



Document Object Model

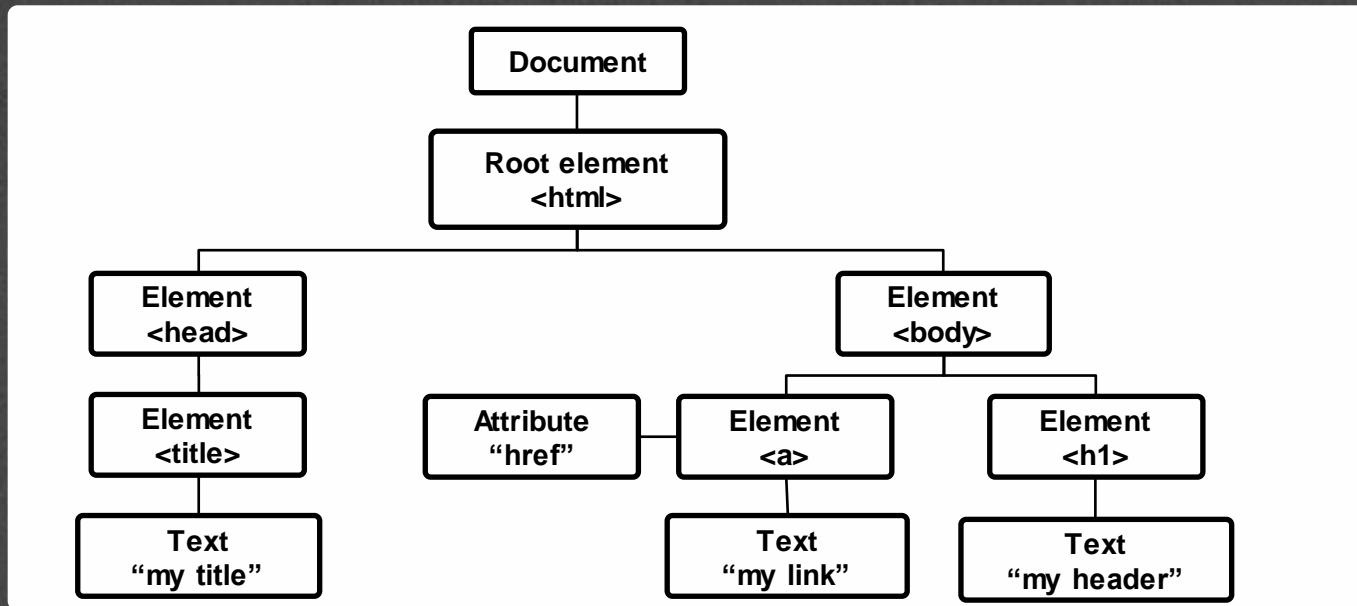
Writing clean code



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



Adapted from w3Schools.com



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**
- **Other than title, meta-data is not displayed**



Body

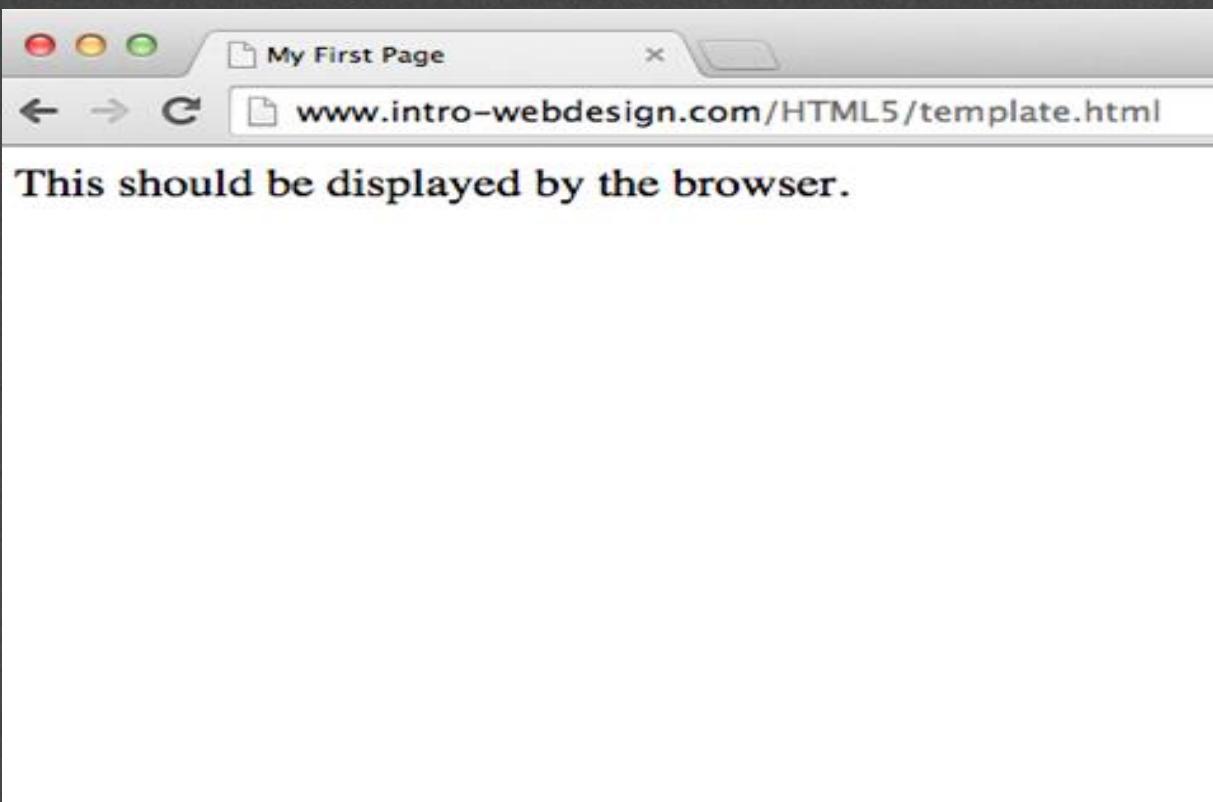
- Bulk of your page
- Important to write well-formatted (tree-like) code.
- Most of the content is displayed by the browser, but there may be some meta data too



Example

Example: template.html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>My First Page</title>
</head>
<body>
    This should be displayed by the browser.
</head>
</html>
```



Validate the Code



This validator checks the [markup validity](#) of Web documents in HTML, XHTML, SMIL, MathML, etc. If you wish to validate specific content such as [RSS/Atom feeds](#) or [CSS stylesheets](#), [MobileOK content](#), or to [find broken links](#), there are [other validators and tools](#) available. As an alternative you can also try our [non-DTD-based validator](#).



[Try now the W3C Validator Suite™](#) premium service that checks your entire website and evaluates its conformance with W3C open standards to quickly identify those portions of your website that need your attention.



Success!!

This document was successfully checked as HTML5!

Result:	Passed, 2 warning(s)
Address :	<input type="text" value="http://www.intro-webdesign.com/HTML5/template.htm"/>
Encoding :	utf-8 <input type="button" value="(detect automatically)"/>
Doctype :	HTML5 <input type="button" value="(detect automatically)"/>
Root Element:	html



Review

- Well-formed pages use the DOM structure
 - Use beginning and end tags
 - Close inner tags before outer ones
 - Use valid attributes
- Browsers will “fix” bad code, but not always well.
Use a validator to check your code



© Colleen van Lent
University of Michigan
School of Information

Unless otherwise noted, this work is licensed under the
CC BY-NC 4.0 license.



HTML5 Tags and Syntax

My first big disappointment to you



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..)

```
<h1>Hello World</h1>
```

Start tag

Closing tag

```

```

Self-closing tag



Display

- One of the most important attributes of an element is its display. The two most common are ***block*** and ***inline***
 - **block (can take width and height)**
 - Newline is inserted before and after, e.g. it “Takes up” whole width
 - **inline (can not take width and height)**
 - Only uses as much space as needed to contain the element.



Common Tags

- **Headings (block)**
 - <h1>, <h2>, <h3>, <h4>, <h5>, <h6>
 - **These tags have syntax and semantics**
- **Paragraphs (block)**
 - <p> </p>
 - **Should only contain inline elements**
- **Divs (block)**
 - <div>...</div>
 - **Generic section that is larger than a paragraph**



More tags

- **Ordered lists**

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

- **Unordered lists**

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

- **Line breaks**

```
<br>
```



Attributes

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



Images

- **Images (inline)**

```
<img src = "myPicture.jpg" alt = "Image of Colleen">
```

- **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.**



Images

```

```

Extra formatting (height,
width, position, etc.)



More Attributes

- **As you learn the tags, you learn their specific attributes. Some apply to any tag**
 - **class – applies special properties to groups of elements**
 - **id – specifies a unique id to one element on the page**
 - **style – specifies a certain visual style (avoid this one!!!)**
 - **accesskey – a shortcut key to activate an element**
 - **tabindex – the order elements will come into focus using the tab key.**



Special Entities

- **Tags always start with a bracket (<)**
- **What if you want the browser to display a bracket, not start a tag?**



Special Entities

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



Review

- How do you know the difference between a tag and an attribute?
- What symbol ends a self-closing tag?



© Colleen van Lent
University of Michigan
School of Information

Unless otherwise noted, this work is licensed under the
CC BY-NC 4.0 license.



Semantic HTML5 Tags

Making the most of the new tags



How to Design

- The most important step in web design is the *design*.
- You need a clear picture of what you want to create, before you can begin coding.



How to Design

<header>

<section>

<article>

<aside>

<footer>



Using Semantic Tags

- In the beginning (insert dramatic music of your choice...) there was div
- <div> was a way to group related content together
- Divs almost always had special classes/ids associated with them

```
<div class = “header”>...</div>
```

```
<div class = “section”>...</div>
```

```
<div class = “footer”>...</div>
```



<header>

- **A group of introductory or navigational aids: title, navigation links, etc.**

```
<header>
  <h1>This is the Title</h1>
  <h2>The author is Colleen</h2>
</header>
```

- **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.**

```
<nav>
    <ul>
        <li><a href="#overview">Overview</a></li>
        <li><a href="#history">History</a></li>
        <li><a href="#development">Development</a></li>
    </ul>
</nav>
```

- **Often found in the <header> tag**



<footer>

- **A section that contains info such as copyright data, related documents, and links to social media**

```
<footer>
    &copy; 2015 by Colleen van Lent<br>
    <a href= "http://www.intro-webdesign.com/HTML5">Introduction to
    HTML5 </a>
</footer>
```

- **Typically at the bottom of the page, but not required.**



<figure>

- More semantics than . Can include:
 - **caption**
 - **multiple multimedia resources**

```
<figure>
    
    <figcaption>
        A sunset over Lake Erie. Taken in Ashtabula Ohio.
    </figcaption>
</figure>
```



Other New Tags

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



Review

- The age of <div> is ending
- Semantic tags help guide users to information in your page



© Colleen van Lent
University of Michigan
School of Information

Unless otherwise noted, this work is licensed under the
CC BY-NC 4.0 license.



Template Page

Create something in Visual Studio Code!!



Minimum requirements

- **Create a page called index.html**
- **Page should utilize header, main, and footer elements. These elements should NOT be empty**
- **Make sure the page validates!**



Start with a Shell

- **Doctype**
- **Meta-Data**
- **Displayable content**



© Colleen van Lent
University of Michigan
School of Information

Unless otherwise noted, this work is licensed under the
CC BY-NC 4.0 license.



Template Page

Create something in Replit!!



Minimum requirements

- **Create a page called index.html**
- **Page should utilize header, main, and footer elements. These elements should NOT be empty**
- **Make sure the page validates!**



Start with a Shell

- **Doctype**
- **Meta-Data**
- **Displayable content**



© Colleen van Lent
University of Michigan
School of Information

Unless otherwise noted, this work is licensed under the
CC BY-NC 4.0 license.



Images

How to add images to your page



Images – it's more than the tag

- Many file types are widely supported
 - JPEG (.jpg and .jpeg), GIF, and PNG
 - SVG and BMP are additional options
 - File extensions must be included
- Every image must be downloaded, so size can be a factor
- Every image requires an HTTP Request



Image Sizes

- When you link to an image the browser displays the image as big (or small) as the file.
 - This size is rarely optimal
 - “Quick” solutions – change file, use width/height attributes



Using an Editor

- Editors can be used to *permanently* change the size of the image
 - Only works on local files
- Built-in software for this includes Preview (Mac) and Paint (Windows)



Using Attributes

- Always strive to keep style out of your HTML files but...
 - Some style may improve accessibility
 - `` tag includes width and height attributes

Default Image Size

```
<figure>
    
    <figcaption> Default image size</figcaption>
</figure>
```



Using Width in Pixels

```
<figure>
    
    <figcaption>Set image size</figcaption>
</figure>
```

Using Width and Height

```
<figure>
    
    <figcaption>Skewed image size</figcaption>
</figure>
```

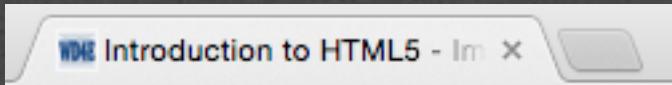


Using Percentages

```
<figure>
    
    <figcaption>Relative image size</figcaption>
</figure>
```

Favicons

- You can put image/logo/icon next to the title of your page (in the tab)



- Must go in <head> section

```
<link rel = "icon" type = "image/png" href = "imgs/wd4elogo.png">
```



Alternative Text Attribute

- Provides a textual alternative to non-text content
- Read by screen readers
- Displayed in place of images
- Provides semantic meaning for search engines



Review

- **Misuse of file extensions, filename, and file paths are often a problem**
- **For now, style the height/width in the html code.**



© Colleen van Lent
University of Michigan
School of Information

Unless otherwise noted, this work is licensed under the
CC BY-NC 4.0 license.



Accessible Images

Making your images accessible and inclusive



Alternative Text Attribute

- Provides a textual alternative to non-text content
- Read by screen readers
- Displayed in place of images
- Provides semantic meaning for search engines



Creating Good alt text

- Be accurate
- Be succinct
- Don't be redundant
- Don't include “picture of..”, “graphic of ..”



Empty alt text

- It is okay to leave alt text empty (null)
 - Decorative images used for non-informative purpose
- Do not skip the alt attribute though!



Long alt text

- Some images (especially infographics) may require elaborate alt text
- Consider replacing alt text with link to separate page with full explanation



Finding Usable Images

- Where can you find images for your site
 - Personal images
 - Images from image-sharing sites
 - Images with **creative commons usage**
 - Icons



Emojis and Icons

- A description of an emoji will be read by a screen reader, but not for an icon.
- Since icons are not images, they can't use the alt attribute.
- Instead, icons can use an aria-label attribute

```
<i class="fa-brands fa-pinterest"></i>
```

```
<i class="fa-brands fa-pinterest" aria-label="Pinterest"></i>
```



Images for Impact

- **Don't constrain yourself to the most common images**
 - **Include images of food from different cultures, athletes in adaptive sports, people of different body types.**
- **Using diverse images has the ability to draw more people to your site.**



Tips

- Utilize guidelines: [alt Decision Tree](#) | [Web Accessibility Initiative \(WAI\)](#) | [W3C](#)
- Add aria-labels when you can't add alt text
- Avoid excessive emojis
- Diversify your images



© Colleen van Lent
University of Michigan
School of Information

Unless otherwise noted, this work is licensed under the
CC BY-NC 4.0 license.



Hyperlinks

Creating a linked document



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.**



Anchor links

- The `<a>` tag stands for *anchor link*
- Needs a **hyper-reference AND content**
 - **href:** reference to location of new content
 - **content:** the “clickable” part (text or image)

```
<a href = "http://www.umich.edu">University of Michigan</a>
```

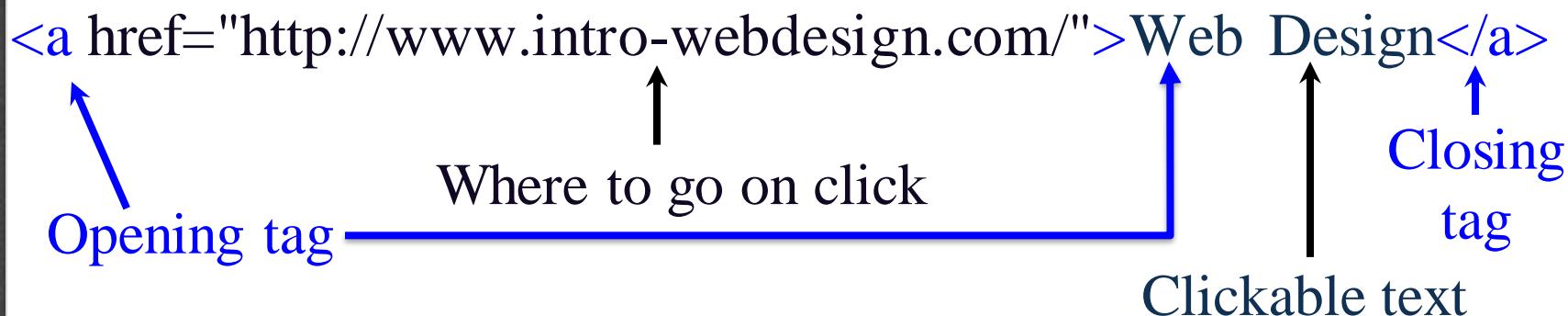


Types of links

- **Absolute**
- **Relative**
- **Internal**
- **Graphical**

Absolute reference

```
<a href="http://www.intro-webdesign.com/">Web Design</a>
```



The diagram illustrates the structure of an HTML anchor tag (`<a>`) with the following annotations:

- Opening tag**: Points to the start of the tag (`<a`).
- Where to go on click**: Points to the attribute (`href="http://www.intro-webdesign.com/"`).
- Clickable text**: Points to the text "Web Design".
- Closing tag**: Points to the end of the tag (``).

Relative References

```
<a href = "page2.html">Second Page</a>
```

Link to a local file in the same folder

```
<a href = "docs/page2.html">Second Page</a>
```

Link to a local file in a different folder called “docs”

```
<a href = "#history">History section</a>
```

Link to a different location in the same file



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

C:\page2.html



Using Images as the Link

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

```
<a href = "http://www.redcross.org">  
  <img src = "imgs/redcross-logo.png" alt = "Red Cross logo"></a>
```

```
<a href = "http://www.redcross.org">  
  <img src = "http://www.redcross.org/images/redcross-logo.png"  
    alt = "Red Cross logo"> </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**



Targets

- Anchors can take a target attribute
 - **_self** - default action
 - **_blank** – open in new tab or window
 - **_top** and **_parent**



Review

- **A page without links is rare**
- **Links can be absolute, relative and internal**
- **Use caution when using images in links**



© Colleen van Lent
University of Michigan
School of Information

Unless otherwise noted, this work is licensed under the
CC BY-NC 4.0 license.



Useful Tags

**Tags for blocks of code and
simple snippets**

Choosing Your tags

- **Generic:** `<p>`, `<div>`
- **Semantic:** `<header>`, `<nav>`, `<footer>`, `<figure>`



Block Tags

- Containers
 - <article>, <aside>, <section>, <main>, ...
- <hr>
- <address>
- <blockquote> - has cite attribute
- <details> with <summary>

Inline tags

- **** was the original inline tag for plain text
- **<cite>**
- **<abbr>**
- **<time>**
- **<code>**
- **<sub>** and **<sup>**

Tags that need “more”

- <button>
- <meter>
- <progress>
- <iframe> – often used to embed documents
- <bdo> attribute **dir** (ltr or rtl)
- <map> with <area> -- creates “clickable element in image” but needs JavaScript



Review

- **Use the most specific tag possible**
- **Sometimes tags “don’t work”**
 - **Run your code through a validator, you may have a syntax error**
 - **Run your code in multiple browsers (good idea even if your code looks good)**



© Colleen van Lent
University of Michigan
School of Information

Unless otherwise noted, this work is licensed under the
CC BY-NC 4.0 license.