



WELCOME TO FEWD

WEEK 1 / LESSON 2

Leonie Dunnett

LESSON 2:

IMAGES AND CSS

BASICS

LEARNING OBJECTIVES

- Predict image paths and apply relative paths to `` and `<a>` tags.
- Define CSS selectors and separation of concerns
- Apply and explain CSS "cascade" including: importance, specificity and inheritance
- Experiment with margin padding and border

TODAY'S SCHEDULE

- HTML Basics Review
- Portfolio HTML **Code Along**
- Adding Images and Folder Paths
- Introducing CSS **Code Along**
- Build Your Portfolio **Lab**

HTML BASICS REVIEW

- Document structure, common elements and attributes, linking external style sheets
- The DOM (Document Object Model)
- Using chrome debugger

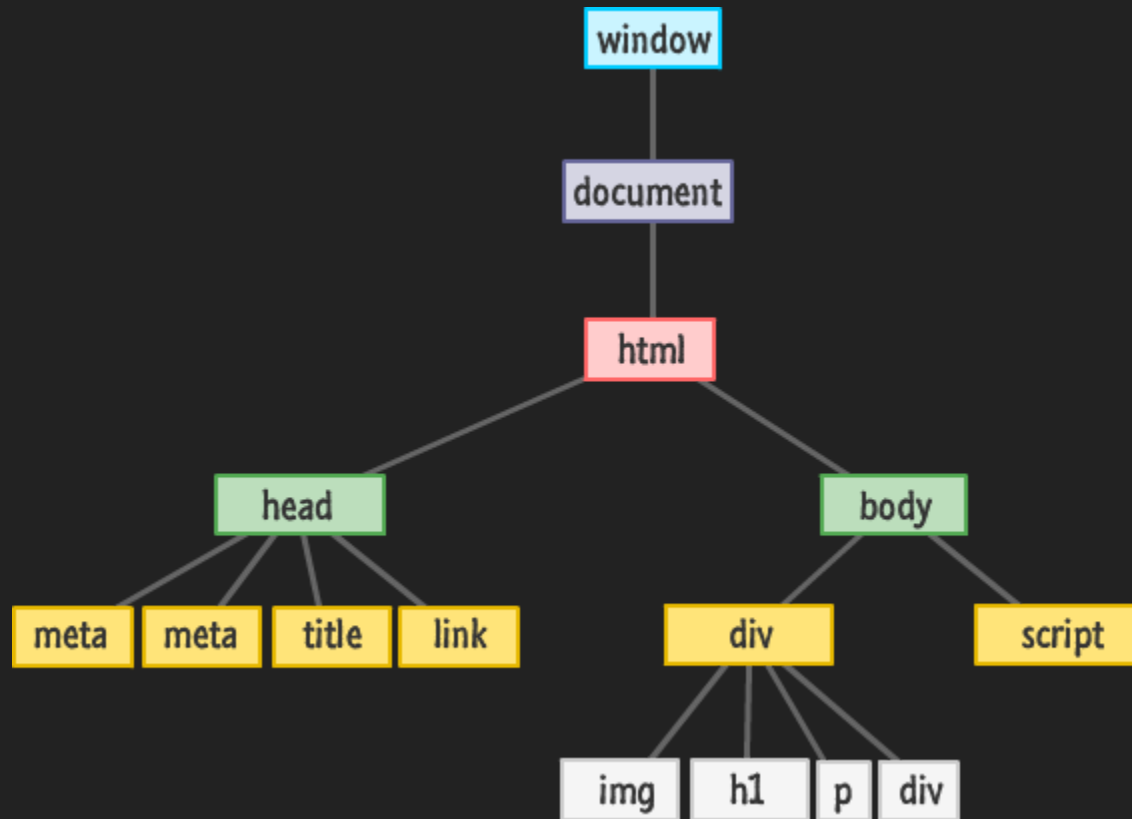
HTML BASICS: PUTTING IT ALL TOGETHER

Structure, elements, attributes, style sheet:

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Some web page</title>
    <link rel="stylesheet" href="css/style.css">
  </head>
  <body>
    <h1>Heading text here</h1>
    <p>Paragraph text here with a
      <a href="LinkUrl">Link</a>
    </p>
  </body>
</html>
```

THE DOM (DOCUMENT OBJECT MODEL)

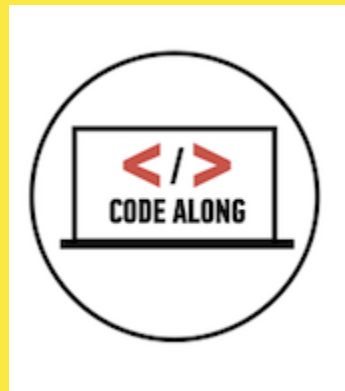
A tree-like structure of your HTML document for accessing and manipulating its content.



CHROME DEBUGGER

Useful to see:

- the DOM tree structure
- the styles applied to the elements
- where the elements are on the page



PORTFOLIO

Build HTML

ADDING IMAGES AND FOLDER PATHS

IMAGE ELEMENT

Images are placed using the `` tag.

```

```

The element has a mandatory `src` attribute.

A recommended attribute is `alt`.

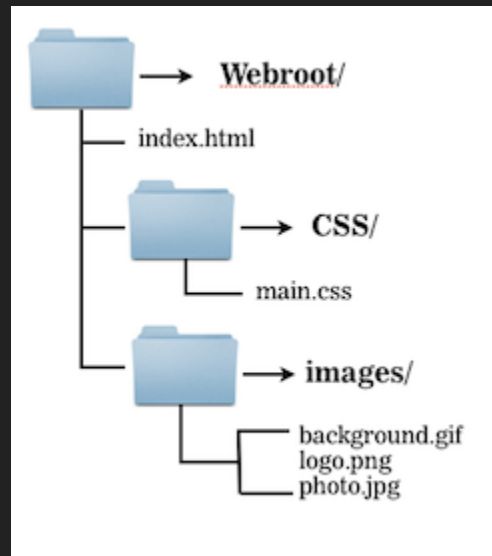
IMAGE SOURCE

The `img` tag requires a `src` attribute, which tells the browser where to find the image.

It's a FILE PATH that's either:

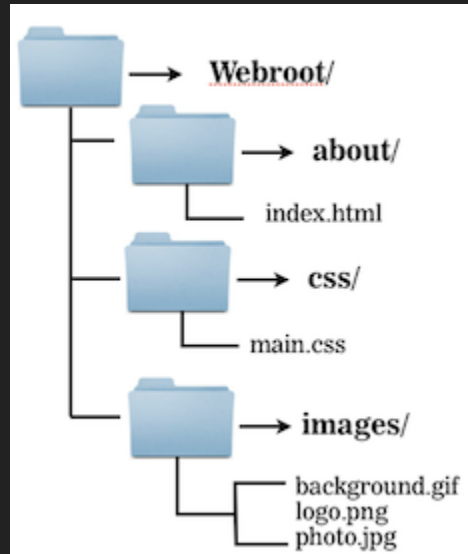
- **relative** (to the current html file)
- **absolute** (with respect to the domain root)
- **a full URL**

RELATIVE: EXAMPLE 1



For a html file served up from the **webroot**, a relative path could be used: ``

RELATIVE: EXAMPLE 2



If the html file is in the **about** folder, the same image would be referenced as ``

Note that `..` means to go up a directory, and can be used repeatedly:
`../../..` would go up two directories.

ABSOLUTE

```

```

- Absolute URLs start with a `/`.
- This tells the web server to start at the domain root.
- Irrespective of the html file location, the image will be retrieved from the images folder under the web root.

ABSOLUTE: PRO AND CON

- **Pro:**

This works on any html page, no matter what its location.

The same `` tag can be used on both the `webroot/index.html` page and the `webroot/about/index.html` page.

- **Con:**

The downside is that the path only works if the project is served from a web server.

FULL URL

```

```

Note: For linking to images, make sure that you have permission to use the image in this way. Even then, it is often better to host a copy of the same image, rather than link to another server, because it reduces dependency.

IMAGE ALT ATTRIBUTE

```

```



- The text that is used when the image is unavailable
- Gives search engines linguistic context for the image
- Screen readers use it to describe the image
- Always include the alt attribute!

REASONS AN IMAGE MAY NOT LOAD:

- Connection error
- File not found
- User running a text-based browser

IMAGE FILE FORMATS

There are three main image file formats:

- png
- gif
- jpg

.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

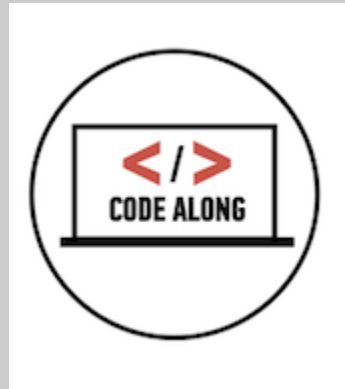
Can have basic transparency, typically a **png** is used instead.

Used for animated images.

.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.)



INTRODUCING CSS

About Me page

WHAT IS CSS?

CSS = style (skin/makeup)

CASCADING STYLE SHEETS

Styles have an affect on the look, the layout, and the positioning of elements.

From the author of CSS, the cascade is defined as:

The process of combining several style sheets and resolving conflicts between them.

WHERE DOES CSS GO?

- Inline
- In the `<head>...</head>`
- In a separate file

CSS *should* go in a separate file. Why ?

USING A SEPARATE CSS FILE

Best practice to link to a separate CSS file in the `<head>`.

The link tag needs two attributes:

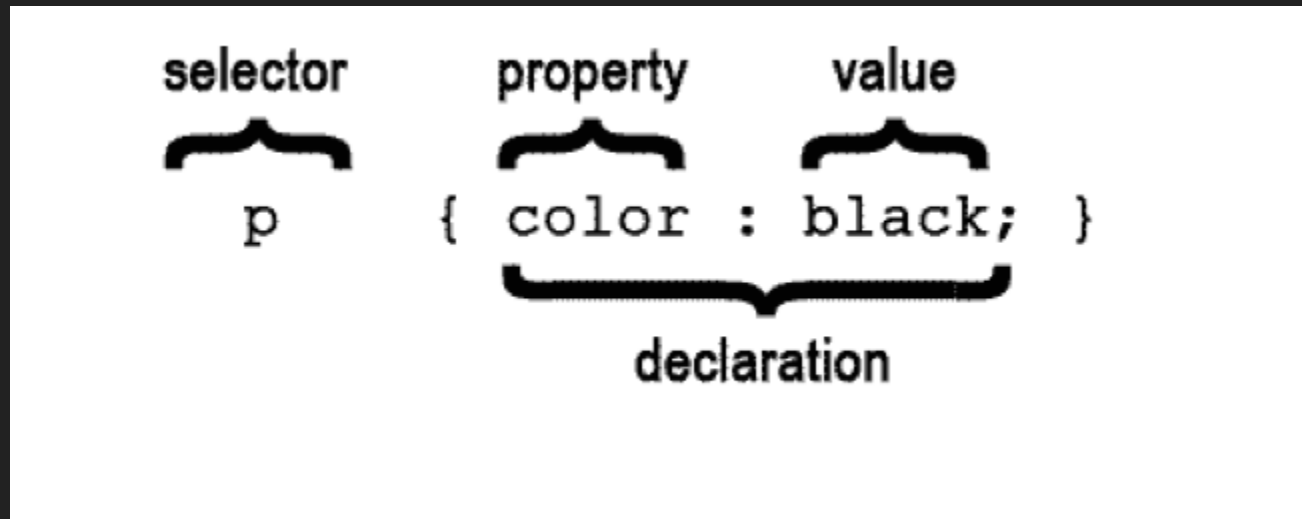
- `rel="stylesheet"` - (relationship)
tells browser linked file is a style sheet
- `href` - works the same as `src` in the `` tag

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

CODE ALONG

Let's add an image and a separate CSS file
to our HTML document

CSS RULE(S)!



This whole thing is called a **rule**.

The browser reads the rules, finds the elements that match the rule, and renders the element using the rules properties.

CSS SELECTOR

```
p {  
  color: red;  
  font-weight: bold;  
}
```

The **selector**, **p** in this case, specifies what parts of the HTML document should be styled by the declaration.

This selector will style all **p** elements on the page.

THE DECLARATION BLOCK

```
{  
  color: red;  
  font-weight: bold;  
}
```

Declarations go inside curly braces.

Every declaration is a **property** followed by a **value**, separated by a colon, ending in a semicolon.

DECLARATIONS

This example has two declarations. Here's the first:

```
color: red;
```

In this declaration, we are setting the **color** property to the value **red**.

DECLARATIONS

Let's look at the second declaration:

```
font-weight: bold;
```

What style **property** are we specifying here?

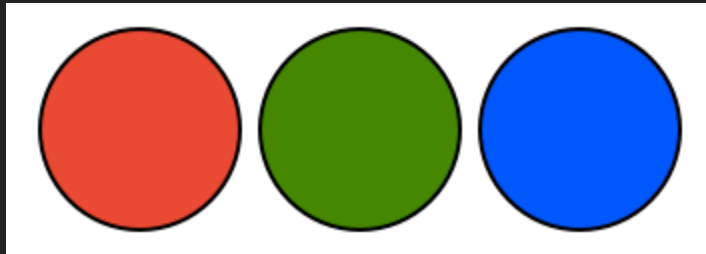
What **value** are we setting that **property** to?

CODE ALONG

Let's write some styles
for text related elements

CSS COLORS

Colors can be specified in CSS in a variety of ways:



- keywords
- hex codes
- `rgb / rgba`
- `hsl / hsla`

KEYWORDS

`color: white;`

These are used less frequently, but are handy for basic colors like `black` and `white`. There are several.

See [here](#) for more.

HEX CODES



color: #6756A7;

The 3 color values (RGB) are expressed after # by two hexadecimal digits, from 00 (no color) to FF (full color).

Hex values can be abbreviated to only 3 digits if each set of digits are doubled. (eg. #00FF00 (green) = #0F0)

RGB

```
color: rgb(0,0,0);
```

- The first value is red, the second green, the third blue.
- Each value can range from 0 to 255, which expresses the same number of color steps as 00 to FF in hex.
- In RGB, `rgb(0,0,0)` is black, `rgb(255,255,255)` is white, `rgb(255,0,0)` is red, etc.

RGBA



```
color: rgba(12, 78, 200, 0.7);
```

- RGBA works identically to RGB, except that it takes a 4th value called the "alpha"
- This is a value between 0 and 1 which will be used to determine a color's opacity on the page.

HSL COLORS

HSL

- Similar notation to RGB values, but specify colors using hue, saturation, and lightness.

HSLa

- As with RGBa, HSLa is exactly like HSL for the first 3 values, but takes a 4th alpha-channel value.

THE CSS CASCADE

CSS reads top to bottom.

If conflicting then last overrides all previous.

```
p {  
  background: blue;  
  font-size: 16px;  
}  
p {  
  background: orange;  
}
```

SPECIFICITY

More specific rules override less specific rules

```
body {background: orange;}  
p {background: blue;}  
  
<body>  
  <p>Paragraph text here.</p>  
</body>
```

INHERITANCE

Child elements inherit rules from parent elements

```
body {font-family: sans-serif;}  
p {font-family: Georgia, serif;}
```

```
<body>  
  <p>Paragraph text here.</p>  
</body>
```

p will inherit **body** font-family if we don't set it

IMPORTANCE

Order of Selector Importance (from least to most):

- Type (eg. `p`, `div`) *0-0-1
- Class (eg. `.example`) *0-1-0
- ID (eg. `#example`) *1-0-0

*Specificity weight point value, calculated using three columns: "ID-Class-Type"

CODE ALONG

Let's add some more styles to our page



YOUR PORTFOLIO

REVIEW: LEARNING OBJECTIVES

- Predict image paths and apply relative paths to `` and `<a>` tags.
- Define CSS selectors and separation of concerns
- Apply and explain CSS "cascade" including: importance, specificity and inheritance
- Experiment with margin padding and border

ASSIGNMENT ASSESSMENT

- Check it out
- Best tool for learning is having a go.
- Enables us to help guide you if you get a little lost

HOMework

Optional Challenge: CodeAcademy HTML/CSS Track

Review: Slides and resources

Code: Finish portfolio page

RESOURCES

[CSS Tricks](#)

[Mozilla's MDN for CSS](#)

[Adobe Kuler](#)

[CSS Specificity](#)

EXIT TICKETS