">Meaning</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <td headers="signifer">First Row</td>
      <td headers="meaning">Yes</td>
    </tr>
    <tr>
      <td headers="signifer">Second Row</td>
      <td headers="meaning">No</td>
    </tr>
    <tr>
      <td headers="signifer">Third Row</td>
      <td headers="meaning">Maybe</td>
    </tr>
    <tr>
      <td headers="signifer">Forth Row</td>
      <td headers="meaning">It depends</td>
    </tr>
  </tbody>
</table>
```

John Slatin AccessU 2022 Powered by Knowbility

Build dynamic tables that are section 508 compliant.

In the strict HTML (correct version):

- Caption tags
- Table head and table body tag pairs
- Uses TH tags instead of formatted TD tags
- scope attribute in TH
- id attribute in TH
- header attribute in TD

(NOTE: The summary attribute is deprecated in HTML5.)

```
<table class="standard">
  <caption>A Table</caption>
  <thead>
    <tr>
      <th scope="col" id="signifer" abbr="signifer">Signifer</th>
      <th scope="col" id="meaning" abbr="meaning">Meaning</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <td headers="signifer">First Row</td>
      <td headers="meaning">Yes</td>
    </tr>
    <tr>
```

```
    <td headers="signifer">Second Row</td>
    <td headers="meaning">No</td>
</tr>
<tr>
    <td headers="signifer">Third Row</td>
    <td headers="meaning">Maybe</td>
</tr>
<tr>
    <td headers="signifer">Forth Row</td>
    <td headers="meaning">It depends</td>
</tr>
</tbody>
</table>
```

ASIDE:

“By using the THEAD tag, the rows that you define within the THEAD tag will be displayed at the top of every printed page. Additionally, you can use the TFOOT tag to define the table footer. By using the TFOOT tag, the rows that you define within the TFOOT tag will be displayed at the bottom of every page. “

Ask Ben: Optimizing Tables For Printing
By Ben Nadel on September 12, 2008

Tables – Loosely Written HTML vs. Strict HTML

Table Made with Loosely Written HTML

Hotel	Room Rate Including Tax	Resort Fee	Total
The Hallucination	\$196.00	\$39.68	\$235.68
The Venusian	\$269.00	\$44.22	\$313.22
Detroit	\$153	\$39.68	\$192.68

Strict HTML (NVDA) **Strict HTML (JAWS)** **Strict HTML (ChromeVox)**

Las Vegas Hotels

Hotel	Room Rate Including Tax	Resort Fee	Total
The Hallucination	\$196.00	\$39.68	\$235.68
The Venusian	\$269.00	\$44.22	\$313.22
Detroit	\$153	\$39.68	\$192.68

John Slatin AccessU 2022 Powered by Knowbility

The first example is a simple HTML table. It doesn't use "strict" -- or semantic -- HTML. It's hard to understand when read by a screen reader.

The second table uses semantic HTML. It's easier for a blind user to understand the table with strict (Semantic) HTML. (NOTE: The experience with the strict (semantic) HTML varies depending on the screen reader used.)

Three Screen Readers on Strict HTML Tables

NVDA	JAWS	ChromeVox
<ul style="list-style-type: none">o Identifies the table.o Identifies the number of rows and the number of columns.o Reads the caption ("Las Vegas Hotels".)o Does not read the summary.o Reads the labels in the header row twiceo Repeats the labels in the data rows	<ul style="list-style-type: none">o Reads the summary.o Identifies the table.o Identifies the number of rows and the number of columns.o Reads the caption ("Las Vegas Hotels".)o Reads the labels in the header once.o Doesn't repeat the labels in the data rows	<ul style="list-style-type: none">o Does not read the summary.o Does not identify the table.o Does not identify the number of rows and the number of columns.o Reads the caption ("Las Vegas Hotels".)o Reads the labels in the header once.o Doesn't repeat the labels in the data rows.

John Slatin AccessU 2022 Powered by Knowbility

NVDA

- o Identifies the table.
- o Identifies the number of rows and the number of columns.
- o Reads the caption ("Las Vegas Hotels".)
- o Does not read the summary.
- o Reads the labels in the header row twice
- o Repeats the labels in the data rows

JAWS

- o Reads the summary.
- o Identifies the table.
- o Identifies the number of rows and the number of columns.
- o Reads the caption ("Las Vegas Hotels".)
- o Reads the labels in the header once.
- o Doesn't repeat the labels in the data rows.

ChromeVox

- o Does not read the summary.
- o Does not identify the table.
- o Does not identify the number of rows and the number of columns.
- o Reads the caption ("Las Vegas Hotels".)
- o Reads the labels in the header once.
- o Doesn't repeat the labels in the data rows.

Additional Benefits to Strict HTML

- Inclusive Design
- Improved Usability
- Better Search Engine Optimization (SEO)
- Mobile Friendly and Responsive Design (for multiple platforms)
- Cross-browser Compatibility
- Future-Proof Development
- Easier Maintenance / Lower Maintenance Cost
- Improved Technical Performance

John Slatin AccessU 2022 Powered by Knowbility

- Better Search Engine Optimization (SEO)
- Lower Maintenance Cost
- Improved Usability
- Mobile Friendly and Responsive Design (for multiple platforms)
- Inclusive Design
- Browser Compatibility or Cross-browser Compatibility
- Easier Maintenance / Future-Proof
- Improved Technical Performance

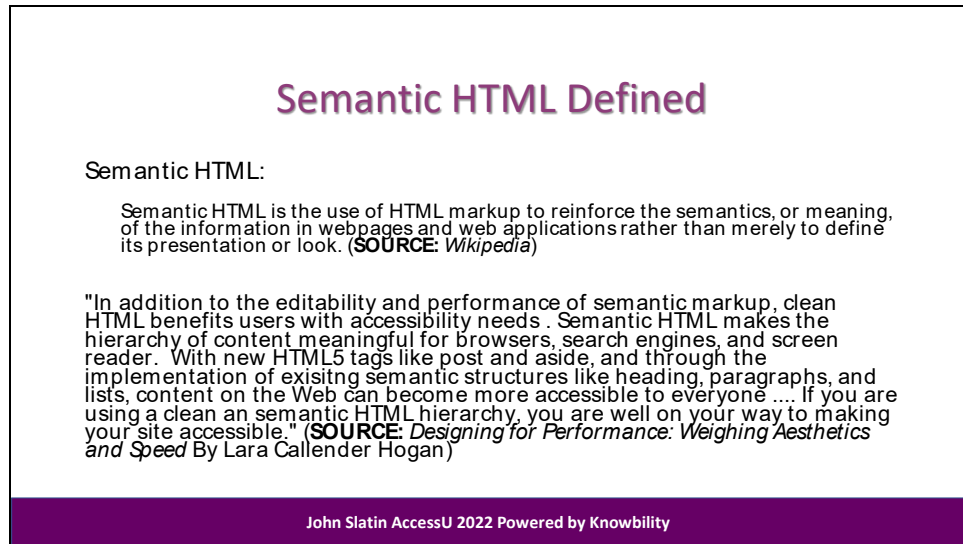
Validators from the W3C

- For Web documents in HTML, XHTML, etc., visit <https://validator.w3.org/>
- For CSS, try <https://jigsaw.w3.org/css-validator/>

John Slatin AccessU 2022 Powered by Knowbility

- For Web documents in HTML, XHTML, etc., visit <https://validator.w3.org/>
- For CSS, try <https://jigsaw.w3.org/css-validator/>

Semantic HTML and The Separation of Presentation and Content



Semantic HTML Defined

Semantic HTML:

Semantic HTML is the use of HTML markup to reinforce the semantics, or meaning, of the information in webpages and web applications rather than merely to define its presentation or look. (SOURCE: *Wikipedia*)

"In addition to the editability and performance of semantic markup, clean HTML benefits users with accessibility needs . Semantic HTML makes the hierarchy of content meaningful for browsers, search engines, and screen reader. With new HTML5 tags like post and aside, and through the implementation of existitng semantic structures like heading, paragraphs, and lists, content on the Web can become more accessible to everyone If you are using a clean an semantic HTML hierarchy, you are well on your way to making your site accessible." (SOURCE: *Designing for Performance: Weighing Aesthetics and Speed* By Lara Callender Hogan)

John Slatin AccessU 2022 Powered by Knowbility

Semantic HTML is the use of HTML markup to reinforce the semantics, or meaning, of the information in webpages and web applications rather than merely to define its presentation or look. (SOURCE: *Wikipedia*)

"In addition to the editability and performance of semantic markup, clean HTML benefits users with accessibility needs . Semantic HTML makes the hierarchy of content meaningful for browsers, search engines, and screen reader. With new HTML5 tags like post and aside, and through the implementation of existitng semantic structures like heading, paragraphs, and lists, content on the Web can become more accessible to everyone If you are using a clean an semantic HTML hierarchy, you are well on your way to making your site accessible." (SOURCE: *Designing for Performance: Weighing Aesthetics and Speed* By Lara Callender Hogan)

Presentational HTML vs. Semantic HTML

Presentational HTML (Appearance)	Semantic HTML (Significance)
<ul style="list-style-type: none">• Strong• Bold• Font size• Font Color• Background Color• Strikethrough• Underline	<ul style="list-style-type: none">• Title• Heading• List• Paragraph• Blockquote• Code• Definition• Cite

John Slatin AccessU 2022 Powered by Knowbility

A table comparing presentational and semantic HTML.

Presentational HTML (Appearance)	Semantic HTML (Significance)
<ul style="list-style-type: none">• Strong• Bold• Font size• Font Color• Background Color• Strikethrough• Underline	<ul style="list-style-type: none">• Title• Heading• List• Paragraph• Blockquote• Code• Definition• Cite

Separation of Presentation and Content (HTML and Styles)

- In the late 1990s the HTML files contained content (plain text), semantic markup, and presentational markup.
- Later it became possible (and preferable) to move all the presentational markup to Styles (e.g., CSS) leaving only plain text and semantic markup in the HTML files.
- JavaScript used to be embedded in the HTML documents. Later it was moved to .js files. And later still to JavaScript libraries.
- Currently Code Reuse through modular development (e.g. MVC, OOP, & Frameworks) is an accepted best practice.

John Slatin AccessU 2022 Powered by Knowbility

- In the late 1990s the HTML files contained content (plain text), semantic markup, and presentational markup.
- Later it became possible (and preferable) to move all the presentational markup to the CSS files leaving only plain text and semantic markup in the HTML files.
- JavaScript used to be embedded in the HTML documents. Later it was moved to .js files. And later still to JavaScript libraries.
- Similarly in the early days of ColdFusion, it was the accepted standard to include the three elements of model, view, and controller in a single template. Look at Forta's earliest CFWACK books for confirmation.
- Currently Code Reuse through modular development (e.g. CFCs, MVC, & Frameworks) is an accepted best practice.

"CSS is independent of HTML and can be used with any XML-based markup language. The separation of HTML from CSS makes it easier to maintain sites, share style sheets across pages, and tailor pages to different environments. This is referred to as the separation of structure (or: content) from presentation." <https://www.w3.org/standards/webdesign/htmlcss>

HTML5 and CSS3

"In addition to the editability and performance of semantic markup, **clean HTML benefits users with accessibility needs** . Semantic HTML makes the hierarchy of content meaningful for browsers, search engines, and screen reader. With new HTML5 tags like post and aside, and through the implementation of existing semantic structures like heading, paragraphs, and lists, content on the Web can become more accessible to everyone **If you are using a clean an semantic HTML hierarchy, you are well on your way to making your site accessible.**"

CITE: *Designing for Performance: Weighing Aesthetics and Speed* By Lara Callender Hogan

"CSS is independent of HTML and can be used with any XML-based markup language. The separation of HTML from CSS makes it easier to maintain sites, share style sheets across pages, and tailor pages to different environments. This is referred to as the separation of structure (or: content) from presentation."

CITE: <https://www.w3.org/standards/webdesign/htmlcss>

John Slatin AccessU 2022 Powered by Knowbility

"In addition to the editability and performance of semantic markup, clean HTML benefits users with accessibility needs . Semantic HTML makes the hierarchy of content meaningful for browsers, search engines, and screen reader. With new HTML5 tags like post and aside, and through the implementation of existing semantic structures like heading, paragraphs, and lists, content on the Web can become more accessible to everyone If you are using a clean an semantic HTML hierarchy, you are well on your way to making your site accessible." (**SOURCE:** *Designing for Performance: Weighing Aesthetics and Speed* By Lara Callender Hogan)

"CSS is independent of HTML and can be used with any XML-based markup language. The separation of HTML from CSS makes it easier to maintain sites, share style sheets across pages, and tailor pages to different environments. This is referred to as the separation of structure (or: content) from presentation." <https://www.w3.org/standards/webdesign/htmlcss>

Benefits of Separation

- Maximizes consistency across web sites and domains
- Reduces file sizes
- Makes development and maintenance easier and faster
- Requires less technical knowledge for most of your content authors
- Minimizes bad A11y by limiting bad markup, bling, and flair.

John Slatin AccessU 2022 Powered by Knowbility

- Maximum consistency across web sites and domains
- Reduced file sizes
- Easier and faster development and maintenance
- Less technical knowledge requirements for most of your content authors
- Minimizing bad A11y by limiting bad markup, bling, and flair.

Slide 30 - Navigation

Sighted users navigate web pages and document *spatially*.

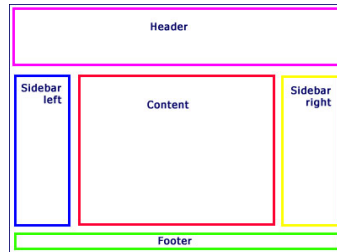


Image Source: <http://n390.com/content/page-layout>

Visually impaired users navigate *sequentially*. Without Tags they must read the document from beginning to end.

John Slatin AccessU 2022 Powered by Knowbility

For sighted users web pages are ordered **spatially**: Headers, Footers, Navigation Bars, and Content Areas. It's fast, easy, and somewhat intuitive for us to move our focus around a page. But for the users of screen readers, web pages are ordered **sequentially**: beginning, middle, and end. Depending on the page it can be difficult to get past the header and the navigation bar to the page content. That's where the concept of the "skip link" comes into play.

And if a document is not "tagged" properly a blind users is unable to navigate to content in the middle or near the end of a document with reading through everything that comes before.

Demonstrating the Power of CSS

CSS Zen Garden
A demonstration of what can be accomplished through CSS-based design. Select any style sheet from the list to load it into this page. The HTML remains the same, the only thing that has changed is the external CSS file.

URL: <http://www.csszengarden.com/>

John Slatin AccessU 2022 Powered by Knowbility

A demonstration of what can be accomplished through CSS-based design. Select any style sheet from the list to load it into this page. The HTML remains the same, the only thing that has changed is the external CSS file.

<http://www.csszengarden.com/>

Slide 32 - CSS Zen Garden Demo Video



HTML vs. Styles (E.G., Typefaces)

FONT tag	FONT Style Selectors
<pre>Hello World!</pre>	<pre>.joker { font-family: 'Times New Roman', Times, serif; font-style: normal; font-weight: bold; font-size: 2.5em; color: red; } Hello World!</pre>

John Slatin AccessU 2022 Powered by Knowbility

The Font Tag is not supported in HTML5.

FONT tag	FONT Style Selectors
<pre>Hello World!</pre>	<pre>.joker { font-family: 'Times New Roman', Times, serif; font-style: normal; font- weight: bold; font-size: 2.5em; color: red; } Hello World!</pre>

Slide 34 - Two Bulleted Lists

Two Bulleted Lists

Bulleted List with Decorative Markup	Bulleted List with Semantic Markup
<p>Bulleted Lists with Decorative Markup</p> <ul style="list-style-type: none">● Groucho● Chico● Harpo● Zeppo● Gummo● Karl <pre><p> Groucho
 Chico
 Harpo
 Zeppo
 Gummo
 Karl</p></pre>	<p>Bulleted Lists with Semantic Markup</p> <ul style="list-style-type: none">● Groucho● Chico● Harpo● Zeppo● Gummo● Karl <pre><ul type="disc" class="blueball"> Groucho Chico Harpo Zeppo Gummo Karl </pre>

The first example would be have been more tedious if it had been laid out in a table (which I've seen!)
Use tables for **tabular** data only. Don't use it for lay out.

John Slatin AccessU 2022 Powered by Knowbility

Don't Make Bulleted Lists with Decorative Markup!

In order for screen readers to recognize a vertical list as a list, the list must be formatted using semantic markup.

Screen readers don't identify the first list (created with decorative markup) as a list.

All screen readers recognize the second list (created with semantic markup) as a list.

I've seen bulleted lists made with decorative markup and laid out using tables which is more tedious than decorative markup alone.

Here's the source code:

Bulleted List Created with Decorative Markup	Bulleted List Created with Semantic Markup
<pre><p> Groucho
 Chico
 Harpo
</pre>	<pre><ul type="disc" class="blueball"> Groucho Chico Harpo Zeppo Gummo </pre>

Slide 35 - Headers Don't Need to Look Like Headers

Headers Don't Need to Look Like Headers

EXECUTIVE SUMMARY:	IT Business Systems Analyst Senior with a 10-year background of full-time professional web development experience maintaining and troubleshooting existing applications and developing and debugging new applications from scratch using ColdFusion and SQL.
OBJECTIVES:	Develop high quality and extremely usable and accessible web applications and business systems, solve problems through technology, and address my employer's requirements with multiple skills and a continually expanding knowledge base and skill set. Make lasting improvements for my employer.
RECOGNITION and AWARDS:	<ul style="list-style-type: none">• First Place (as part of the four member "Indy #2" team) in the Accessibility Internet Rally (AIR) 2019 on January 16, 2020. AIR is powered by Knowbility
EDUCATION:	University of Texas, Austin Texas. Bachelor of Fine Arts Candidate. BACHELOR of ARTS in FINE ARTS, GPA: 3.3

John Slatin AccessU 2022 Powered by Knowbility

This section of a web page looks like a table, but it this is actually composed of H2 tags, div tags, and text.

Slide 36 - Modify the Header Styles

Modifying the Header Styles

CSS	HTML
<pre>H2.resumeTHleft { display: block; font-weight: bold; color: #0000FF; font-family: Calibri, Serif; font-size: 100%; text-align: left; background-color: #DDDDFF; }</pre>	<pre><div class="resumeWrapper"> <h2 class="resumeTHleft">EXECUTIVE
SUMMARY:</h2> <div class="resumeTDright"> ... </div> </div></pre>

John Slatin AccessU 2022 Powered by Knowbility

CSS	HTML
<pre>H2.resumeTHleft { display: block; font-weight: bold; color: #0000FF; font-family: Calibri, Serif; font-size: 100%; text-align: left; background-color: #DDDDFF; }</pre>	<pre><div class="resumWrapper"> <h2 class="resumeTHleft">EXECUTIVE
SUMMARY:</h2> <div class="resumeTDright"> ... </div> </div></pre>

Tips, Techniques, and Best Practices

Some Best Practices

1. Indent your HTML and CSS for legibility
2. Comment your HTML and CSS for understanding
3. Limit the number of font sizes on a site to three (not counting header sizes such as H1, H2, etc.)
 - Default font size
 - Secondary font (slightly smaller than default)
 - Small font size (for captions, labels, etc.)
4. Limit the number of typefaces to three or four.
 - Decorative typeface (including Fantasy typefaces) for title and headers
 - Default body typeface (should be easy to read)
 - Sans Serif typeface for small text
 - Monospace typeface

John Slatin AccessU 2022 Powered by Knowbility

1. Indent your HTML and CSS for legibility

2. Comment your HTML and CSS for understanding

3. Limit the number of font sizes on a site to three (not counting header sizes such as H1, H2, etc.,)


- Default font size
- Secondary font (slightly smaller than default)
- Small font size (for captions, labels, etc.,)

4. Limit the number of typefaces to three (or four.)


- Decorative typeface (including Fantasy typefaces) for title and headers
- Default body typeface (should be easy to read)
- Sans Serif typeface for small text
- Monospace typeface

Slide 39 - "Skip Links" Demonstration (VIDEO)

"Skip Links" Demonstration (VIDEO)



Pitfall: If you hide the skip link from sighted users, you might be preventing those users who can't use mice from taking advantage of the technique.



John Slatin AccessU 2022 Powered by Knowbility

For sighted users web pages are ordered **spatially**. But for the users of screen readers, web pages are ordered **sequentially**. Depending on the page it can be difficult to get past the header and the navigation bar to the page content. That's where the concept of the "skip link" comes into play.

The common practice is to hide the skip link or make it invisible in browsers but discoverable to screen readers. This helps with the aesthetic and the skip links remain available to blind users.

VIDEO: A display of a "skip link" in action.

Pitfall: If you hide the skip link from sighted users, you might be preventing those users who can't use mice from taking advantage of the technique. The version of the "skip link" technique demonstrated here benefits blind users as well as users who navigate with the keyboard only.

See also <https://webaim.org/techniques/skipnav/>

"Skip Link" Resources

Understanding SC 2.4.1 - Bypass Blocks (w3.org)

<https://www.w3.org/TR/UNDERSTANDING-WCAG20/navigation-mechanisms-skip.html>

Web Accessibility: Skip Navigation Links

Andrew Jones. Dec 17, 2013

<https://www.bignerdranch.com/blog/web-accessibility-skip-navigation-links/>

"Skip Navigation" Links (WebAIM)

<https://webaim.org/techniques/skipnav/>

Understanding SC 2.4.1 - Bypass Blocks (w3.org)

<https://www.w3.org/TR/UNDERSTANDING-WCAG20/navigation-mechanisms-skip.html>

Web Accessibility: Skip Navigation Links

Andrew Jones. Dec 17, 2013

<https://www.bignerdranch.com/blog/web-accessibility-skip-navigation-links/>

"Skip Navigation" Links (WebAIM)

<https://webaim.org/techniques/skipnav/>

Best Practice: Text Size in CSS

```
html {
  font-size: 100%; // 100% of the browser's base font
  size
}

p, td, th, ul, ol, li, dl, dt, dd, span, div, caption {
  font-size: 1rem; // relative to the HTML font size
}

/* Avoid absolute lengths (in, cm, mm, pt, pc) except
when printing. They do not resize well. */
```

John Slatin AccessU 2022 Powered by Knowbility

```
ht {
font-size: 100%; // 100% of the browser's base font size
}
p, td, th, ul, ol, li, dl, dt, dd, span, div, caption
{ font-size: 1rem; // relative to the HTML font size
}
}
```

/* Avoid absolute lengths (in, cm, mm, pt, pc) except when printing. They do not resize well. */

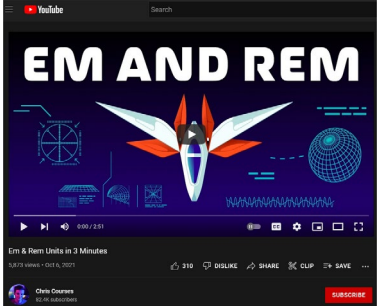
It's generally considered best practice for accessibility to set your base font-size to 100% on the <html> element, and use relative units to handle all other font-size rules in your CSS. This allows users to change the base font size in their browser, and have control over the font sizing for webpages."

CITE:

- <https://blog.logrocket.com/the-elements-of-responsive-typography/>
- <https://betterwebtype.com/articles/2019/06/16/5-keys-to-accessible-web-typography/>

Slide 42 - Why REM? Why not EM?

Why REM? Why not EM?



The image shows a YouTube video player thumbnail. The title is 'EM AND REM' in large white letters. Below the title is a stylized graphic of a white bird with orange wings. The video is by 'Chris Coopers' and has 310 likes. The video title is 'EM & Rem Units in 3 Minutes'.

EM – The size is based on the element's parent tags

REM – The size is based on the root tag (i.e., the HTML or Body element.)

URL: https://youtu.be/j7zf_iZjQB4

John Slatin AccessU 2022 Powered by Knowbility

- EM – The size is based on the element's parent tags
- REM – The size is based on the root tag (i.e., the HTML or Body element.)

For better explanation, view the **VIDEO** "EM and REM" at https://youtu.be/j7zf_iZjQB4

For a longer explanation, view the **VIDEO** "Why you shouldn't set font-sizes using em" by Kevin Powell (13:42) at <https://youtu.be/pautqDqa54I>

Best Practice: Select Your Site's Typeface(s)

```
HTML, BODY {
    font-family: Calibri,
    "Trebuchet MS", Arial,
    Georgia, "Times New Roman",
    Helvetica, Serif;
    font-size: 100%;
}

p, td, th, ul, ol, li, dl, dt,
dd, span, div, caption {
    font-size: 1REM;
    line-height: 1.5rem;
    margin-bottom: 2rem;
    margin-top: 0rem;
    letter-spacing: .12rem;
    word-spacing: .16rEM;
}

/* WCAG 2.1 1.4.12 Spacing */
```

John Slatin AccessU 2022 Powered by Knowbility

```
HTML, BODY {
    font-family: Calibri, "Trebuchet MS", Arial, Georgia, "Times New
Roman", Helvetica, Serif;
    font-size: 100%;
}
```

```
p, td, th, ul, ol, li, dl, dt, dd, span, div, caption {
    font-size: 1REM;
    line-height: 1.5rem;
    margin-bottom: 2rem;
    margin-top: 0rem;
    letter-spacing: .12rem;
    word-spacing: .16rEM;
}
```

```
/*
WCAG 2.1 1.4.12 Spacing
o Line height to at least 1.5 times the font size;
o Spacing following paragraphs to at least 2 times the font size;
o Letter spacing (tracking) to at least 0.12 times the font size;
o Word spacing to at least 0.16 times the font size.
*/
```

TIP: Codify your Color Schemes and Typefaces in your CSS

WCAG guidelines:

- Large Text (greater than or equal to 24px) should have a color contrast ratio of at least 3:1.
- Most text (less than 24px) should have a color contrast ratio of at least 4.5:1.
- Bold text with a size less than 18.5px should have a color contrast of 4.5:1 minimum.

Read "Contrast and Color Accessibility" at <https://webaim.org/articles/contrast/>

John Slatin AccessU 2022 Powered by Knowbility

WCAG 2.0 guidelines:

- Large Text (greater than or equal to 24px) should have a color contrast ratio of at least 3:1.
- Most text (less than 24px) should have a color contrast ratio of at least 4.5:1.
- Bold text with a size less than 1
- 8.5px should have a color contrast of 4.5:1 minimum.

Read "Contrast and Color Accessibility" at <https://webaim.org/articles/contrast/>

Slide 45 - TIP: The Benefit of a Grayscale Wireframe



IMAGE: Screen capture of the FWC in complete greyscale.

- When demonstrating a early design of a web site to clients. It will pre-empt discussions of site aesthetics so that the clients and the design team can focus of layout, site architecture, usability, accessibility, and content.
- Prevent color-blind issues

Sample CSS for Grayscale Theme

```
BODY {  
    background-color: #CCCCCC; //  
}  
IMG {  
    filter: grayscale(100%);  
}
```

John Slatin AccessU 2022 Powered by Knowbility

```
BODY {  
    background-color: #CCCCCC;  
}
```

```
IMG {  
    filter: grayscale(100%);  
}
```

Slide 47 - Alternating Row Colors in Tables (Example)

Alternating Row Colors in Tables (Example)

Signifier	Meaning
First Row	Yes
Second Row	No
Third Row	Maybe
Forth Row	It depends

John Slatin AccessU 2022 Powered by Knowbility

Image: example table with alternating row background colors

Signifier	Meaning
First Row	Yes
Second Row	No
Third Row	Maybe
Fourth Row	It depends

Alternating Row Colors in Tables (Old CSS)

HTML

```
<tr class="OddRow">
  <td headers="signifer">First Row</td>
  <td headers="meaning">Yes</td>
</tr>
<tr class="EvenRow">
  <td headers="signifer">Second
Row</td>
  <td headers="meaning">No</td>
</tr>
```

Old CSS

```
.OddRow {
  background-
color: #CCCCCC;
}

.EvenRow{
  background-
color: #FFFFFF;
}
```

HTML	Old CSS
<pre><tr class="OddRow"> <td headers="signifer">First Row</td> <td headers="meaning">Yes</td> </tr> <tr class="EvenRow"> <td headers="signifer">Second Row</td> <td headers="meaning">No</td> </tr></pre>	<pre>.OddRow { background-color: #CCCCCC; } .EvenRow{ background-color: #FFFFFF; }</pre>

Alternating Row Colors in Tables (New CSS)

HTML	New CSS
<pre> <tr> <td headers="signifer">First Row</td> <td headers="meaning">Yes</td> </tr> <tr> <td headers="signifer">Second Row</td> <td headers="meaning">No</td> </tr> </pre>	<pre> tr:nth-child(even) { background-color: #CCCCCC; } tr:nth-child(odd) { background-color: #FFFFFF; } </pre>

John Slatin AccessU 2022 Powered by Knowbility

HTML	New CSS
<pre> <tr> <td headers="signifer">First Row</td> <td headers="meaning">Yes</td> </tr> <tr> <td headers="signifer">Second Row</td> <td headers="meaning">No</td> </tr> </pre>	<pre> tr:nth-child(even) { background: #CCC; } tr:nth-child(odd) { background: #FFF; } </pre>

Slide 50 - Consolidate your legacy site's existing font sizes

TIP: Consolidate your legacy site's existing font sizes

Point	Pixel	Em	Percent	Keyword
7.5pt	10px	0.625em	62.5%	x-small
10pt	13px	0.8em	80%	small
12pt	16px	1em	100%	medium
13.5pt	18px	1.125em	112.5%	large
18pt	24px	1.5em	150%	x-large
24pt	32px	2em	200%	xx-large

Use percent on the body element, and then use the Em unit to size it from there. Seen as a standard in design.

Over time a large site with multiple designers will collect a large number of distinct but similar font sizes.

By consolidating similar styles into a smaller number of styles, your site will look more consistent.

These 30 text sizes could be consolidated into 6 test styles.

John Slatin AccessU 2022 Powered by Knowbility

Over time a large site with multiple designers will collect a large number of distinct but similar font sizes.

By consolidating similar styles into a smaller number of styles, your site will look more consistent.

IMAGE: A table roughly comparing the various measurement units used in HTML and CSS to determine the size of the text.

These 30 text sizes could be consolidated into 6 test styles.

Refer to

- <https://websemantics.uk/tools/font-size-conversion-pixel-point-em-rem-percent/>
- <https://weboverhauls.github.io/conversion-table/>

Example of Consolidated Font Sizes

Instead of 22 distinct font sizes, this CSS file specifies 7.

```
.font-tag-size-1,
.keyword-x-small,
.px10 {
  font-size: .635REM;
}

.font-tag-size-2,
.keyword-small,
.px13,
{
  font-size: .8REM;
}

.font-tag-size-3,
.keyword-medium,
.per100, .em1, .rem1,
.px16, .pt12 {
  font-size: 1REM;
}

.font-tag-size-4,
.keyword-large,
.px18 {
  font-size: 1.125REM;
}

.font-tag-size-5,
.keyword-x-large,
.px24 {
  font-size: 1.5REM;
}

.fonttag-size-6,
.keyword-xx-large,
.px32 {
  font-size: 2REM;
}

.fonttag-size-7,
.px48 {
  font-size: 3REM;
}
```

John Slatin AccessU 2022 Powered by Knowbility

Instead of 21 distinct font sizes, this CSS file specifies 7.

```
.font-tag-size-1,
.keyword-x-small,
.px10 {
  font-size: .635REM;
}

.font-tag-size-2,
.keyword-small,
.px13,
{
  font-size: .8REM;
}

.font-tag-size-3,
.keyword-medium,
.px16,
.pt10 {
  font-size: 1REM;
}

.font-tag-size-4,
.keyword-large,
.px18 {
  font-size: 1.125REM;
}

.font-tag-size-5,
.keyword-x-large,
.px24 {
  font-size: 1.5REM;
}

.fonttag-size-6,
.keyword-xx-large,
.px32 {
  font-size: 2REM;
}

.fonttag-size-7,
.px48 {
  font-size: 3EM;
}

BIG {
  font-size: larger; /* Sets the font-size to a
larger size than the parent element */
}

SMALL {
  font-size: smaller; /* Sets the font-size to a
smaller size than the parent element */
}
```

View Source Before and After

Before Consolidation	After Consolidation
<pre><p>This little pig went to market,
 This little pig stayed home,
 This little pig had roast beef,
 And this little pig had none,
 And this little pig went wee-wee-wee
 All the way home.</p></pre>	<pre><p>This little pig went to market,
 This little pig stayed home,
 This little pig had roast beef,
 And this little pig had none,
 And this little pig went wee-wee-wee
 All the way home.</p></pre>

John Slatin AccessU 2022 Powered by Knowbility

Before Consolidation

```
<p><font size="12pt">This little pig went to market,</font><br />
<font size="16px">This little pig stayed home,</font><br />
<font size="1em">This little pig had roast beef,</font><br />
<font size="100%">And this little pig had none,</font><br />
<font size="medium">And this little pig went wee-wee-wee</font><br />
<font size="1rem">All the way home.</font></p>
```

After Consolidation

```
<p><span class="pt12">This little pig went to market,</span><br />
<span class="px16">This little pig stayed home,</span><br />
<span class="em1">This little pig had roast beef,</span><br />
<span class="cent100">And this little pig had none,<br />
<span class="keyword-medium">And this little pig went wee-wee-wee</span><br />
<span class="rem1">All the way home.</span></p>
```

Slide 53 - Does it make a difference?

Does it make a difference?

Legacy Site Before Consolidation

This little pig went to market,
This little pig stayed home,
And this little pig had none,
And this little pig went wee-wee-wee
All the way home.

Legacy Site After Consolidation

This little pig went to market,
This little pig stayed home,
This little pig had roast beef,
And this little pig had none,
And this little pig went wee-wee-wee
All the way home.

John Slatin AccessU 2022 Powered by Knowbility

IMAGES: The nursery rhyme shown before and after consolidations

Legacy Site Before Consolidation

The six lines have noticeably different sized letters.

Legacy Site After Consolidation

All six lines have the exact same letter size, even though the names of the styles are different.

.

Tip: Graphic Headers

```
<h1 style="background: url('h1-introduction.png')
no-repeat; height: 200px; width: 600px;">
  <span
    style="text-indent: -5000px">Introduction</span>
</h1>
```

John Slatin AccessU 2022 Powered by Knowbility

```
<h1 style="background: url('h1-introduction.png') no-repeat; height:
200px; width: 600px;">
  <span style="text-indent: -5000px">Introduction</span>
</h1>
```

This is one available technique. I'm partial to it. But it is not the definitive solution. There are other options out there.

Slide 55 - The Non-breaking Space Markup (“ ”)

The Non-breaking Space Markup (“ ”)

`<p>First paragraph.</p>` First paragraph.

`<p> </p>`

`<p>Second paragraph.</p>` Second paragraph.

JAWS encountering “ ”

John Slatin AccessU 2022 Powered by Knowbility

The following HTML will result in an extra large space between the first and second paragraph.

When the JAWS screen reader encounters the non-breaking space markup (“ ”), it announces the word “Blank.”

Slide 56 - Hiding Content from Screen Readers

Hiding Content From Screen Readers

- JAWS will read `<p> </p>` aloud as the word "blank."
- NVDA will ignore it.
- ChromeVox will ignore it.

Wrong

`<p> </p>`

Right

`<p aria-hidden="true"> </p>`

Problem with the non-breaking space

- JAWS will read the non-breaking space aloud as the word "blank."
- NVDA will ignore it.
- ChromeVox will ignore it.

Solution: Hide the non-breaking space using the aria-hidden attribute.

```
<p aria-hidden="true">non-breaking space</p>
```

AUDIO clip: When JAWS encounters a stand alone instance of “ ” it announces “BLANK”.

User / Custom Style Sheet

User or custom style sheets, allow users to apply their own CSS to web sites they visit in browsers. This gives users control of font sizes, colors, type faces, spacing, etc.,

Microsoft Edge doesn't seem to support user or custom style sheets.

For instructions for adding this functionality to **Firefox** or **Chrome** read the following article:

What Is a User Style Sheet?
By Jennifer Kyrnin
Updated on January 26, 2021
<https://www.thoughtco.com/user-style-sheet-3469931>

John Slatin AccessU 2022 Powered by Knowbility

User or custom style sheets, allow users (including PWD) to apply their own CSS to web sites they visit in browsers. This gives users control of font sizes, colors, typefaces, spacing, etc.,

Microsoft Edge doesn't seem to support user or custom style sheets.

For instructions for adding the functionality to Firefox or Chrome read the following article:

What Is a User Style Sheet?

By Jennifer Kyrnin

Updated on January 26, 2021

<https://www.thoughtco.com/user-style-sheet-3469931>

Warning Against the !important Property

- Tech savvy PWDs might use custom style sheets with their browsers.
- For their sake, you will want to avoid using the **!important** property from over-riding their custom style sheets.
- Avoid using it except in alternative style sheets.

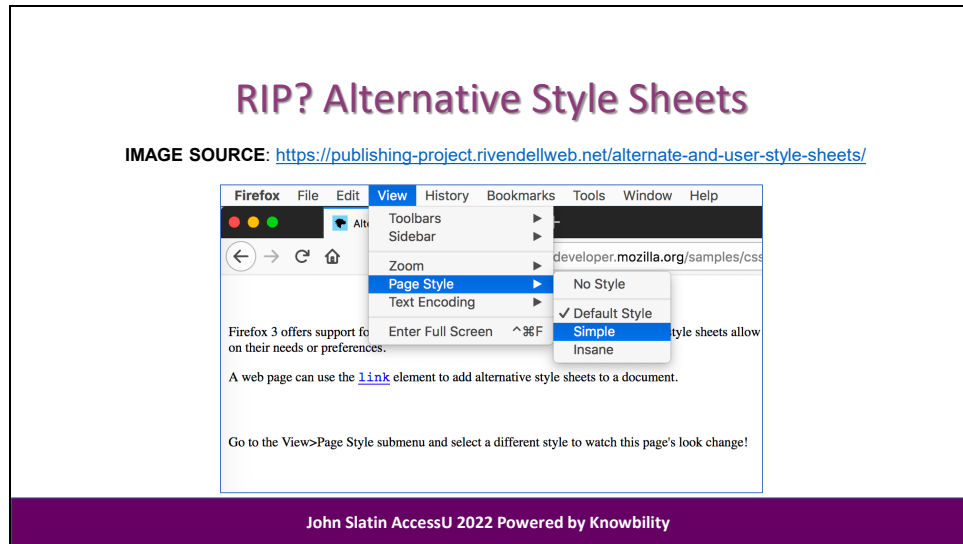
```
html {  
  font-size: 10px !important;  
  font-family: "curlz mt", fantasy !important;  
  color: blue !important;  
  background-color: DodgerBlue !important;  
}
```

John Slatin AccessU 2022 Powered by Knowbility

- Some tech savvy PWDs might use custom style sheets with their browsers.
- For their sake, you will want to avoid using the **!important** property from over-riding their custom style sheets.
- Avoid using it except in alternative style sheets.

```
html {  
  font-size: 10px !important;  
  font-family: "curlz mt", fantasy !important;  
  color: blue !important;  
  background-color: DodgerBlue !important;  
}
```

Slide 59 - RIP? Alternative Style Sheets



None of the major browsers support Alternative Style Sheets.

IMAGE: Screen capture showing how easy it used to be to change style sheets using the user interface of certain browsers.

IMAGE SOURCE: <https://superuser.com/questions/594358/modify-chrome-user-agent-stylesheet>

"Google decided to drop user stylesheets altogether even though they have been an essential part of web-browsers since practically the dawn of time (I started using it in IE4 or 5)."

Synetech

Mar 27, 2014 at 5:01

Some Possible Alternative Style Sheets

- Check accessibility (E.G., <http://ffoodd.github.io/a11y.css/>)
- Hide images
- Fonts for Dyslexia (E.G., OpenDyslexic CSS - <https://gist.github.com/mvilrocx/4536584>)
- Larger Fonts
- Print Preview
- Grayscale
- Alternative color schemes for the design period
- No Layout
- Lynx.css (Text only browser emulator) - <https://jshawl.github.io/lynx/>

John Slatin AccessU 2022 Powered by Knowbility

- Check accessibility (E.G., <http://ffoodd.github.io/a11y.css/>)
- Hide images
- Fonts for Dyslexia (E.G., OpenDyslexic CSS - <https://gist.github.com/mvilrocx/4536584>)
- Larger Fonts
- Print Preview
- Grayscale
- Alternative color schemes for the design period
- No Layout
- Lynx.css (Text only browser emulator) - <https://jshawl.github.io/lynx/>

Slide 60 - You can do it, but should you?

You can do it, but should you?

PROS	CONS
<ul style="list-style-type: none">• Empowers the user• Allows for a variety of neurodiverse experiences. (The benefit of the style sheets will vary by user.)• Can allow for greater collaboration between web designers and PWD	<ul style="list-style-type: none">• Assumes the users to be super users• Assumes the user to know that alternative styles are available on your site• Assumes the user has a typeface preferences• No longer supported by major browsers• Requires programming for it to work

John Slatin AccessU 2022 Powered by Knowbility

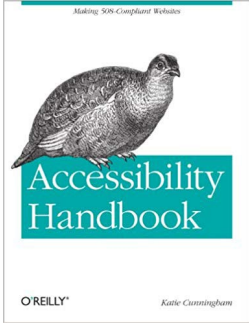
PROS

- Empowers the user
- Allows for a variety of neurodiverse experiences. (The benefit of the style sheets will vary by user.)
- Can allow for greater collaboration between web designers and PWD

CONS

- Assumes the users to be super users
- Assumes the user to know that alternative styles are available on your site
- Assumes the user to have typeface preferences
- Not supported by major browsers
- Requires programming for it to work
 - Change the selected CSS in the DOM
 - Add cookie to make the change “sticky”

Additional Resources



Accessibility Handbook
by Katie Cunningham

Paperback: 100 pages
Publisher: O'Reilly Media; 1 edition (September 14, 2012)
ISBN-10: 1449322859
ISBN-13: 978-1449322854
\$19.34

Concise and thorough

John Slatin AccessU 2022 Powered by Knowbility

Paperback: 100 pages
Publisher: O'Reilly Media; 1 edition (September 14, 2012)
ISBN-10: 1449322859
ISBN-13: 978-1449322854
\$19.34

Concise and thorough

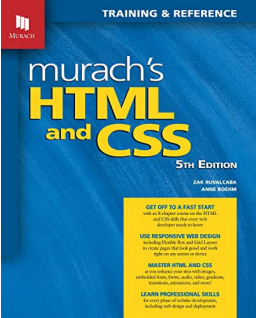
Slide 63 - *Universal Design for Web Applications: Web Applications That Reach Everyone* 1st Edition by Wendy Chisholm and Matt May



Paperback: 198 pages
Publisher: O'Reilly Media; 1 edition (November 24, 2008)
Language: English
ISBN-10: 0596518730
ISBN-13: 978-0596518738
\$21.17

Slide 64 - *Murach's HTML and CSS (5th Edition)* by Anne Boehm and Zak Ruvalcaba

Murach's HTML and CSS (5th Edition) by Anne Boehm and Zak Ruvalcaba



This the most recent edition. It replaces *Murach's HTML5 and CSS3*

Publisher : Mike Murach & Associates; 5th edition (December 14, 2021)
Paperback : 602 pages
ISBN-10 : 1943872864
ISBN-13 : 978-1943872862

I like the format of this book series. It embraces a variety of learning styles.

John Slatin AccessU 2022 Powered by Knowbility

This the most recent edition. It replaces Murach's HTML5 and CSS3

Publisher : Mike Murach & Associates; 5th edition (December 14, 2021)

Paperback : 602 pages

ISBN-10 : 1943872864

ISBN-13 : 978-1943872862

I like the format of this book series. It embraces a variety of learning styles.

Slide 65 - When shopping for books on HTML and CSS

When shopping for books on HTML and CSS

HTML, CSS, and web browsers are constantly being updated.

All other things being equal, choose the most recently published book.

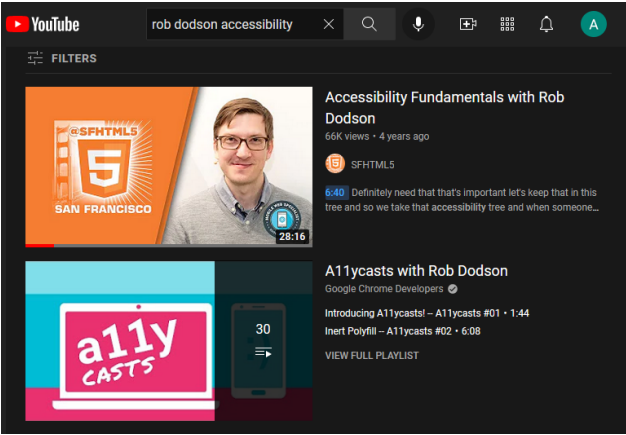
John Slatin AccessU 2022 Powered by Knowbility

HTML, CSS, and web browsers are constantly being updated.

All other things being equal, choose the most recently published book.

Slide 66 - Rob Dodson's Accessibility Videos on YouTube

URL: <https://youtu.be/HtTyRajRuyY>



The screenshot shows a YouTube search results page for the query 'rob dodson accessibility'. The top navigation bar includes the YouTube logo, the search term, and various utility icons. Below the search bar, there are two video results. The first result is 'Accessibility Fundamentals with Rob Dodson' by SFHTML5, featuring a video thumbnail with Rob Dodson and a duration of 28:16. The second result is 'A11ycasts with Rob Dodson' by Google Chrome Developers, featuring a video thumbnail with 'a11y CASTS' and a duration of 30. A purple footer bar at the bottom of the slide contains the text 'John Slatin AccessU 2022 Powered by Knowbility'.

YouTube rob dodson accessibility

FILTERS

Accessibility Fundamentals with Rob Dodson
66K views · 4 years ago
SFHTML5
6:40 Definitely need that that's important lets keep that in this tree and so we take that accessibility tree and when someone...

A11ycasts with Rob Dodson
Google Chrome Developers
Introducing A11ycasts! - A11ycasts #01 - 1:44
Inert Polyfill - A11ycasts #02 - 6:08
VIEW FULL PLAYLIST

John Slatin AccessU 2022 Powered by Knowbility

URL: <https://youtu.be/HtTyRajRuyY>

Slide 67 - Kevin Powell's CSS Videos on YouTube

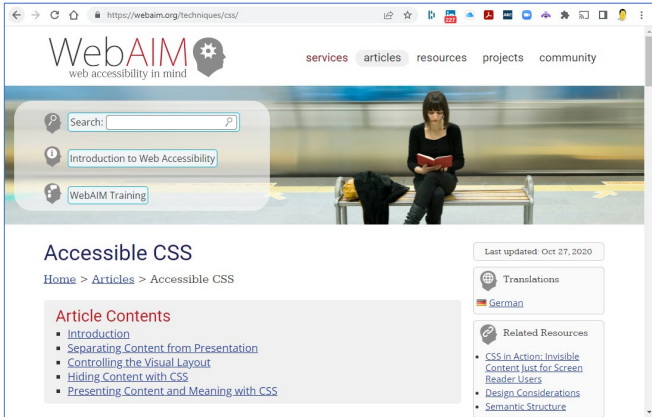
URL: <https://www.youtube.com/channel/UCJZv4d5rbIKd4QHMPkcABCw>

John Slatin AccessU 2022 Powered by Knowbility

URL: <https://www.youtube.com/channel/UCJZv4d5rbIKd4QHMPkcABCw>

Slide 68 - "Accessible CSS" from WebAIM

URL: <https://webaim.org/techniques/css/>



The screenshot shows a web browser displaying the WebAIM website. The URL in the address bar is <https://webaim.org/techniques/css/>. The page features the WebAIM logo (web accessibility in mind) and a navigation menu with links for services, articles, resources, projects, and community. A search bar is visible, along with a featured article titled "Introduction to Web Accessibility" and a "WebAIM Training" button. The main content area is titled "Accessible CSS" and includes a breadcrumb trail: Home > Articles > Accessible CSS. Below this, there is an "Article Contents" section with a list of links: Introduction, Separating Content from Presentation, Controlling the Visual Layout, Hiding Content with CSS, and Presenting Content and Meaning with CSS. On the right side, there is a "Last updated" date (Oct 27, 2020), a "Translations" section with a "German" link, and a "Related Resources" section with links to "CSS in Action: Invisible Content Just for Screen Reader Users", "Design Considerations", and "Semantic Structure".

John Slatin AccessU 2022 Powered by Knowbility

URL: <https://webaim.org/techniques/css/>

Some Upcoming Sessions

Slide 70 - How to HTML

1:45 PM
to 3:15 PM
Wed, May 11

Track: **Development/Coding/Testing**
Live: **How to HTML**

taught by: **Amy Carney**

Trustee Room 104
sponsored by Code Mantra

HTML (HyperText Markup Language) is critical for creating accessible websites. Find out why HTML is important and what elements are available to make your site make sense to everyone.

Recognize and implement 50 semantic HTML elements.

John Slatin AccessU 2022 Powered by Knowbility

Wednesday, May 11, 2022

1:45 P.M. to 3:15 P.M.

Session Summary

HTML (HyperText Markup Language) is critical for creating accessible websites. Find out why HTML is important and what elements are available to make your site make sense to everyone.

Description

We've all heard that HTML plays a critical part in web accessibility. But how does one get started learning and building that HTML knowledge when there are 200+ elements available? Whether you're a designer, developer, or tester, this course is for you! We'll discuss why HTML is accessible. Then we'll dive right into learning which HTML elements are available and how to use those elements.

Practical Skills

- Understand how HTML makes or breaks the accessibility of a webpage.
- Plan out the architecture of a web document.
- *Recognize and implement 50 semantic HTML elements.*

Slide 71 - Web Accessibility is Broken. It's Time to Fix it.

The slide features a purple header bar at the top with the text "John Slatin AccessU 2022 Powered by Knowbility". The main content area is white and contains the following text:

9:45 AM Q&A: Web Accessibility is Broken. It's Time to Fix It. Trustee Room 104
to 10:30 AM taught by: [Eric Eggert](#) sponsored by Code Mantra
Thu, May 12

This is not a full presentation. This is a Q&A for the pre-recorded session available to AccessU attendees in our Learning Center.

The web is still an accessibility wasteland. It must not be one. However, systematic issues plague progress that would enable millions of disabled people to properly participate as netizens.

Thursday, May 12, 2022
9:45 A.M. to 10:30 A.M.

Session Summary

The web is still an accessibility wasteland. It must not be one. However, systematic issues plague progress that would enable millions of disabled people to properly participate as netizens.

Description

This is not a full presentation. This is a Q&A for the pre-recorded session available to AccessU attendees in our Learning Center.

If we want the web to be a better, more accessible place, we have to align the easy, cost-effective decision with the correct, accessible decision. As long as our guidance is complicated, our tools hard to use, and our standards imprecise, even people who want an accessible outcome feel that they have to work “against” the accessibility requirements.

That is not how it should be. Making an accessible decision should feel the default, natural. Let’s create a world where we support people who want to do the right thing and not penalize them for not knowing the details of how a screen reader works.

Learn how we got to the web we have right now. Learn what we can learn from the roads taken and from the paths we left unexplored. Understand why the web’s accessibility is an entangled network with various constituents. And then, find out how we all can untangle this network to make the vision of the web a reality: This is for everyone.

Practical Skills

- Understand the history of web accessibility.
- Learn why accessibility on the web is the way it is.
- Get into the mindset of how to fix web accessibility. Systematically.

Slide 72 - A Tangled Web of ARIA

10:45 AM
to 11:30 AM
Thu, May 12

Track: **Development/Coding/Testing**
Q&A: A Tangled Web of ARIA
taught by: [Mark Steadman](#)

Trustee Room 104
sponsored by Code Mantra

This is not a full presentation. This is a Q&A for the pre-recorded session available to AccessU attendees in our Learning Center.

Web development has seen an alarming trend of using ARIA-first development instead of using the power of Semantic HTML, which has led to components that are a complete tangled web of ARIA that is hard to unwind. In this session, will walk through a scenario of a development team creating a component from scratch, the different iterations of that component, and how using ARIA only can create the tangled web of difficult-to-read HTML code that is also inaccessible.

- Accessibility benefits of semantic HTML.

John Slatin AccessU 2022 Powered by Knowbility

Thursday, May 12, 2022
10:45 A.M. to 11:30 A.M.

Session Summary

Web development has seen an alarming trend of using ARIA-first development instead of using the power of Semantic HTML, which has led to components that are a complete tangled web of ARIA that is hard to unwind. In this session, will walk through a scenario of a development team creating a component from scratch, the different iterations of that component, and how using ARIA only can create the tangled web of difficult-to-read HTML code that is also inaccessible.

Description

This is not a full presentation. This is a Q&A for the pre-recorded session available to AccessU attendees in our Learning Center.

In this day and age of web development is done using mostly JavaScript frameworks, semantic HTML has been left behind. In its place we see more developers using ARIA (Accessible Rich Internet Applications) to fill in the gaps and make their content accessible.

However, a scary trend that has been seen lately is that development teams have solely relied on ARIA for making content accessible. What is left is a giant mess of ARIA that is either incorrect, difficult to understand, or leaves developers wondering what they have to do to make it accessible again.

In this talk, we will walk through a scenario of a development team creating a component from scratch, the different iterations of that component, and how using ARIA only can create the tangled web of difficult-to-read HTML code that is also inaccessible.

(ICT 2022 Presenter)

Practical Skills

- *Accessibility benefits of semantic HTML.*
- The dangers of ARIA-first web development.
- How to create accessible coding standards.

Slide 73 - What's in a Name? Why and How to Add Accessible Name to User Interface Elements.

2:30 PM
to 3:15 PM
Thu, May 12

Track: **Development/Coding/Testing**
Q&A: **What's in a Name? Why and How to Add Accessible Names to User Interface Elements**
taught by: [Alicia Evans](#)

Trustee Room 104
sponsored by Code Mantra

This is not a full presentation. This is a Q&A for the pre-recorded session available to AccessU attendees in our Learning Center.

In this lesson, you'll learn why it's crucial to ensure user interface elements have accessible names, how to use developer tools to find the accessible name of an element, various techniques for implementing accessible names, and how those names should be crafted.

- Techniques for adding an accessible name through HTML, CSS, and ARIA.
- WCAG violations and best practices associated with accessible names.

John Slatin AccessU 2022 Powered by Knowbility

Thursday, May 12, 2022
2:30 P.M. to 3:15 P.M.

Session Summary

In this lesson, you'll learn why it's crucial to ensure user interface elements have accessible names, how to use developer tools to find the accessible name of an element, various techniques for implementing accessible names, and how those names should be crafted.

Description

This is not a full presentation. This is a Q&A for the pre-recorded session available to AccessU attendees in our Learning Center.

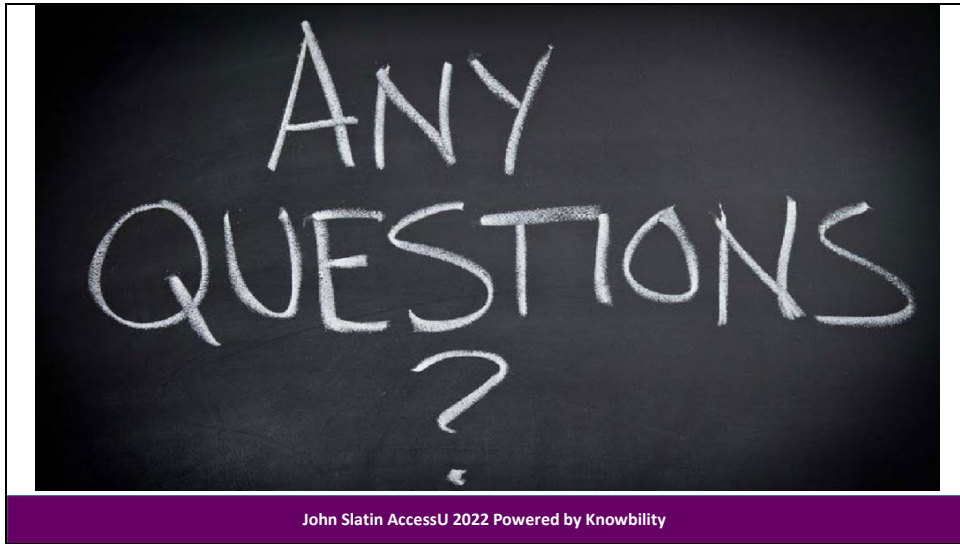
Assistive technologies use the accessible name to identify User Interface (UI) elements to users. But the web is riddled with nameless UI elements, causing uncertainty and confusion. As developers, it's important to know why the accessible name is important, how it differs from a visual label, and how to provide an accessible name.

In this technical lesson, you'll learn:

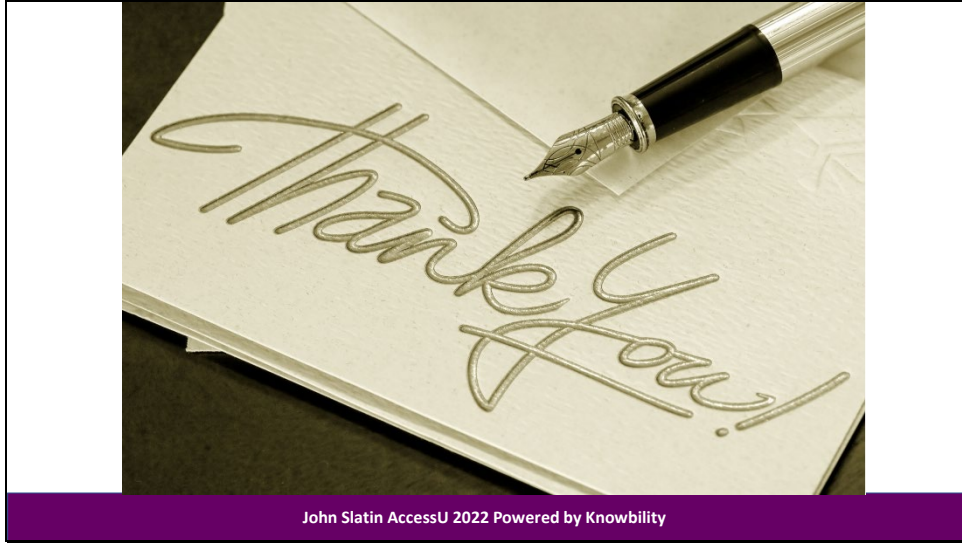
- What the accessible name is and why it is important.
- How to find the accessible name in Chrome's developer tools.
- Techniques for adding an accessible name through HTML, CSS, and ARIA.
- WCAG violations and best practices associated with accessible names.

- Practical Skills
- What is in an accessible name?
- Where do you find it, and how is it computed?
- Techniques for crafting and assigning an accessible name.

Slide 74 - Any Questions?



Slide 75 - Thank You. This concludes this session.



John Slatin AccessU 2022 Powered by Knowbility