FEWD - CSS Basics

James Gallichio

Full-Stack Developer, Plattar

Learning Objectives

- Apply and explain CSS cascade including: importance, specificity and inheritance.
- Predict image paths and apply relative paths to `<img>` and `<a>` tags.
- Apply and explain different image file types and when to use them
- Apply and explain CSS colour values and when to use them

Agenda

- HTML Review
  - Image types
  - CSS Hierarchy
  - CSS Colours
- Building A Simple Web Page
- Lab Time

HTML Basics Review

What Tag Is It?

Images

- Images are placed using the `<img>` tag.

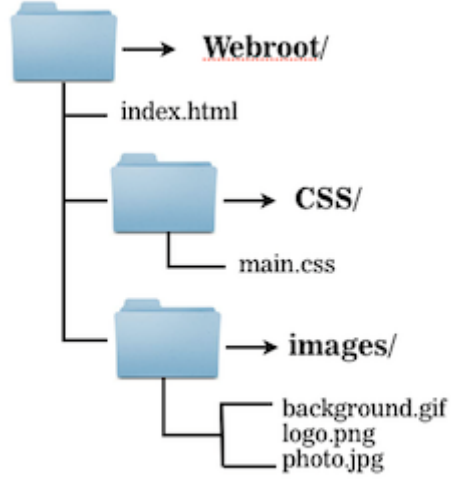
```
<img src="img/imageName.jpg" alt="alternative text">
```

Images

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

Images

How would you write the src?



- There are different approaches to specifying an image location

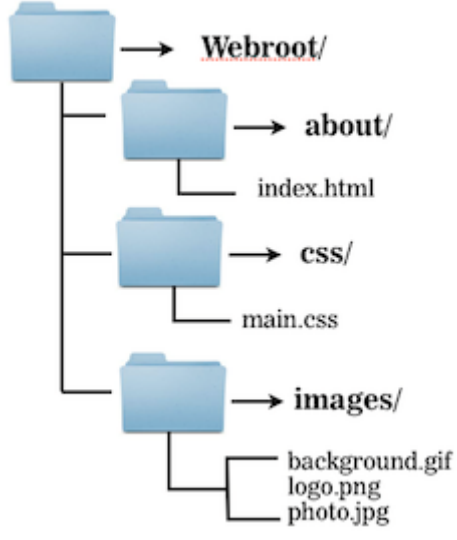
Images

- Inside `webroot`, a relative path could be used:

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

Images

Relative Path



Images

Absolute Path

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

HTML Basics - Images

Full URL

```
<img src="https://ga-core.s3.amazonaws.com/production/uploads/program/default_image/397/thum
```

HTML Basics - Images

alt attribute

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

- A piece of text to be used in lieu of the image when the 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.

HTML Basics - Images

There are three main image file formats:

Image File Formats

#.png

- Supports transparency and semi-transparency, great for logos, icons, and repeating background tiles
- Filesize grows with the number of colours
- Almost always preferable to `gif`, unless semi-transparency is not needed, and the `gif` format is significantly smaller

Image File Formats

#.gif

- Can have basic transparency
- Can be animated
- `png` is preferred unless `gif` is a smaller filesize

Image File Formats

#.jpg

- Can be stored at different compression levels with varying amounts of "lossy-ness"
- No transparency
- Typically the best format for photos. (Try to balance between photo quality and file size)

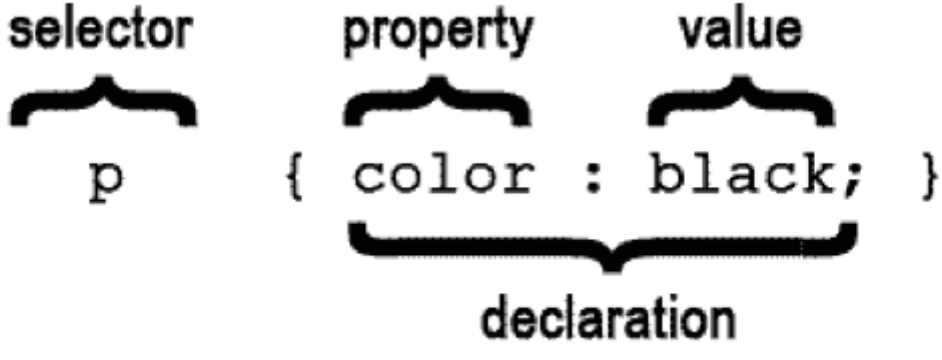
CSS - Cascading Style Sheets

Why CSS?

CSS

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

CSS



CSS Break Down

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

The `p` is called a **selector**, and it's followed by a set of **declarations** in a **declaration block**.

CSS Break Down

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.

CSS Break Down

The **declaration block** here is:

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

Declarations go inside curly braces.

CSS Break Down

Declarations

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

```
color: red;
```

CSS Break Down

Let's look at the second declaration:

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

CSS

Where does CSS go?

- Inline
- In the `head`
- In a separate file (Best practice)

CSS

It's best practice to put CSS in its own file and link to it from the `<head>`.

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

Important:

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

CSS

Why might we want to link to a separate CSS file?

CSS

The Cascade

- Rules are applied in **cascade order**
- The hierarchy of the cascade is determined by

```
* Sequence
* Specificity
* Inheritance
* Importance
```

CSS

Sequence

Later rules take priority over earlier rules

```
h1 {
    color: green;
    color: blue;
    color: red;
}
```

CSS

Specificity

More specific selector rules will apply over more general ones

```
header h1 { color: blue; }
```

wins over

```
h1 { color: red; }
```

CSS

Inheritance

- Elements will inherit (some) rules of their parent

```
<header>Heading
    <p>I am also red and 18px<p>
</header>

header {
    color: red;
    font-size: 18px;
}
```

- Layout rules like `margin`, `padding` etc are not automatically inherited

CSS

Importance

The natural cascade flow can be overidden by:

- Inlining CSS in HTML
- Declaring CSS in HTML `<head>`
- `!important`

  Generally best avoided except in very rare situations.

CSS

Colours

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

Colour

Colour Keywords

Very limited compared to other value types but handy for basic colors like `black` and `white`

Colour

Hex Codes (RGB)

#FF0000 (full red, no green, no blue)

#00FF00 (no red, full green, no blue)

#0000FF (no red, no green, full blue)

Colour

RGB Colour Values

`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 0 to FF in base-16.
- In RGB, `rgb(0,0,0)` is black, `rgb(255,255,255)` is white, `rgb(255,0,0)` is red, etc.

Colour

RGBa Colours

Colour

RGBa Colours

- RGBa works identically to RGB, expect 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
- 0 is completely transparent, and 1 being solid. 0.5 or .5 is 50% opacity.

Colour

HSL Colours

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

Colour

HSLa

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

Colour

HSL Colours

Hue is expressed as a degree angle measure (or colour circle), with red being at 0, green at 12 and blue at 240.

Colour

HSL Colours

Saturation is expressed as a percentage, with 100% being a fully saturated color, and 0% bein
a shade of gray.

Colour

HSL Colours

Lightness is also expressed as a percentage, 0% being black, and 100% being white.

CSS Colour

- Keywords, Hex and RGBA are most commonly used and widely supported
- HSLA can be easier to work with if importing colours from Photoshop or performing colour transformations

CSS Colour

Where can we use it?

- Text (`color`)
- Backgrounds (`background-color`)
- Borders (`border-color`)

About Me

Lab Time

- Your Portfolio!

Exit Ticket