

Review HTML

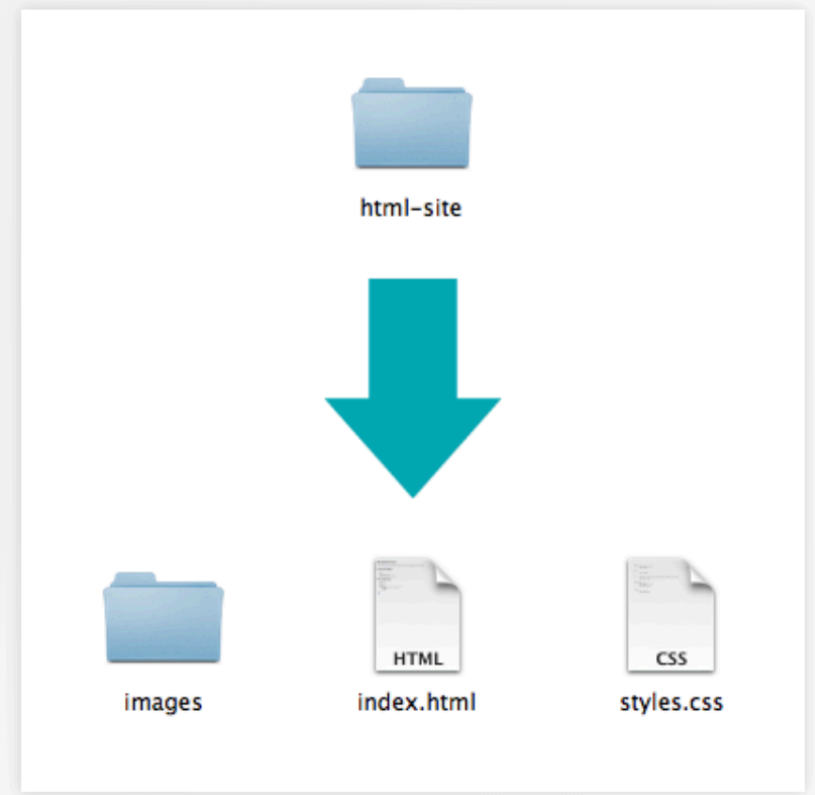
file paths and directories

Get Started: Folder Structure

All the files for your site should be stored within the same folder.

This includes:

- HTML Files
- CSS Files
- Images
- Script files
- Anything else that will appear on your site



ABSOLUTE URLS

When you link to a different website, the value of the href attribute will be the full web address for the site, which is known as an **absolute** URL.

ABSOLUTE URLS

URL stands for Uniform Resource Locator. Every web page has its own URL. This is the web address that you would type into a browser if you wanted to visit that specific page.

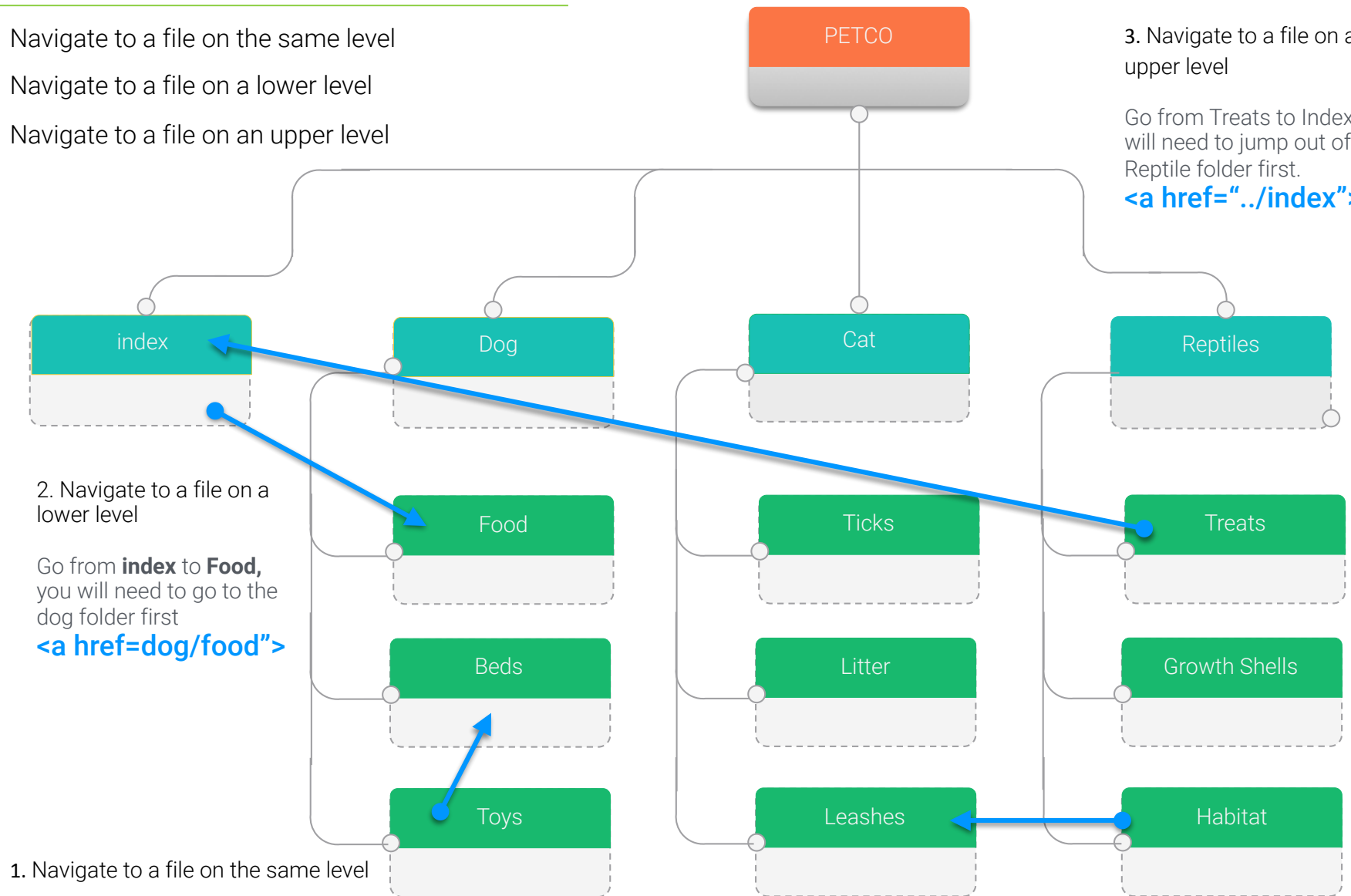
An absolute URL starts with the domain name for that site, and can be followed by the path to a specific page. If no page is specified, the site will display the homepage.

RELATIVE URLS

When linking to other pages within the same site, you can use relative URLs. These are like a shorthand version of absolute URLs because you do not need to specify the domain name.

3 types of relative paths

1. Navigate to a file on the same level
2. Navigate to a file on a lower level
3. Navigate to a file on an upper level



3. Navigate to a file on an upper level

Go from Treats to Index, you will need to jump out of the Reptile folder first.

[](../../index)

2. Navigate to a file on a lower level

Go from **index** to **Food**, you will need to go to the dog folder first

[](/dog/food)

1. Navigate to a file on the same level

Go from **Toys** to **Beds**

[](/toys)

Another Example: If you are on **Habitat** and you want to go to **Leashes**:

[](../../cat/leashes)



```
<p>
  <ul>
    <li><a href="index.html">Home</a></li>
    <li><a href="about-us.html">About</a></li>
    <li><a href="movies.html">Movies</a></li>
    <li><a href="contact.html">Contact</a></li>
  </ul>
</p>
```

When you are linking to other pages within the same site, you do not need to specify the domain name in the URL. You can use a shorthand known as a **relative** URL.

If all the pages of the site are in the same folder, then the value of the href attribute is just the name of the file.

PARENT

The *examplearts* folder is a parent of the *music* folder.

CHILD

The *music* folder is a child of the *examplearts* folder.



GRANDPARENT

The *examplearts* folder is a grandparent of the *dvd* folder.

GRANDCHILD

The *dvd* folder is a grandchild of the *examplearts* folder.

If all of the files in your site are in one folder, you simply use the file name for that page.

If your site is organized into separate folders (or directories), you need to tell the browser how to get from the page it is *currently on* to the page that you are *linking to*.

If you link to the same page from two different pages you might, therefore, need to write two different relative URLs.

These links make use of the same terminology

STRUCTURE

The diagram on the right shows the directory structure for a fictional entertainment listings website called ExampleArts.

The top-level folder is known as the **root** folder. (In this example, the root folder is called *examplearts*.) The root folder contains all of the other files and folders for a website.

Each section of the site is placed in a separate folder; this helps organize the files.

RELATIONSHIPS

The relationship between files and folders on a website is described using the same terminology as a family tree.

In the diagram on the right, you can see some relationships have been drawn in.

The *examplearts* folder is a parent of the *movies*, *music* and *theater* folders. And the *movies*, *music* and *theater* folders are children of the *examplearts* folder.

HOMEFILES

The main homepage of a site written in HTML (and the homepages of each section in a child folder) is called *index.html*.

Web servers are usually set up to return the *index.html* file if no file name is specified.

Therefore, if you enter `examplearts.com` it will return `examplearts.com/index.html`, and `examplearts.com/music` will return `examplearts.com/music/index.html`.

Every page and every image on a website has a **URL** (or Uniform Resource Locator). The URL is made up of the domain name followed by the **path** to that page or image.

The path to the homepage of this site is `www.examplearts.com/index.html`. The path to the logo for the site is `examplearts.com/images/logo.gif`.

You use URLs when linking to other web pages and when including images in your own site. On the next page, you will meet a shorthand way to link to files on your own site.

The root folder contains:

- A file called *index.html* which is the homepage for the entire site
- Individual folders for the movies, music and theatre sections of the site

Each sub-directory contains:

- A file called *index.html* which is the homepage for that section
- A reviews page called *reviews.html*
- A listings page called *listings.html* (except for the DVD section)

The movies section contains:

- A folder called *cinema*
- A folder called *DVD*.

RELATIVE LINK TYPE

EXAMPLE (from diagram on previous page)

SAME FOLDER

To link to a file in the same folder, just use the file name. (Nothing else is needed.)

To link to music reviews from the music homepage:

```
<a href="reviews.html">Reviews</a>
```

CHILD FOLDER

For a child folder, use the name of the child folder, followed by a forward slash, then the file name.

To link to music listings from the homepage:

```
<a href="music/listings.html">Listings</a>
```

GRANDCHILD FOLDER

Use the name of the child folder, followed by a forward slash, then the name of the grandchild folder, followed by another forward slash, then the file name.

To link to DVD reviews from the homepage:

```
<a href="movies/dvd/reviews.html">Reviews</a>
```

PARENT FOLDER

Use ../ to indicate the folder above the current one, then follow it with the file name.

To link to the homepage from the music reviews:

```
<a href="../index.html">Home</a>
```

GRANDPARENT FOLDER

Repeat the ../ to indicate that you want to go up two folders (rather than one), then follow it with the file name.

To link to the homepage from the DVD reviews:

```
<a href="../../index.html">Home</a>
```

A forward slash will return the homepage for the entire site, and a forward slash followed by a file name will return that file providing it is in the root directory.

Naming Folders + Files

- Name your file index.html
- In file and folder names, only use lowercase letters, numbers, hyphens/dashes.
- File names are usually case sensitive: INDEX.html vs. index.html
- Use the right extension: .html vs .css vs .js

Element: Link

To make a link, you need 3 parts:

- `<a>` tag
- href attribute: web address where the link points
- content: text or images between the tags that become the clickable link.

```
<a href="www.codingoneandtwo.com">Our Class website</a>
```

Will produce

[Our Class Website](http://www.codingoneandtwo.com)

Paths for Links & Images

Relative

Links within the same directory need no path information:

```
<a href="filename.jpg">A file in same folder</a>
```

Subdirectories are listed without preceding slashes:

```
<a href="projects/another_file.html">A file from the projects folder</a>
```

Absolute

Typically used when pointing to a link that is not within your own domain. Points to a specific location of a file, including the domain.

```
<a href="www.codingoneandtwo.com/coding-1-syllabus.html">Our Class Syllabus</a>
```

Link Targets

Optional target attribute tells the browser to open the link in a new tab.

```
<a href="home.html" target="_blank">Link Text</a>
```

Other target properties are:

_blank

Opens the linked document in a new window or tab

_parent

Opens the linked document in the parent frame

_self

Opens the linked document in the same frame as it was clicked (this is default)

_top

Opens the linked document in the full body of the window named frame

Check for Understanding

What are a few different HTML tags?

Check for Understanding

Which tag is used to create a link to another page?

1. `<p>`
2. `<link>`
3. `<a>`
4. `<america>`

Check for Understanding

What are the two tags that nest directly within the `<html>` tags?

Check for Understanding

What is a relative versus absolute path?

Intro to CSS

Intro to CSS

1. **css syntax**
2. **linking html to css**
3. **cascade specificity**



CSS

Specificity,
Inheritance and
Cascade
(and multiple classes)

<http://www.vanseodesign.com/css/css-specificity-inheritance-cascade/>

clear float
clear fix

Inheritance and Cascade

If the rules are equal in specificity individual rules get overridden **in the order they're defined in the CSS**, so in your example red wins because it comes later in the CSS definitions. The same rule applies in other cases as well, for example:

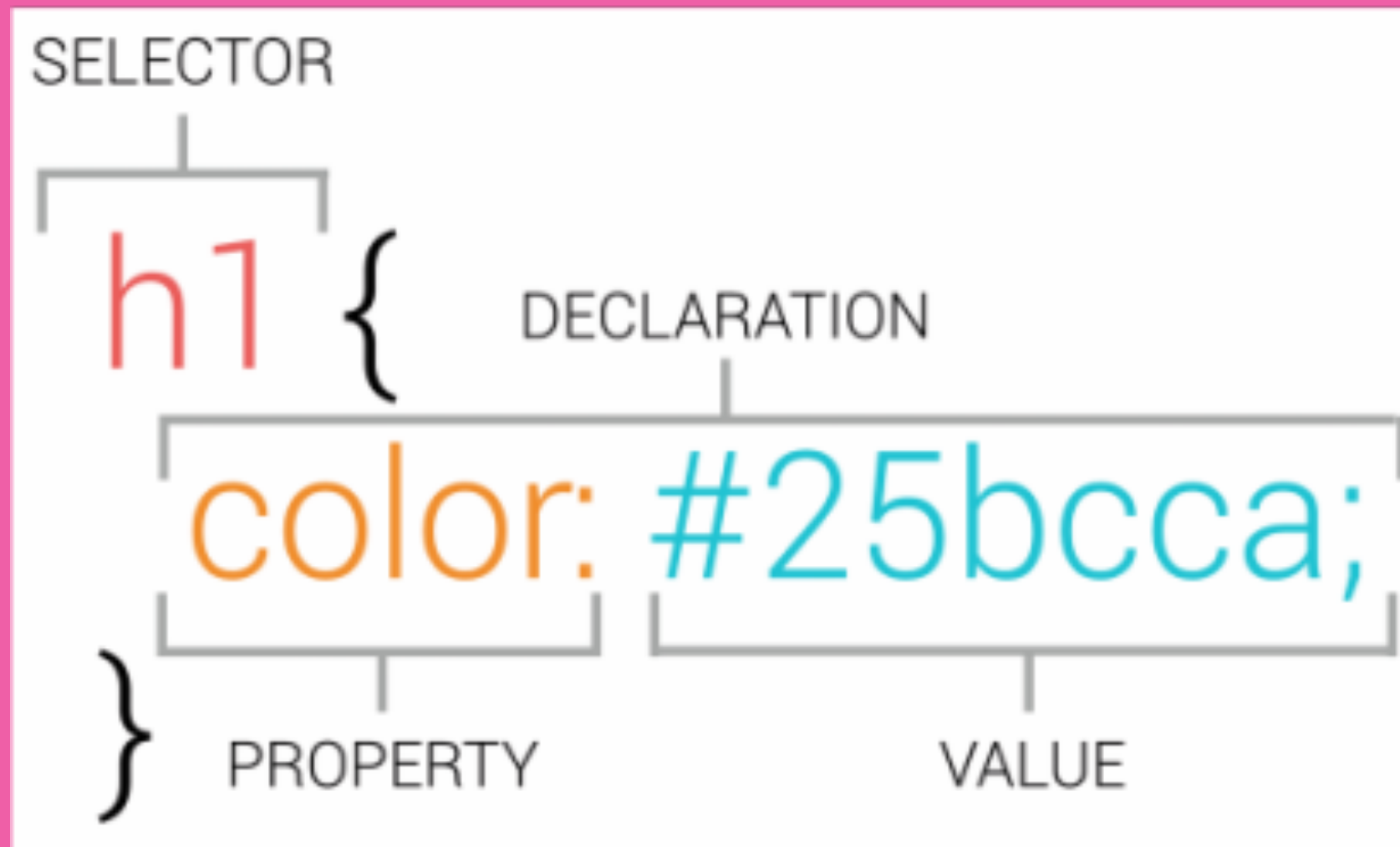
```
<div class="red green">
```

Which of these wins?

```
.green { color: green; }  
.red { color: red; }
```

.red wins here, it doesn't matter the order in the class attribute, all that matters is the order the styles are defined in the CSS itself.

css syntax



Anatomy of CSS

CSS consists of style rules. Each style rule consists of a selector and declarations of property-value pairs:

```
selector {  
  property: value;  
  property: value;  
}
```

For example:

```
body {  
  color: yellow;  
  background-color: black;  
}
```

Property: Color

The color property changes the color of the text.

```
p {  
  color: red;  
}
```

Property: Background-color

The background-color property changes the color of the background.

```
body {  
  background-color: hotpink;  
}
```

CSS Color Values

Your browser can accept colors in many different ways:

Color name:	red
Hexadecimal value:	#FF0000
RGB value:	rgb(255, 0, 0)
HSL value:	hsl(0, 100%, 100%)

[W3Schools Color Picker](#)
[CSS Color Names](#)

Let's develop it

- Add some rules to your css file
- Change the font color and background color of different types of elements
- Try changing the color of paragraphs, headings, and lists

Types of Selectors: Element

```
p {  
  property: value;  
}
```

Selects all paragraph elements.

```
img {  
  property: value;  
}
```

Selects all image elements.

selector value property

```
<head>
  <title>Example</title>
  <style>
    h3 {
      font-family: "Oswald", Arial, sans-serif;
    }
  </style>
</head>
```

intro to css: 3 ways to link to styles

1. inline
2. embedded
3. external

intro to css: 3 ways to link to styles

1. connect CSS to HTML inline

```
<p style="color:red">Some text.</p>
```

- Uses the HTML style attribute.
- Only applies to one element at a time.
- Not preferred.

intro to css: 3 ways to link to styles

2. connect css to html embedded

```
<head>
  <style>
    p {
      color: blue;
      font-size: 12px;
    }
  </style>
</head>
```

- Inside <head> element.
- Uses <style> tag.
- Only applies to that page.

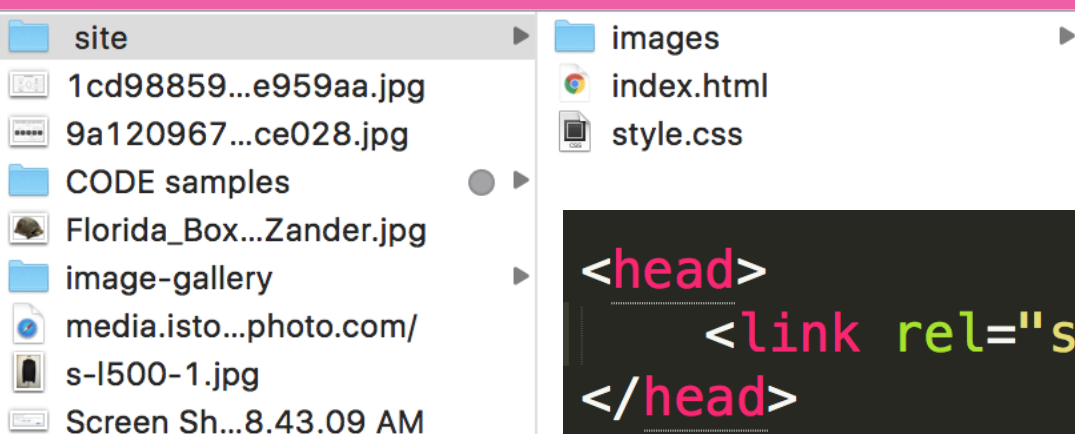
intro to css: 3 ways to link to styles

3. connect CSS to html: **external**

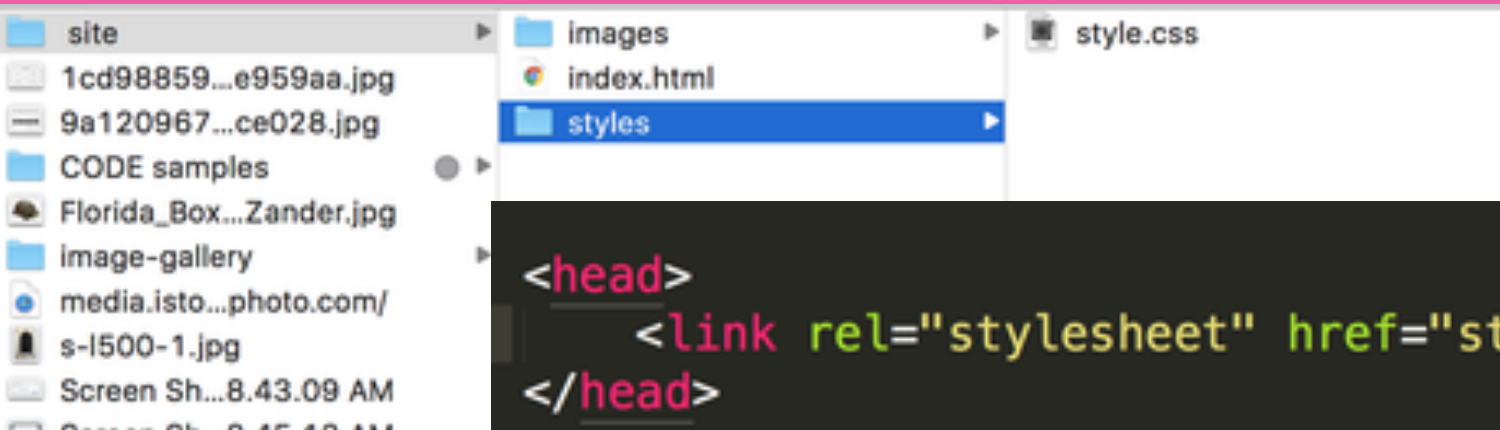
```
<head>  
  <link rel="stylesheet" href="style.css">  
</head>
```

- Can be referenced from multiple pages.
- Reduced file size & bandwidth.
- Easier to maintain in larger projects.

external link - preferred method



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



```
<head>
  <link rel="stylesheet" href="styles/viserion.css"/>
</head>
```

css new vocabulary

CSS for Cascading Style Sheets

Markup code to tell browsers how to style HTML for a web page

DECLARATION

Is the CSS statement that includes the property and value for a selector inside the pair of curly brackets.

Example: `{font-size: 1.2em;}`

SELECTOR

The selected HTML element to apply styling too.

Example: `p` for paragraph is the selector:

`p {font-size: 12px;}`

PROPERTY

The HTML characteristic that can be changed by CSS.


Example: color, font-size, padding.

VALUE

Paired with property, this indicates the setting for the CSS property,

Example: 1.25em, 12px, green, #fff.

Intro to CSS classes and ids



The diagram illustrates the structure of an HTML attribute. A horizontal line is labeled 'ATTRIBUTE' above it. This line branches into two parts: 'NAME' on the left and 'VALUE' on the right. Below 'NAME' is the word 'class' in red. Below 'VALUE' is the string '"important"' in red. These two parts are joined by an equals sign '='.

<p class="important"> Lots of words </p>

In-class exercise

* poem

* travel poster (perfect full background)

Homework

- 404 page (perfect full background)
- memorize 10 selectors
- fix broken code
- Jon Duckett