GENERAL ASSEMBLY 6.2

Front-End Web Development

Albert Penticoss
Web Designer, University of Melbourne

Start your engines!

- Connect to wifi
- Open your text editor
- Open GitHub desktop and GitHub website
- Open schoology

Submit your assignment

- Check your doctype - <!doctype html>
- Commit your assignment code and sync repository in GitHub desktop
- Go to your repository at the GitHub website
- Copy repository link
- Goto schoology Materials > Homework > Assignment 1
- Follow instructions there.

What are we going to cover today?

- Lesson 2 Review - 10min
- Assignment 1 Review - 10min
- Box Model - 45min
- Block vs inline elements - 10min
- CSS Reset vs Normalize - 10min
- Break - 15min
- Nested Selectors & the DOM - 20min
- Lab Time - 60min

LESSON 2 REVIEW

IMAGES (AND FILE PATHS)

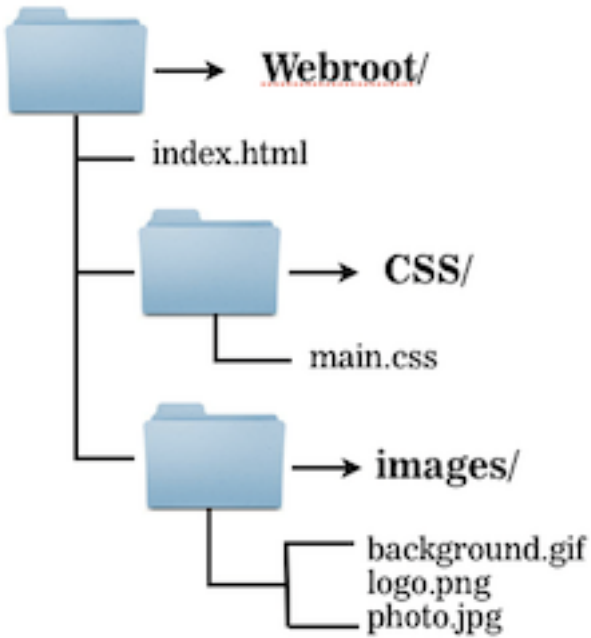
The <img> tag

- Images are placed using the <img> tag.

```
<img src="img/imageName.jpg" alt="alternative text”/>
```

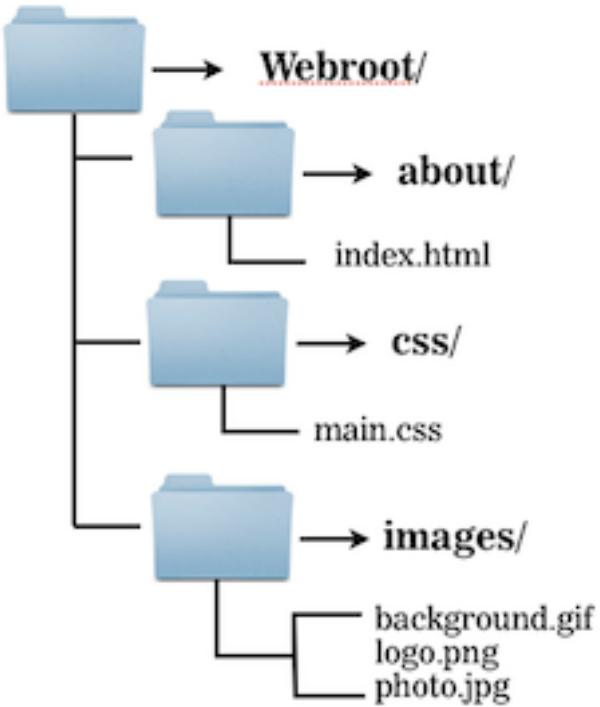
- The img tag requires a src attribute, which tells the browser where to find the image, and
- an alt attribute which provides a text description of the image

How would you write the src?



- There are different approaches to specifying an image location:
  - Relative path
  - Absolute path
  - Fully qualified URL

Relative Path

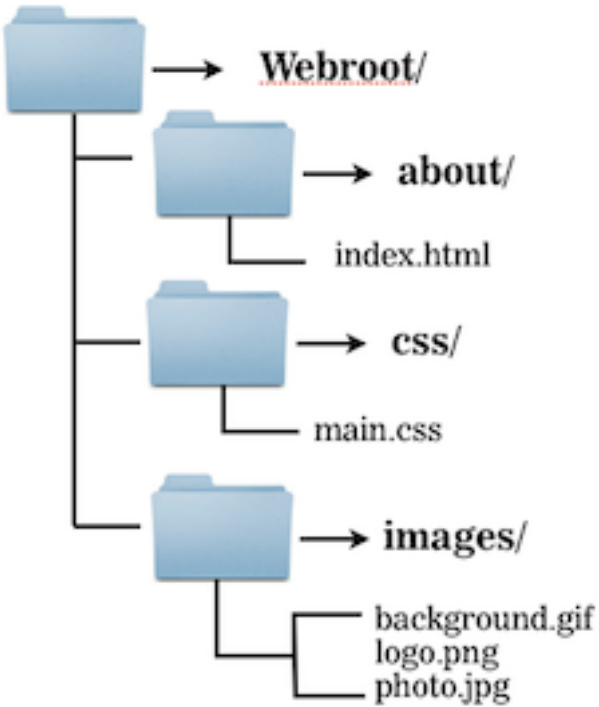


- A relative path is a file path that starts at your html files location
- For this folder structure an <img> tag within the about/index.html file using a relative path would be:

  <img src="../images/logo.png">

- ../ means to go up one directory, and can be used repeatedly: ../.. would go up two directories.

Absolute Path



- An absolute path is a file path that starts at the webfoot
- For this folder structure an <img> tag within the about/index.html file using an absolute path would be:

  <img src="/images/logo.png">

- .. means to go up one directory, and can be used repeatedly: ../.. would go up two directories.

Absolute Path

- The benefit of an absolute path is that this same src path works on any html page, no matter what its location, so the same img tag can be used on both the webroot/index.html page and the webroot/about/index.html page.
- The downside is that the path only works if the project is stored to a proper location for serving.
  - Note: Use http://anvilformac.com/ or https://www.apachefriends.org/index.html for a local development server or Live preview in Brackets

Fully qualified URL

<img src="https://ga-core.s3.amazonaws.com/production/uploads/program/default_image/397/thumb_User-Experience-Sketching.jpg">

- A fully qualified URL is a complete URL just as you would type into your browser.
- Use these for externally hosted images (don't forget to check license before using someone else's image)

Protocol relative URL

- An alternative to a fully qualified URL is the protocol relative url, same syntax but drop the http:

<img src="//ga-core.s3.amazonaws.com/production/uploads/program/default_image/397/thumb_User-Experience-Sketching.jpg">

- This way you don't have to worry if your page is being served by http or https. See this page for details.

Guess what

- These rules for web paths apply to all other tags that need a path
  - e.g. hyperlink (anchor) tag

Link to another page in your site

alt attribute

```
<img src="puppy.jpg" alt="My cute puppy">
```

- A text description of the image or alternative text to be shown if image is unavailable.
- Using alt attributes has the added benefit of giving search engines more linguistic context about the image as it is used on your page.
- alt attributes give screen readers the ability to provide the image content as spoken word or via a braille reader.
- alt attributes are also very important in HTML email because most email programs won't display images unless user chooses to do so.

Image File Formats

There are three main image file formats:

- .png - Supports transparency and semi-transparency, great for logos, icons, and repeating background tiles. Almost always preferable to a gif, unless semi-transparency is not needed, and the gif format is significantly smaller.
- .gif - Can have basic transparency, typically a png is used instead.
- .jpeg - No transparency, can be stored at different compression levels with varying amounts of "lossy-ness", typically the best format for photos. (Try to balance between photo quality and file size.)

Image Size

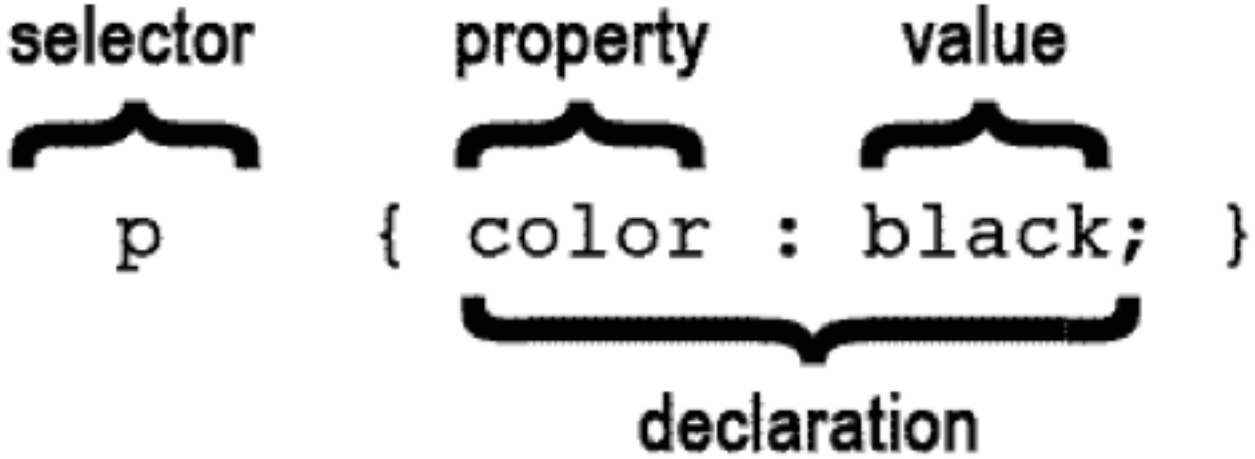
Pro tip:

- Make your images the smallest size possible to increase download speed and minimise your users bandwidth usage.
  - Save your images using ‘save for web’ in adobe products.
  - Choose height and width no bigger than the biggest size you need for your site.
  - Use best format for the type of image, .png, .gif or .jpg.

CSS

CSS rules

A CSS rule-set consists of a selector and a declaration block:



Where does CSS go?

CSS can be added three different ways:

- Inline

  ```
  <p style="color: red;">…</p>
  ```

- In the <head>

  ```
  <style>p {color: red;}</style>
  ```

- In a separate file

Where does CSS go?

Using a separate CSS file

- Its best practice to put CSS in its own file and link to it from the <head>.

```
<link rel="stylesheet" href="style.css"/>
```

- "The link tag needs two attributes: rel="stylesheet" and a href attribute.
- The href attribute value works very similarly to linking to an image, or to another page.

Colours

- Colours can be specified in CSS in a variety of ways:
  - keyword e.g. `p { color: red; } /* red */`
  - hex codes e.g. `p { color: #FF0000; } /* also red */`
  - rgb e.g. `p { color: rgb(255,0,0); } /* also red*/`
  - hsl e.g. `p { color: hsl(360, 100%, 10%); } /* red again */`
  - rgba e.g. `p { color: rgb(255,0,0,0.5); } /* red with 50% opacity */`
  - hsla e.g. `p { color: hsl(360, 100%, 10%, 0.5); } /* also red with 50% opacity */`
  - See http://www.w3schools.com/cssref/css_colors_legal.asp for more examples

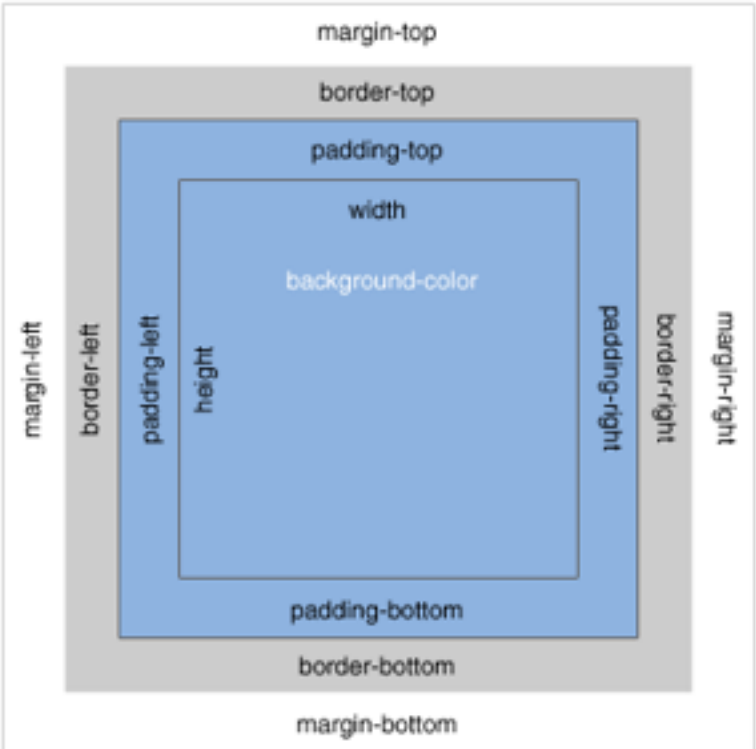
FEWD – LESSON 3

FEWD – LESSON 3

BOX MODEL & MORE CSS

Box model

- Every HTML element is a box.
- We can set the following properties of the box in CSS:
  - width
  - padding
  - border
  - margin



Box model

The way the box model works has been an annoyance to web developers since the beginning of web time:

Because setting width and height in CSS sets the content width or height only.

Actual Width = width + padding-left + padding-right + border-left + border-right

Actual Height = height + padding-top + padding-bottom + border-top + border-bottom

EXERCISE – Tags and Boxes

KEY OBJECTIVES

Understand the box model

AGENDA

20 mins

1. Download L3-Exercise1.zip from schoology and unzip into your exercises repository
2. Code along with instructor

DELIVERABLE

Web page explaining the box model

RESOURCES

Code editor
Starter code

Box model

- Its important to understand the box model because its the default model.

- But we don't need to put up with this @#$% any more!
  - Use the border-box model instead.
    - http://www.paulirish.com/2012/box-sizing-border-box-ftw/
  - It’s safe for projects that support ie8 and above.

BLOCK VS INLINE ELEMENTS

Block vs inline elements

Block level elements e.g. Headings (<h1>through to <h6>), paragraphs <p>, <ul>, <li>, <ol>, etc

- Are 100% wide - i.e. as wide as their parent element
- Therefore they ‘clear’ the element before them, starting on a new line
- Are as high as there children require them to be
- Can have margins and padding applied
- The vertical-align property does not apply

Block vs inline elements

Inline elements e.g. Links <a>, images <img>, , , <code>, etc

- Flows along with text content
- Therefore they are as wide as the content within them and do not ‘clear’ the element before them
- CSS properties width, height, margin-top, margin-bottom, padding-top, padding-bottom cannot be applied (left and right margins can be applied)
- The vertical-align property does apply - it determines how the elements content should align with the surrounding text.

CSS RESET VS NORMALIZE

CSS reset vs normalize

- One of the skills of a veteran front-end developer is experience dealing with browser inconsistencies.
- Even though the HTML standard and its governing body the W3C, has been around for long time, browser companies are only recently adopting a standards based approach to their product development.
- But because of the past and the need to be ‘backward compatible’ there still exists a lot of differences in the default way each different browser renders HTML.

CSS reset vs normalize

- So the front-end development community had to find a way to solve this problem.
- The first solution to this problem to become widely adopted was the ‘CSS reset’
  - The idea behind this was to reset all default styles to a zero base, so everything had no style at all and the developer would define all styles.
  - This was a bit of overkill!

CSS reset vs normalize

- Today the most widely used solution is normailze.css http://necolas.github.io/normalize.css/
- As opposed to CSS resets, Normalize.css:
  - targets only the styles that need normalising
  - preserves useful browser defaults rather than erasing them
  - corrects bugs and common browser inconsistencies
  - improves usability with subtle improvements
  - doesn't clutter the debugging tools
  - has better documentation

CSS reset vs normalize

- Let’s add normalize.css to our template

BREAK TIME

FEWD – LESSON 3

NESTED SELECTORS & THE DOM

Nested Selectors

- A nested selector is a CSS selector that targets elements nested within other elements
- For example an inline element within a block element:

```
<p>Hi I'm a paragraph with an <em>empahsised<em> inline element nested inside me.</p>
```

- Or a block element within another block element:

```
<ul>
    <li>An unordered list</li>
    <li>Has list items</li>
    <li>Nested within it</li>
</ul>
```

EXERCISE - Nested selectors

KEY OBJECTIVES

Understand nested selectors

DELIVERABLE

Web page demonstrating CSS nested element selectors

AGENDA

10 mins

1. Download L3-Exercise2.zip from schoology and unzip into your exercises repository
2. Code along with instructor

RESOURCES

- Text editor
- Starter code

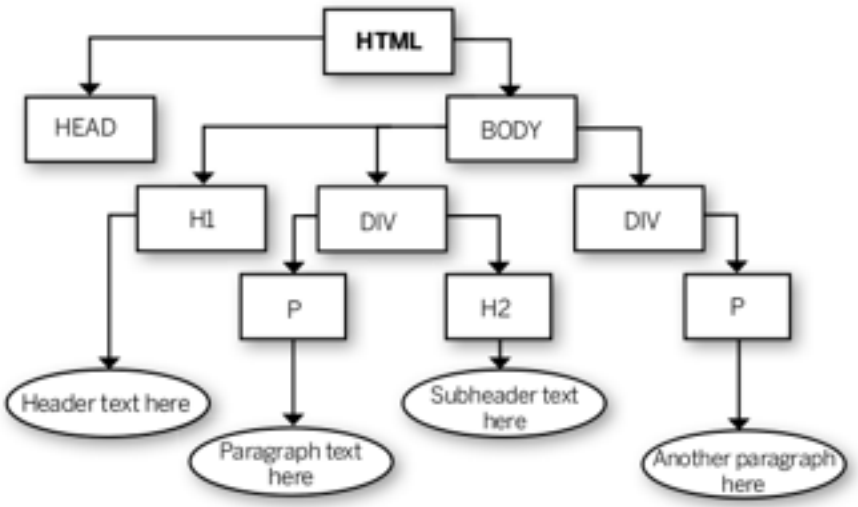
The Document Object Model (DOM)

What is the DOM?

“The Document Object Model (DOM) is an application programming interface (API) for valid HTML and well-formed XML documents. It defines the logical structure of documents and the way a document is accessed and manipulated.” - https://www.w3.org/TR/DOM-Level-2-Core/introduction.html

The Document Object Model (DOM)

- Basically the DOM is the model of a HTML document our browser creates and makes available as an object with nodes made up of elements, attributes, content
- Graphically the DOM can be expressed in tree structure
- Nodes can be parents, siblings, children, descendants, ancestors



EXERCISE - Draw me a DOM

KEY OBJECTIVES

Understand the DOM and how to draw a DOM tree

DELIVERABLE

Dome tree

AGENDA

10 mins

1. Create a new GitHub repository for Assignment 2.
2. Download Assignment2.zip from schoology and unzip into your new repository.
3. Get into groups, review the index.html file, and draw a DOM tree representation of it.

RESOURCES

Assignment 2 starter code
Whiteboard or blackbloard

LAB TIME