

HTML5 – The basics



What is HTML?

- HTML stands for Hypertext Markup Language
- Markup languages are not the same as programming languages, they use **tags** to annotate documents.
- Tags indicate where headings, images, lists, links, line breaks, etc should go.



.html files

- Your computer “knows” to open an .html file in a browser.
- The browser can read this file (and others) and know how to render it.
- Screen readers/assistive devices utilize the HTML tags to present the information as well.



Web Standards

- No one “runs” the Internet or the Web, some groups do take proactive roles:
 - Internet Engineering Task Force (IETF)
 - World Wide Web Consortium (W3C)
 - The Web Accessibility Initiative (WAI)



Evolution of HTML

| Year | Advancement |
|------|---|
| 1993 | HTML 1.0-Developed by Tim Berners-Lee to link documents |
| 1995 | HTML 2.0 - Developed by Internet Engineering Task Force RFC to include stylized text and tables |
| 1996 | CSS 1 |
| 1997 | HTML 3.2–Developed by W3C and had browser specific features |
| 1997 | HTML 4.0–A move back to normalizing across platforms. |
| 1998 | CSS 2 |
| 1999 | HTML 4.01 – Introduced different document types |



Where we are now

- Established Guidelines
 - New features should be based on HTML, CSS, the DOM, and JavaScript
 - Reduce the need for external plugins (e.g. Flash)
 - More markup to replace scripting
 - HTML5 should be device independent

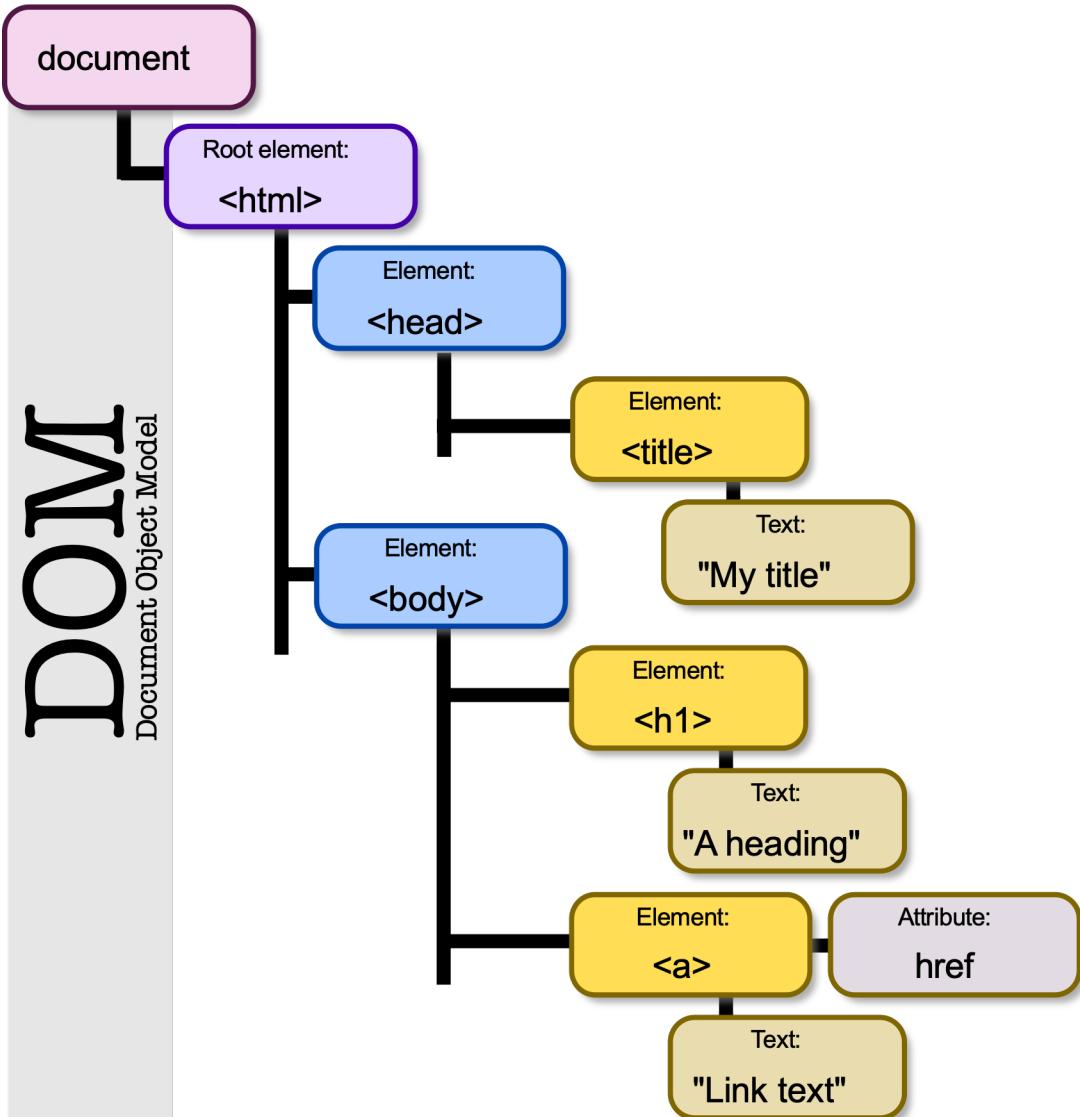


The Document Object Model (DOM)

- Basis of HTML5 is “*New features should be based on HTML, CSS, the DOM, and JavaScript...*”
- DOM provides common tree-like structure that all pages should follow
- Computer Scientists love trees (the mathematical kind) because you can test them.



HTML is built on the DOM



The `html` tag is the root and has two children - the `head` and the `body`.

[Image Source](#)



Three parts of a well-formed document

- Doctype
 - Version of HTML that you will be using
- Head
 - Metadata
- Body
 - Displayable content



Doctype

- HTML5
 - <!DOCTYPE html>
- Previous versions dictated backwards compatibility
 - <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd">
 - <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">



Head

- Additional information used by the browser
 - Meta data – language, title
 - Supporting files – JavaScript, Styling, Add-ons
- Typically, other than title and favicon the meta-data is not displayed

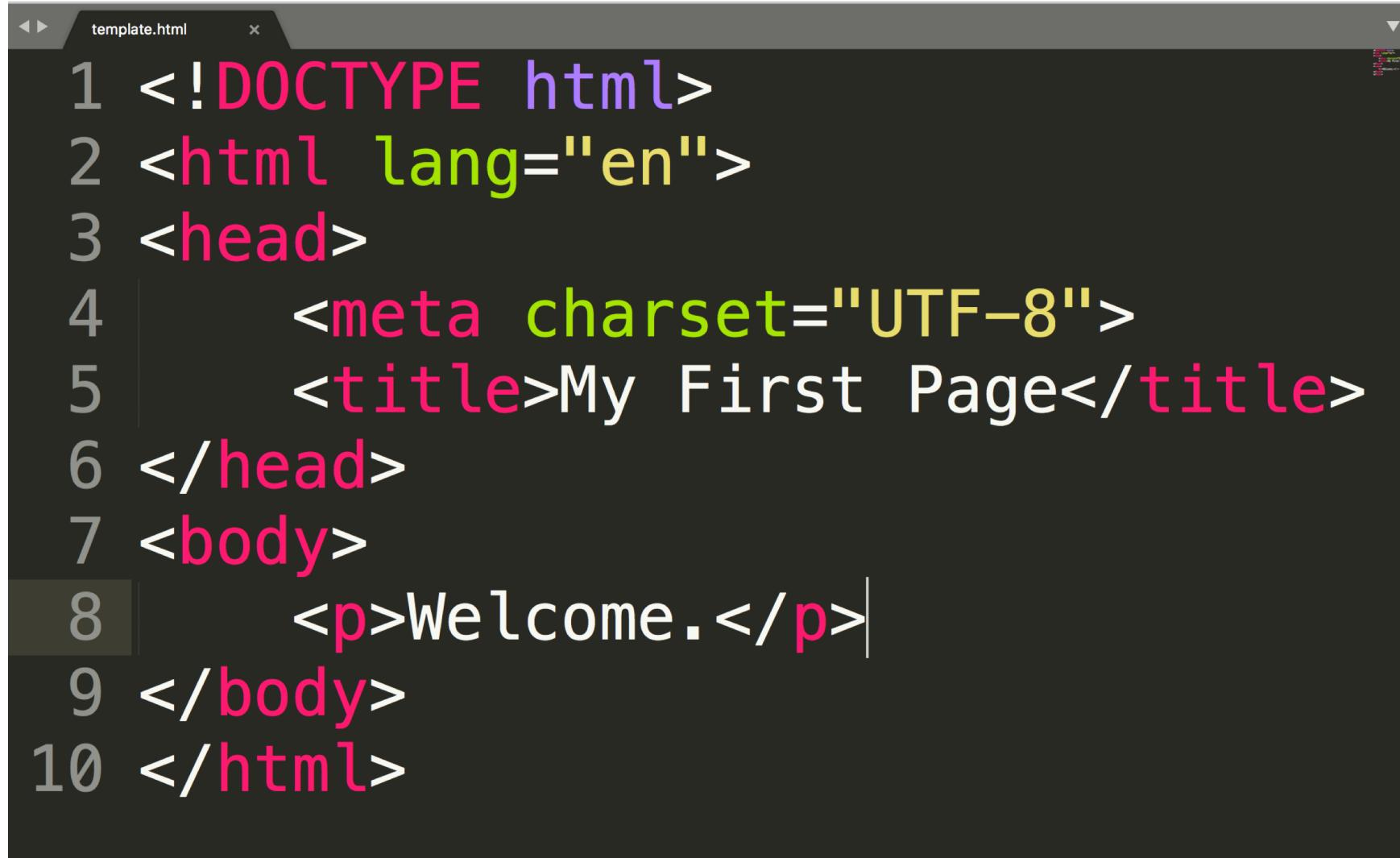


Body

- Bulk of your page
- Important to write well-formatted (tree-like) code.
- Most of the content is displayed by the browser, but there may be some meta-data too!!



Example



A screenshot of a code editor window titled "template.html". The code is an HTML document with the following structure:

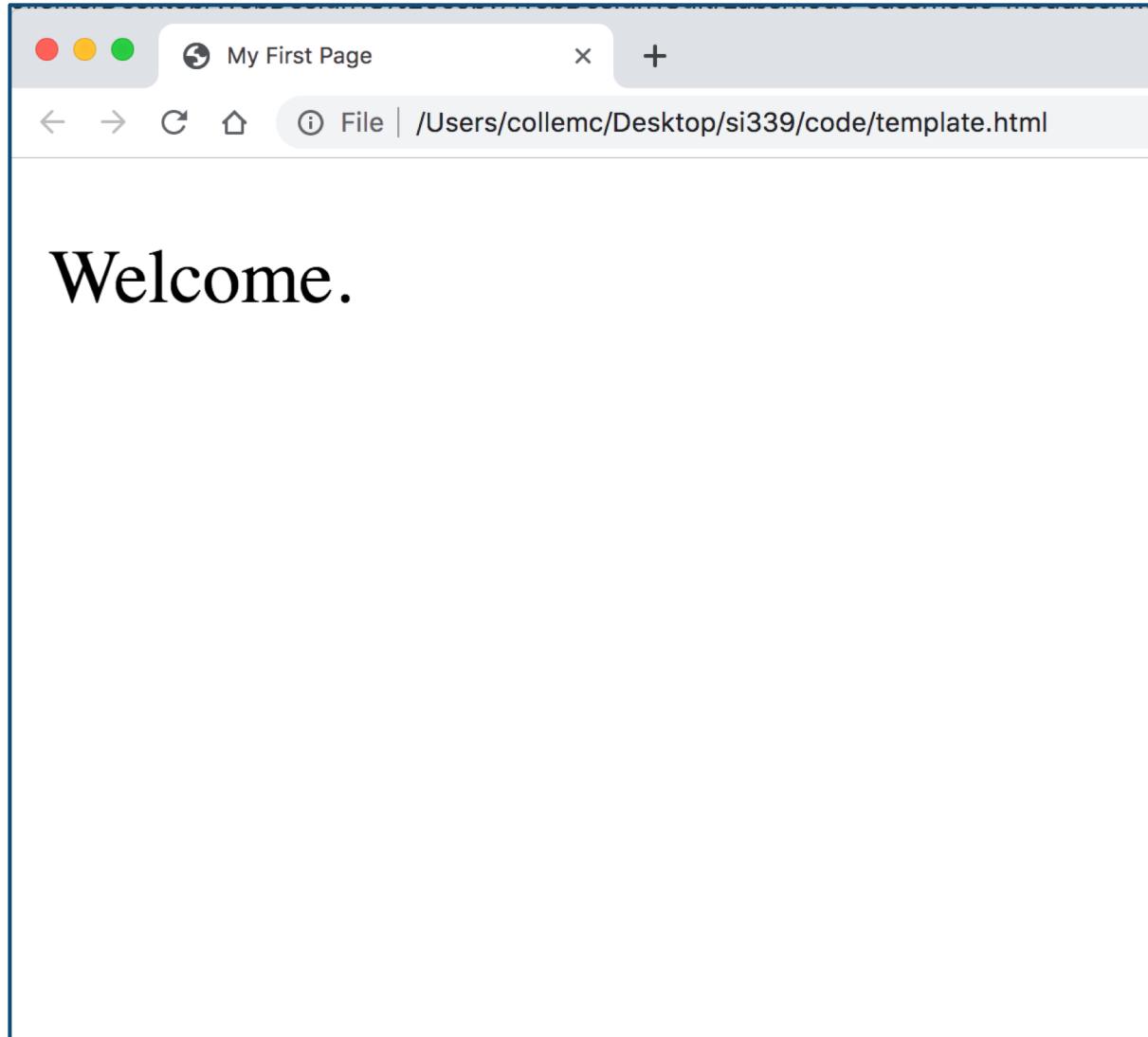
```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4     <meta charset="UTF-8">
5     <title>My First Page</title>
6 </head>
7 <body>
8     <p>Welcome.</p>
9 </body>
10 </html>
```

The code is color-coded: purple for the DOCTYPE, blue for the html tag, green for the head tag, yellow for the meta tag, red for the title tag, orange for the body tag, pink for the p tag, and black for the text content.

The code in this image can be found in the alt text.



Example Results



Can you find the title?

Get in the habit of checking the URL.

Why is my text “bigger”?



Validate the Code

- The browser will “hide” mistakes so always validate.
 - validator.w3.org
 - wave.webaim.org
 - [aXe extension](#)



validator.w3.org

Ready to check

Checker Input

Show source outline image report

Options...

address

file upload

Check by text input CSS

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>My First Page</title>
</head>
<body>
    <p>Welcome.</p>
</body>
</html>
```

Document checking completed. No errors or warnings to show.

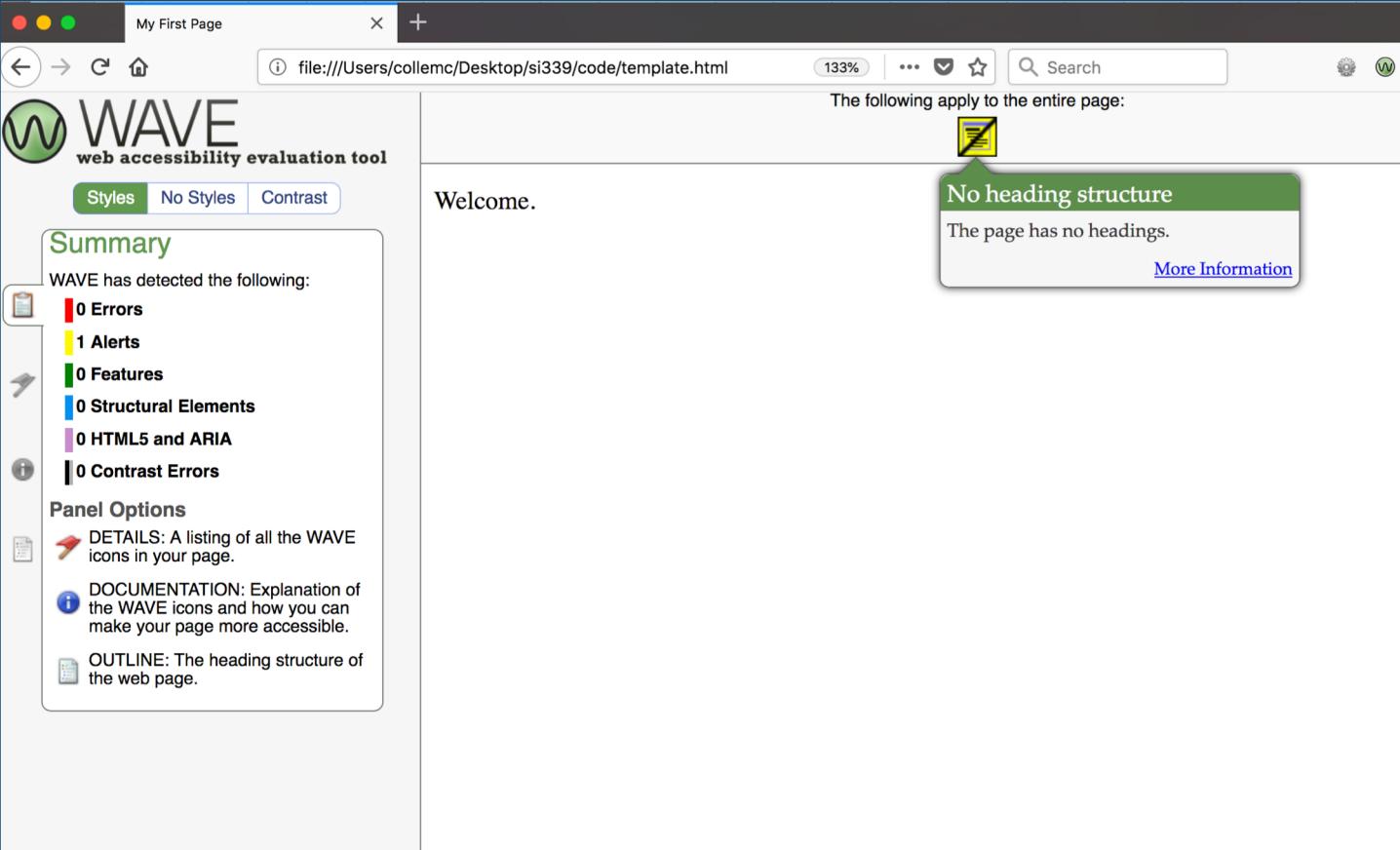
Source

1. <!DOCTYPE html>↔
2. <html lang="en">↔
3. <head>↔
4. <meta charset="UTF-8">↔
5. <title>My First Page</title>↔
6. </head>↔
7. <body>↔
8. <p>Welcome.</p>↔
9. </body>↔
10. </html>

Warnings can be
ignored, not errors!



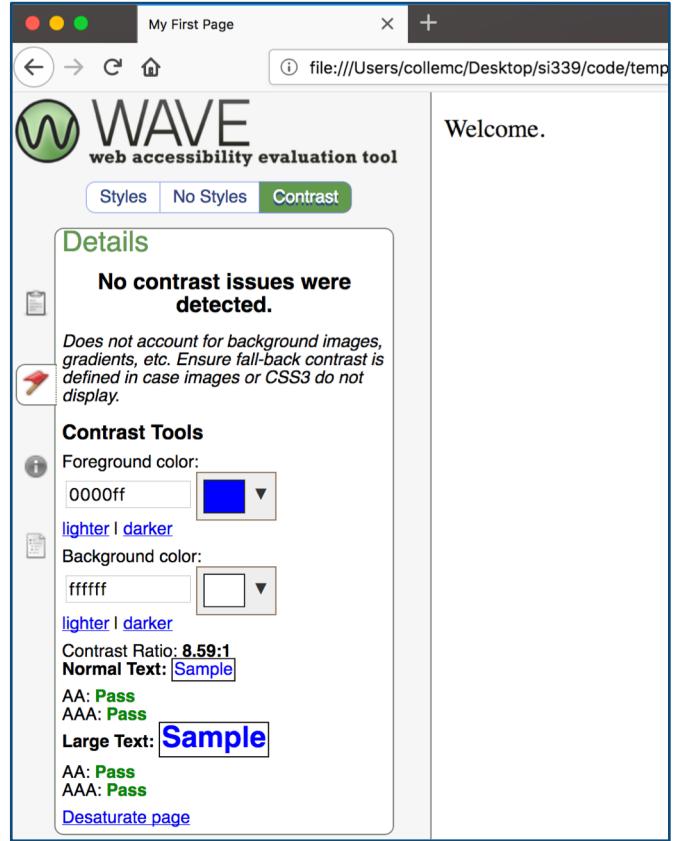
wave.webaim.org



The screenshot shows the WAVE web accessibility evaluation tool interface. On the left, there's a sidebar with a summary of findings: 0 Errors, 1 Alert, 0 Features, 0 Structural Elements, 0 HTML5 and ARIA, and 0 Contrast Errors. Below this are panel options for Details, Documentation, and Outline. The main content area displays the text "Welcome." and a yellow icon representing an alert. A tooltip box is overlaid on the icon, stating "No heading structure" and "The page has no headings." with a "More Information" link.

- Check for Errors in the Styles

- Check for contrast issues



The screenshot shows the WAVE tool with the "Contrast" tab selected. It displays a message: "No contrast issues were detected." Below this, it says "Does not account for background images, gradients, etc. Ensure fall-back contrast is defined in case images or CSS3 do not display." On the right, there are contrast tools for foreground and background colors, both set to black. It also shows contrast ratios: "Contrast Ratio: 8.59:1", "Normal Text: Sample", "AA: Pass", "AAA: Pass", and "Large Text: Sample". At the bottom, there are links for "AA: Pass", "AAA: Pass", and "Desaturate page".



aXe

axe
v3.9.0 ([axe-core 3.3.1](#))

All issues found 3 ▾ [Run again](#)

| | |
|---|---|
| Document must have one main landmark | 1 |
| Page must contain a level-one heading | 1 |
| All page content must be contained by landmarks | 1 |

All of these issues should be addressed!



HTML tags

- I can't teach you all of the tags
- I can't teach you all of the tags
- You don't want me to teach you all of the tags



Finally, some tags...

- Tags have a beginning and an end



- Some tags have *attributes* (src, href, etc..)



Common Tags

- Headings (<h1> </h1> ... <h6> </h6>)
 - These tags have **syntax** and **semantics**
- Paragraphs (<p> </p>)
 - Should only contain inline elements
- Divs (<div> </div>)
 - Generic section that is larger than a paragraph



Lists

- Ordered lists (` `)
- Unordered lists (` `)
- Ensure every “child” is ``
- Indentation isn’t necessary, but helpful

```
<ul>
    <li> Item One </li>
    <li> Item Two </li>
</ul>
```



Attributes

- Attributes provide additional information about an element
- Always specified in the **start tag**
- Attributes come in name/value pairs *unless they are Boolean.*



Images

```

```

- src and alt attributes are always required!!
- alt attribute (can) be empty, but not in your autograded assignments



Notes on Images

- Images rarely work the first time
 - Show a broken link, too big, too small, etc.
- Save yourself heartache and size/carefully name your picture before you use it.



Note on Attributes

- class – used to group HTML elements
- id – used to identify specific element
- style – specifies a certain visual style
(do not use this!!!)
- accesskey – a shortcut key to activate an element
- tabindex – the order that the tab button brings elements into focus



Links

- Links are what make the Web a web.
- The interlinked nature of the web leads to the “knowledge” that search engines appear to have.
- Links can be internal, external, absolute, relative, and graphical



Anchor links

- The <a> tag stands for “anchor”
- Links have one attribute within the tag AND need additional content
 - Attribute: reference to location of new content
 - Content is the “clickable” part, may be text or image.



Absolute reference

```
<a href="http://www.cnn.com/">CNN</a>
```



Relative References

```
<a href="menu.html">Menu</a>
```

```
<a href="docs/menu.html">Menu</a>
```

```
<a href="#menu">Menu</a>
```



Absolute vs Relative

- When would you use absolute links?
- Are there any benefits to using local links?
- Your links should NEVER have folders that are specific to your computer



Using Images as the Link

- The “clickable” component doesn’t have to be text.

```
<a href="http://www.linkedin.com">  
  </a>
```

```
<a href="http://www.linkedin.com"><i class="fab  
fa-linkedin-in" aria-label = "LinkedIn"></i></a>
```



Usability Issues

- Make sure the clickable component has an informative name
- Information in the images should be available to those who can't see the image using **alt text** or an **aria-label** attribute



Special Entities

- Tags always start with a bracket (<)

| If you want.... | Then use... |
|-----------------|-------------|
| < | < |
| > | > |
| © | © |
| blank space | |
| ¢ | ¢ |
| & | & |



Using Semantic Tags

- In the beginning (insert dramatic music of your choice...) there was **<div>**
- Divs almost always had special classes/ids associated with them
 - `<div class = "header">...</div>`
 - `<div class = "navigation">...</div>`
 - `<div class = "footer">...</div>`



<header>

- A group of introductory or navigational aids: title, navigation links
- Not to be confused with <head> or the different headings.



<nav>

- A section of the page that links to other pages or to parts within the page.
 - Often found in the <header> tag
 - Often contains a tag
- Use instead of <div role="navigation">



<footer>

- A section that contains info such as copyright data, related documents, and links to social media.
- Typically at the bottom of the page, but not required.



<figure>

- More semantic than . Can include captions and multiple multi-media resources.

```
<figure>
  
  
  
  <figcaption> Our menu items </figcaption>
</figure>
```



(Some) Other New HTML5 Tags

- Structural Elements
 - article, aside, main, summary, section
- Form Elements
 - datalist, keygen, output
- Input Types
 - color, date, email, list
- Graphics Elements
 - canvas, svg
- Media Elements
 - audio, embed, source, track, video



“Notes”

1. Learn to Use Inspect Element – You will use it OFTEN
2. Get in the habit of using the validators
3. Don't put your class files in a “deep” folder structure. You will be accessing the files frequently.



“Learning” HTML

- In the beginning you worry about *syntax*
 - How do I use tags?
- Later, you will worry about *semantics*
 - Is there a tag that better conveys the meaning I am trying to get across?
 - If someone is searching my page can they find what they need and access it easily?



Odds and Ends

- The Request/Response cycle
- The importance of:
 - index.html and knowing your URL
 - Code organization – always know what file is being edited.
 - Tag display values – often *block* or *inline*
 - Adding extensions to your editor
 - Accessibility – you don't get to decide what is/isn't important



Acknowledgements

- These slides are Copyright 2019- Colleen van and made available under a Creative Commons Attribution 4.0 License. Please maintain this last slide in all copies of the document to comply with the attribution requirements of the license. If you make a change, feel free to add your name and organization to the list of contributors on this page as you republish the materials.
- Initial Development: Colleen van Lent , University of Michigan School of Information

