Introduction to CSS

CISC-2350-R01 | Fall 2017 | Week 5

Ruta Kruliauskaite

Today’s Agenda

- Attendance
- Class reschedule 10/5
- Developer Tools homework show & tell
- Web review presentations
- Introduction to CSS
    - CSS Syntax
    - Inline vs. External CSS
    - CSS Selectors
    - IDs & Classes
- In-class exercise
- Homework assignment

Class reschedule: 10/2 -> 10/5

Thursday, October 5: 4:00-5:15pm

Thursday, October 5, 7:00-8:15pm

Homework show & tell: Developer tools

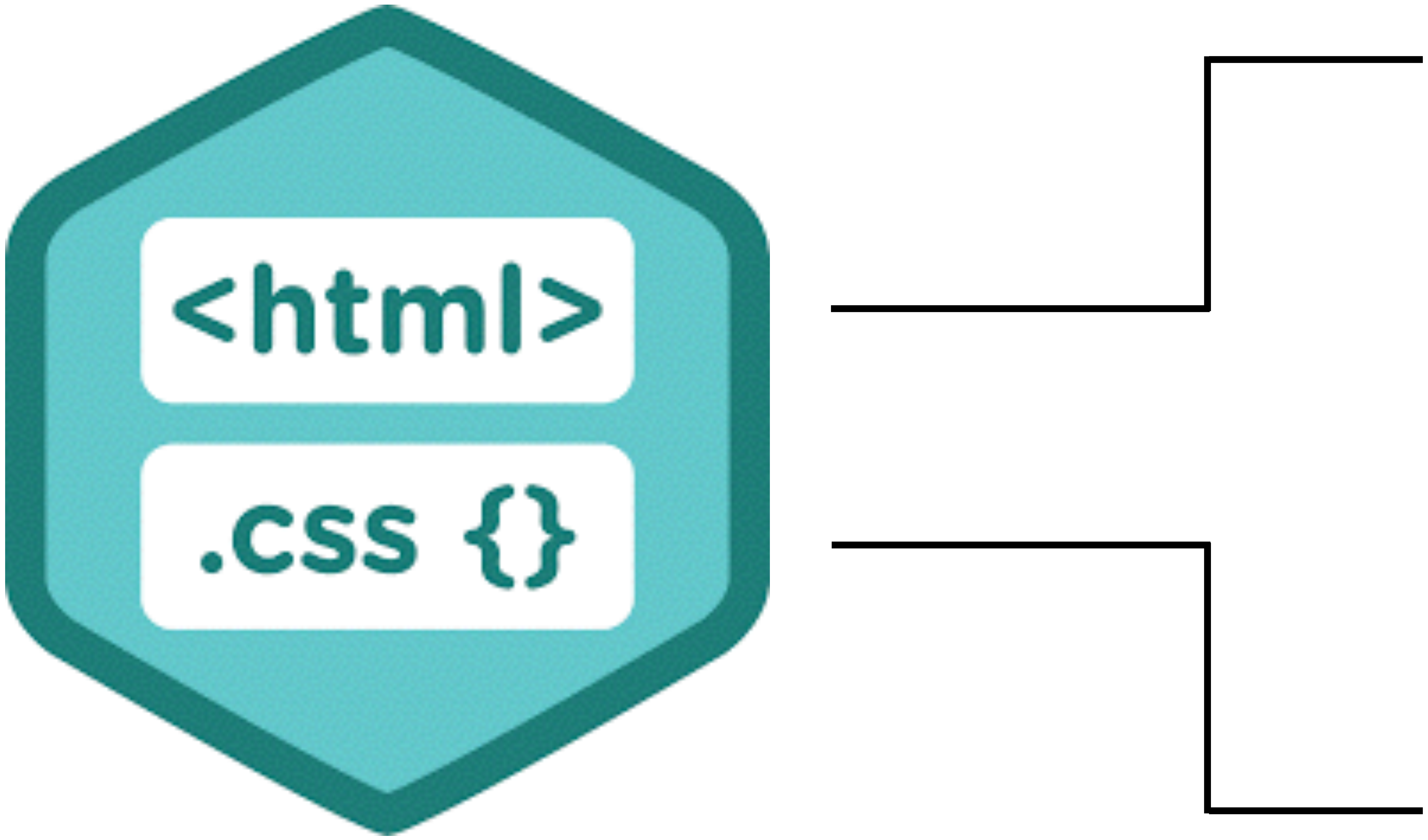
Web review presentations

Introduction to CSS

1. What is CSS?

Cascading Style Sheets (CSS) is a collection of formatting rules that control the appearance of content in a web page.

What is CSS?

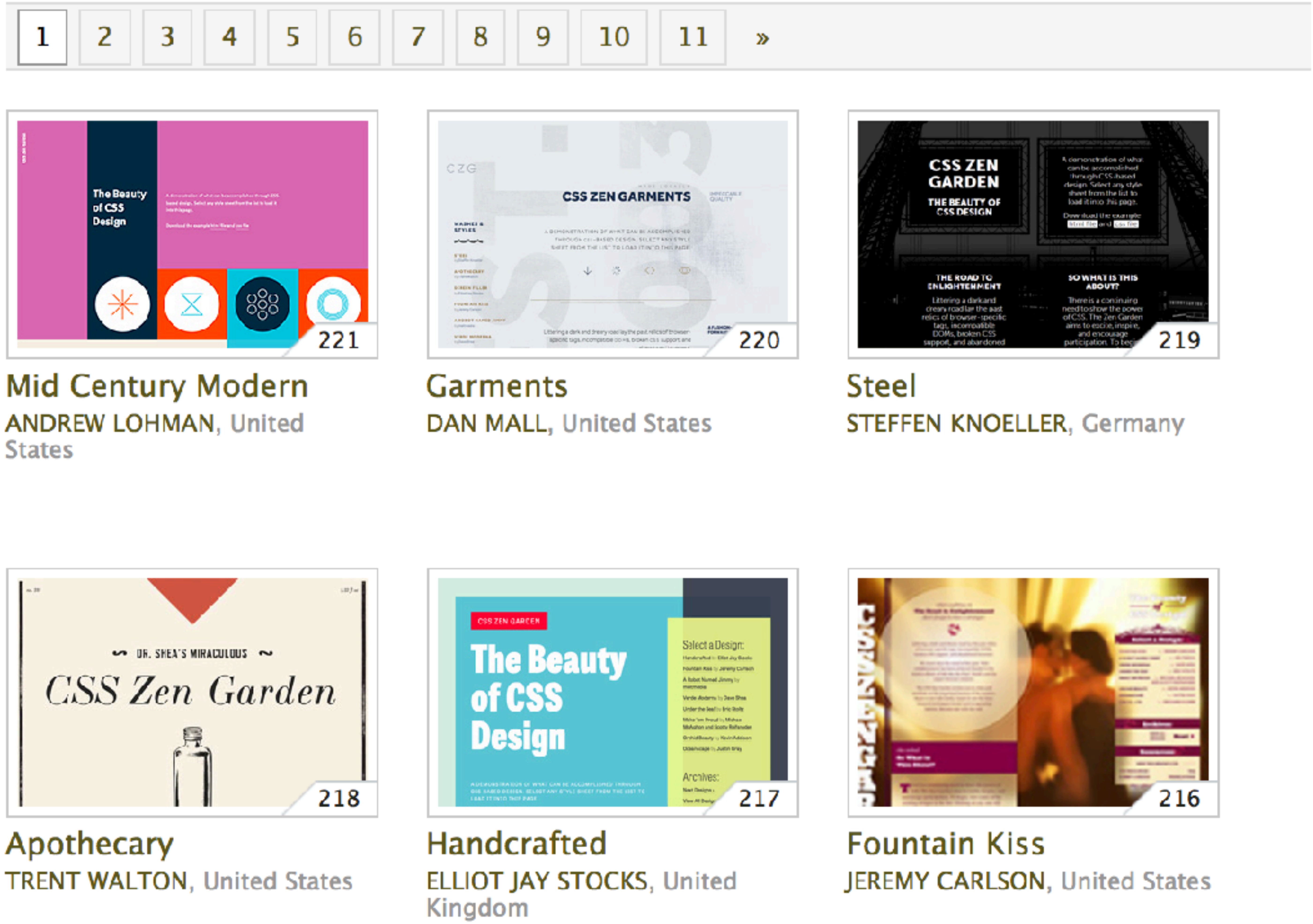


HTML = CONTENT

- Paragraphs
- Divs
- Lists
- Form elements

CSS = STYLE

- Colors
- Fonts
- Layout
- Transitions


1 2 3 4 5 6 7 8 9 10 11 »

221
Mid Century Modern
ANDREW LOHMAN, United States

220
Garments
DAN MALL, United States

219
Steel
STEFFEN KNOELLER, Germany

218
Apothecary
TRENT WALTON, United States

217
Handcrafted
ELLIOT JAY STOCKS, United Kingdom

216
Fountain Kiss
JEREMY CARLSON, United States

Check out the site CSS ZenGarden to see how you can make a page with the same HTML look so different just by changing the CSS

http://www.csszengarden.com/

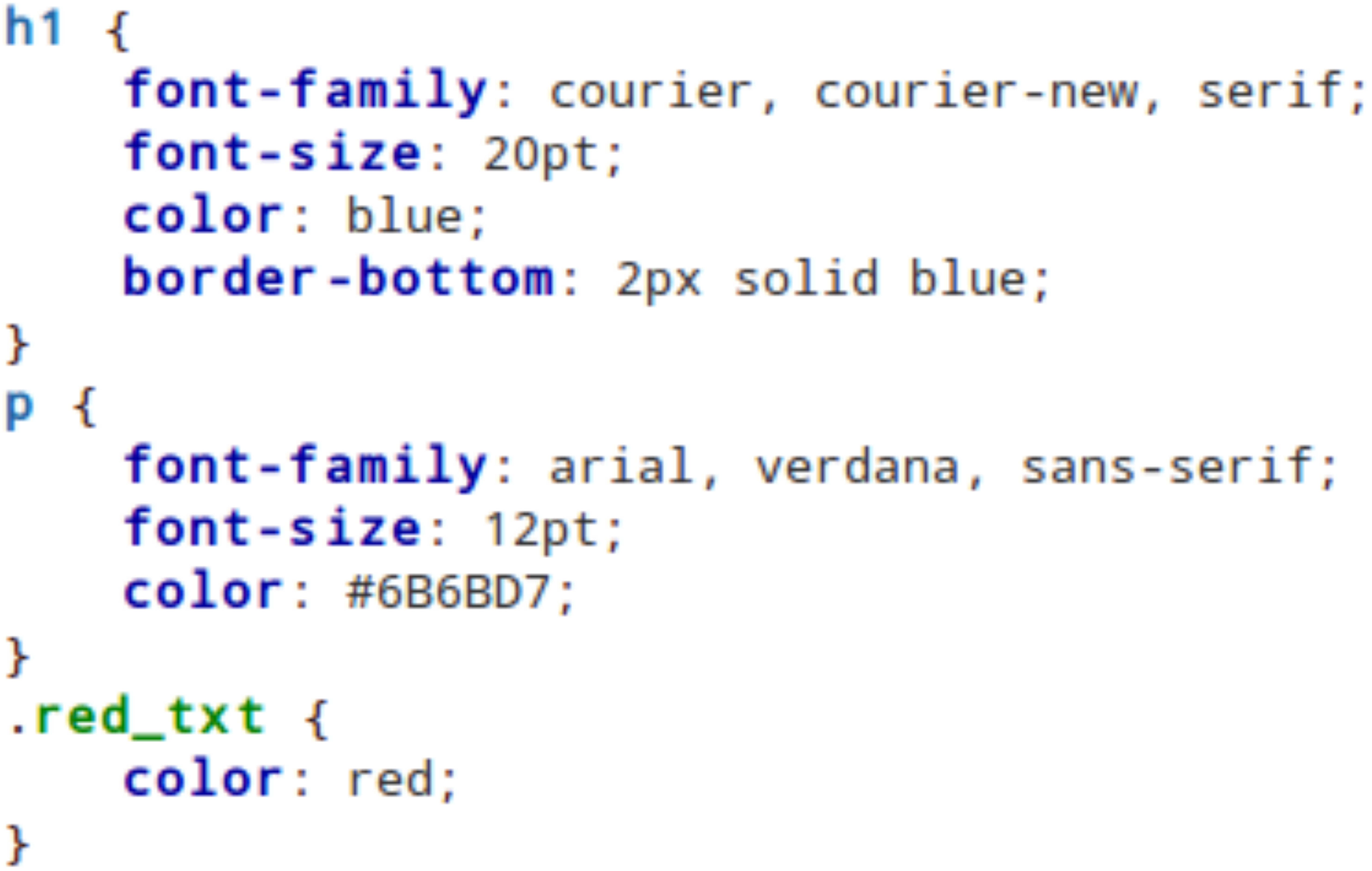
CSS works by associating rules with HTML elements. These rules govern how the content of specified elements should be displayed.

A CSS rule contains two parts: a selector and a declaration.

It takes 5% to learn how to write CSS rule and 95% to learn different properties that you can use.

1. CSS Syntax

CSS Syntax looks like this



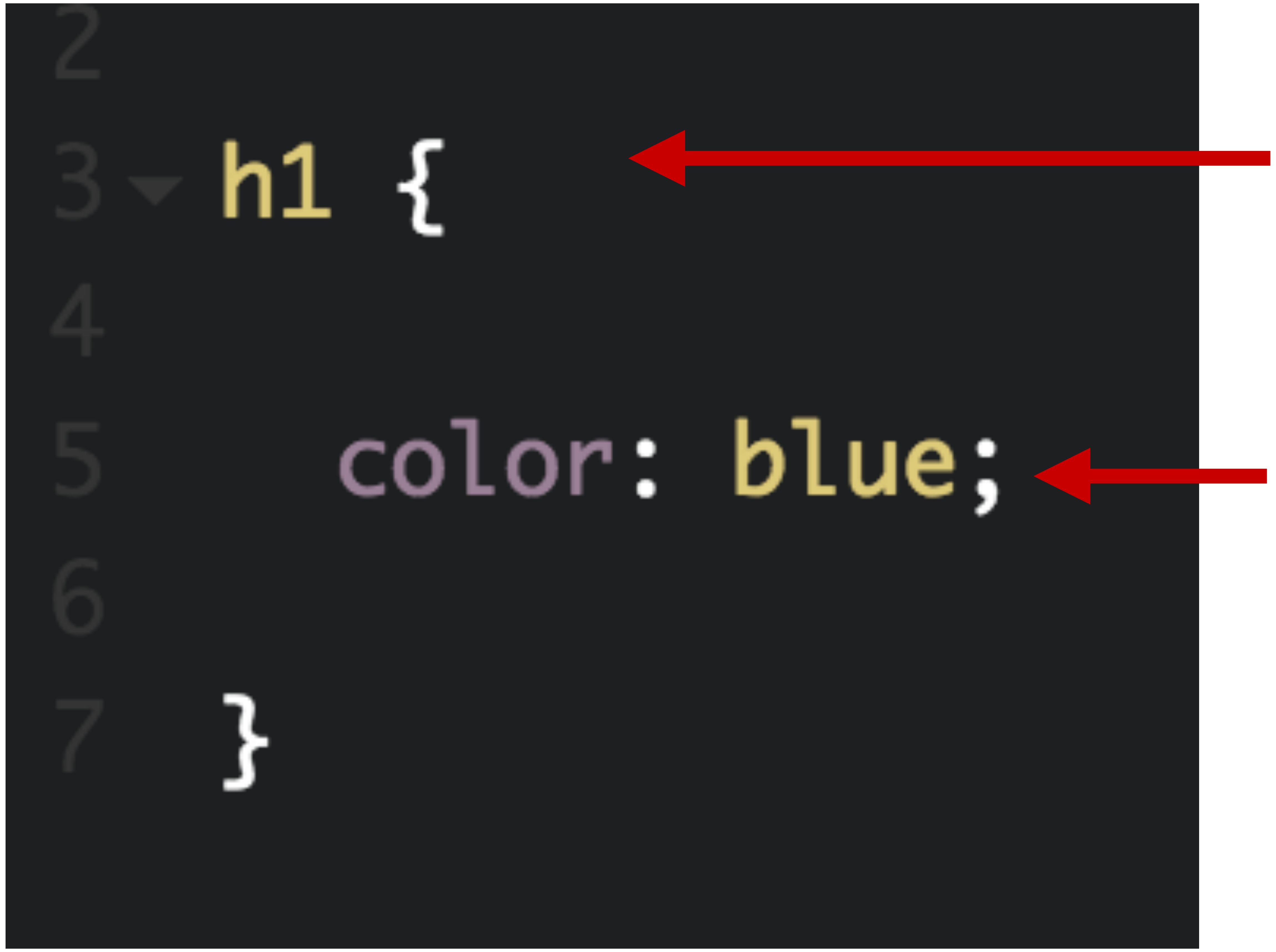
```
h1 {
    font-family: courier, courier-new, serif;
    font-size: 20pt;
    color: blue;
    border-bottom: 2px solid blue;
}
p {
    font-family: arial, verdana, sans-serif;
    font-size: 12pt;
    color: #6B6BD7;
}
.red_txt {
    color: red;
}
```

CSS Syntax is different than HTML.

HTML uses tags and content.

CSS works with selectors and declarations.

CSS Syntax breakdown



```
h1 {

   color: blue;

}
```

H1 = SELECTOR

Selector is a term such as p, h1 that identifies the element you want to format or apply a rule to. Can be multiple.

Color = Property

You're saying which property you want to change, in this case color. Other ones are font-size, background, border, etc.

Blue = Value

You're saying for this property color, give it a value of blue. For color you can also give it an RGB value.

These two together or anything in between curly brackets {} is called a declaration block.

To recap, you write CSS like this:

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

CSS Comments

HTML comments are written like this

```
<!-- This is a comment -->
```

CSS comments are written like this

```
/* This is a comment */
```

2. Where do I put my CSS?

How HTML + CSS work together

We write CSS to separate content (HTML) from presentation (CSS).

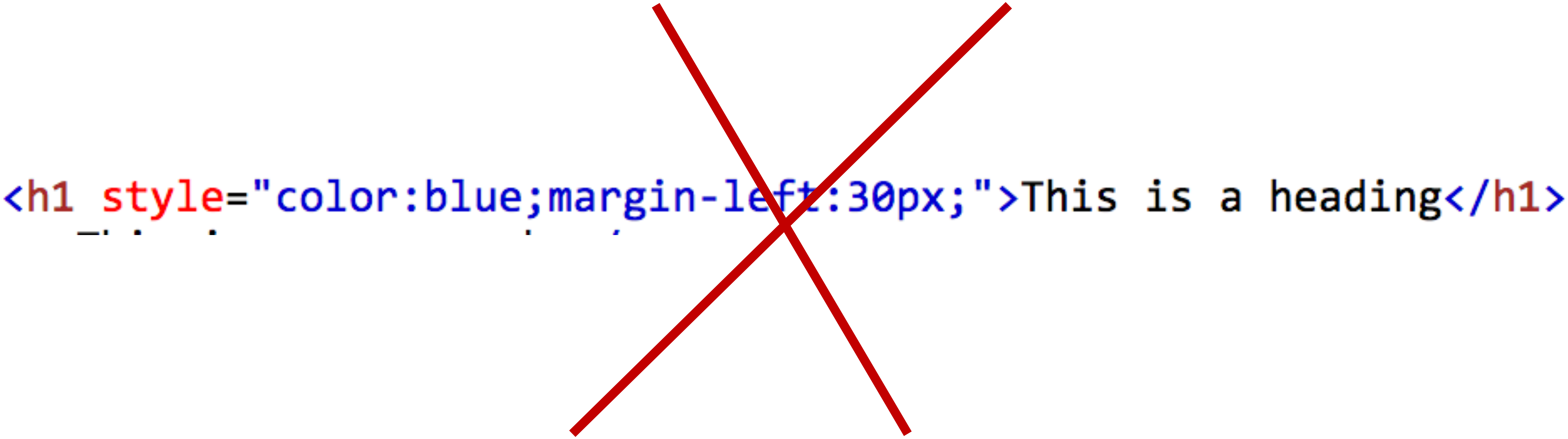
The content of your page—the HTML code—resides in the HTML file, and the CSS rules defining the presentation of the code reside in another file an external style sheet or in another part of the HTML document usually the head section.

Separating content from presentation and having only ONE CSS file means that when we make changes to our styling, we only have to do it in one place. It also results in simpler and cleaner HTML code, and shorter loading times for people viewing your site.

Where do we put our CSS?

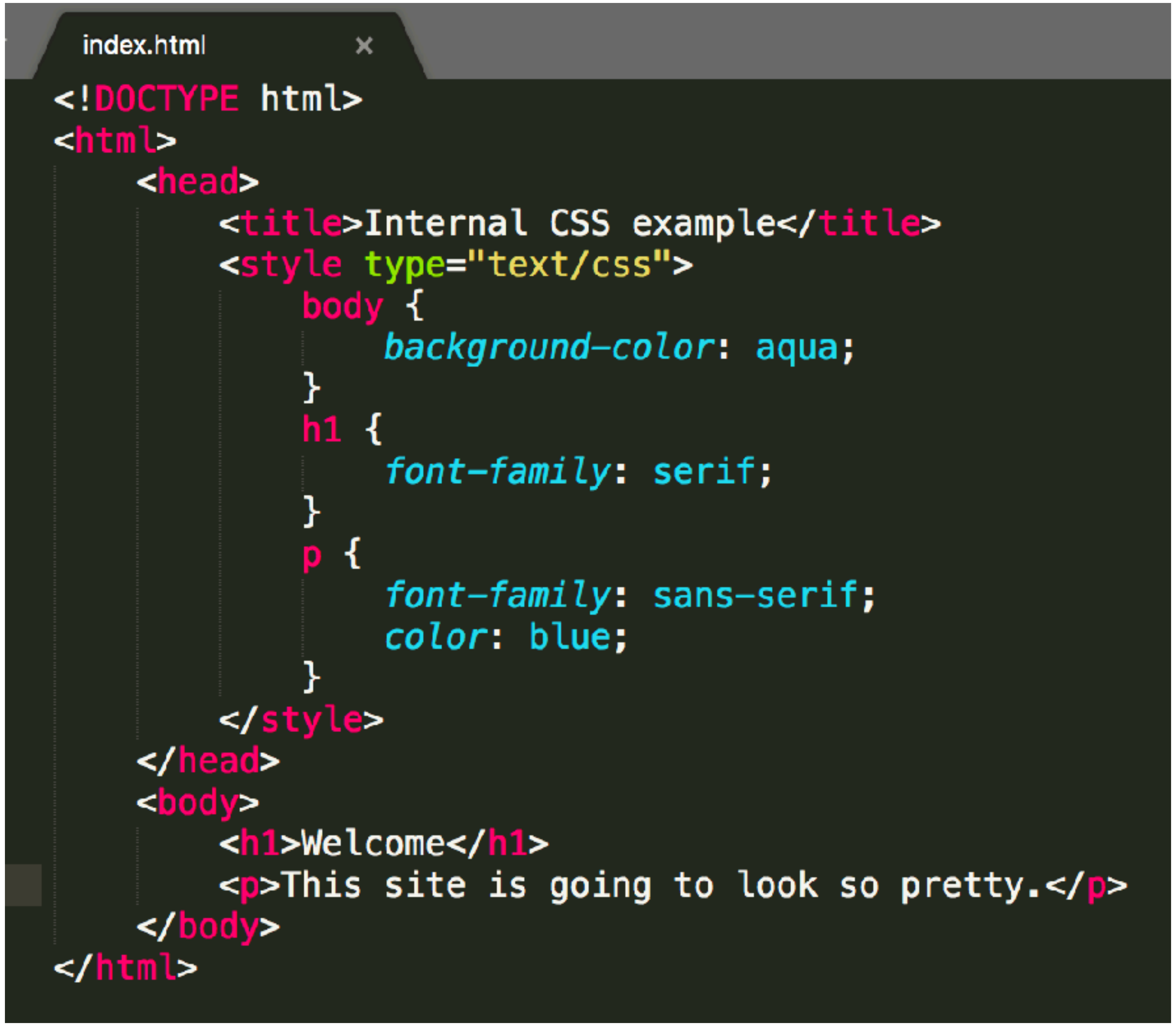
- **CSS can go in three different places**
    1. Right within our HTML tags (referred to as inline CSS)
    2. Right on our HTML page (referred to as internal CSS)
    3. On a separate file within our file structure (referred to as external CSS)
- Though Internal and Inline present their own ease of use by being right within the document we're working on, external CSS is the most common and the convention in Web Development.
- You can have one file (often called style.css) that all of you pages on your site can reference
- Separating the content from the style makes the code easier to read and also allows us to more easily swap out one style for another.

This is what writing CSS within our HTML tags (in-line CSS) looks like, and we will not do this in our class!

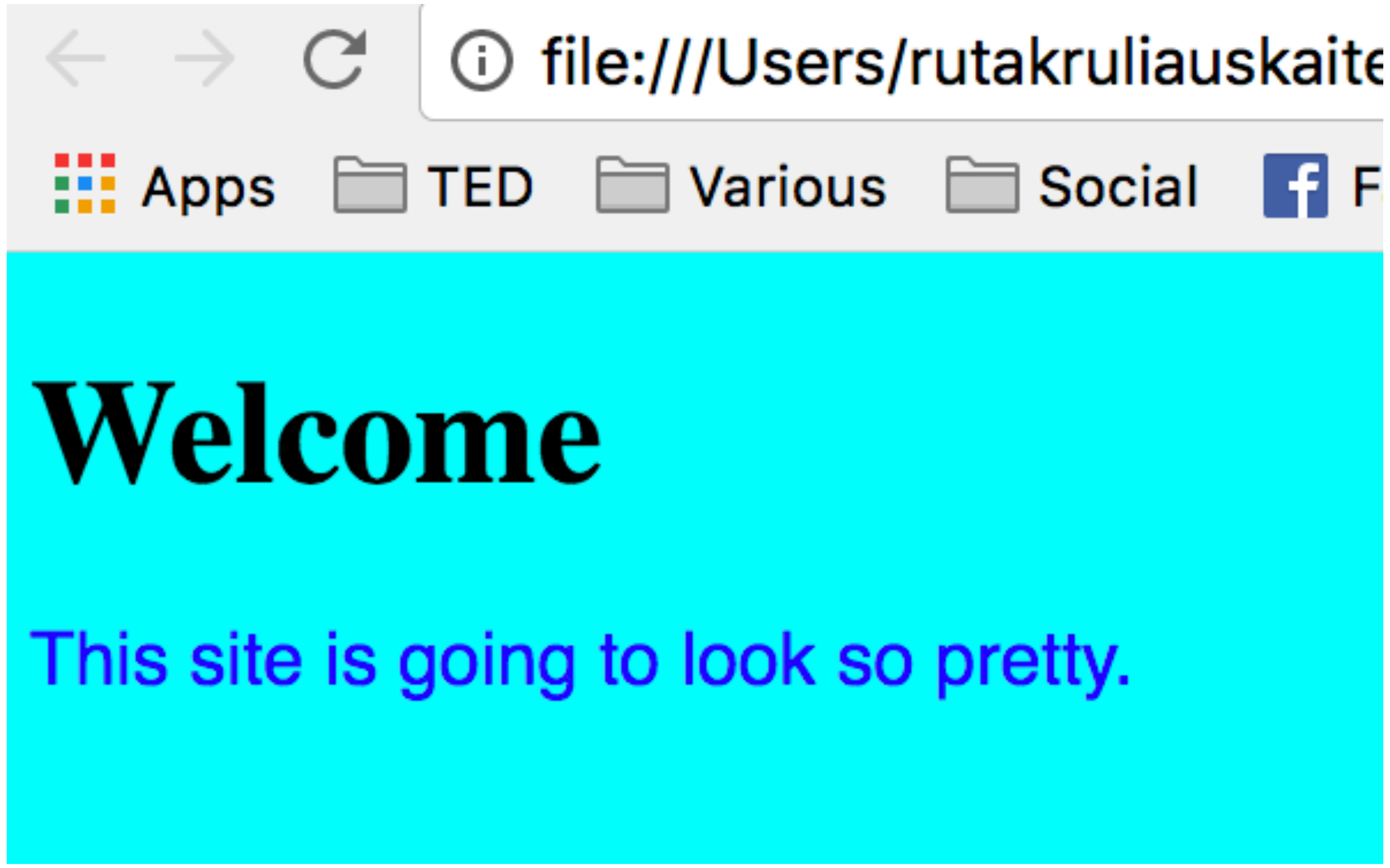


```
<h1 style="color:blue;margin-left:30px;">This is a heading</h1>
```

Example: internal CSS on the page



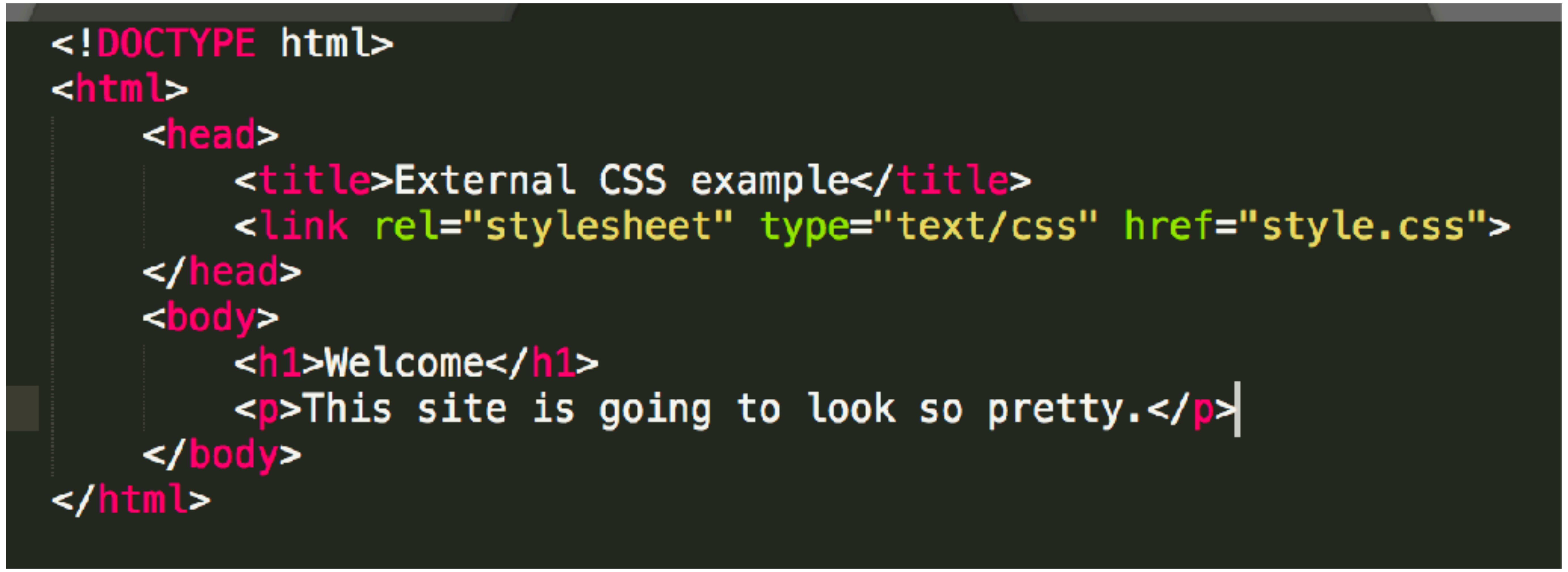
```html
<!DOCTYPE html>
<html>
    <head>
        <title>Internal CSS example</title>
        <style type="text/css">
            body {
                background-color: aqua;
            }
            h1 {
                font-family: serif;
            }
            p {
                font-family: sans-serif;
                color: blue;
            }
        </style>
    </head>
    <body>
        <h1>Welcome</h1>
        <p>This site is going to look so pretty.</p>
    </body>
</html>
```



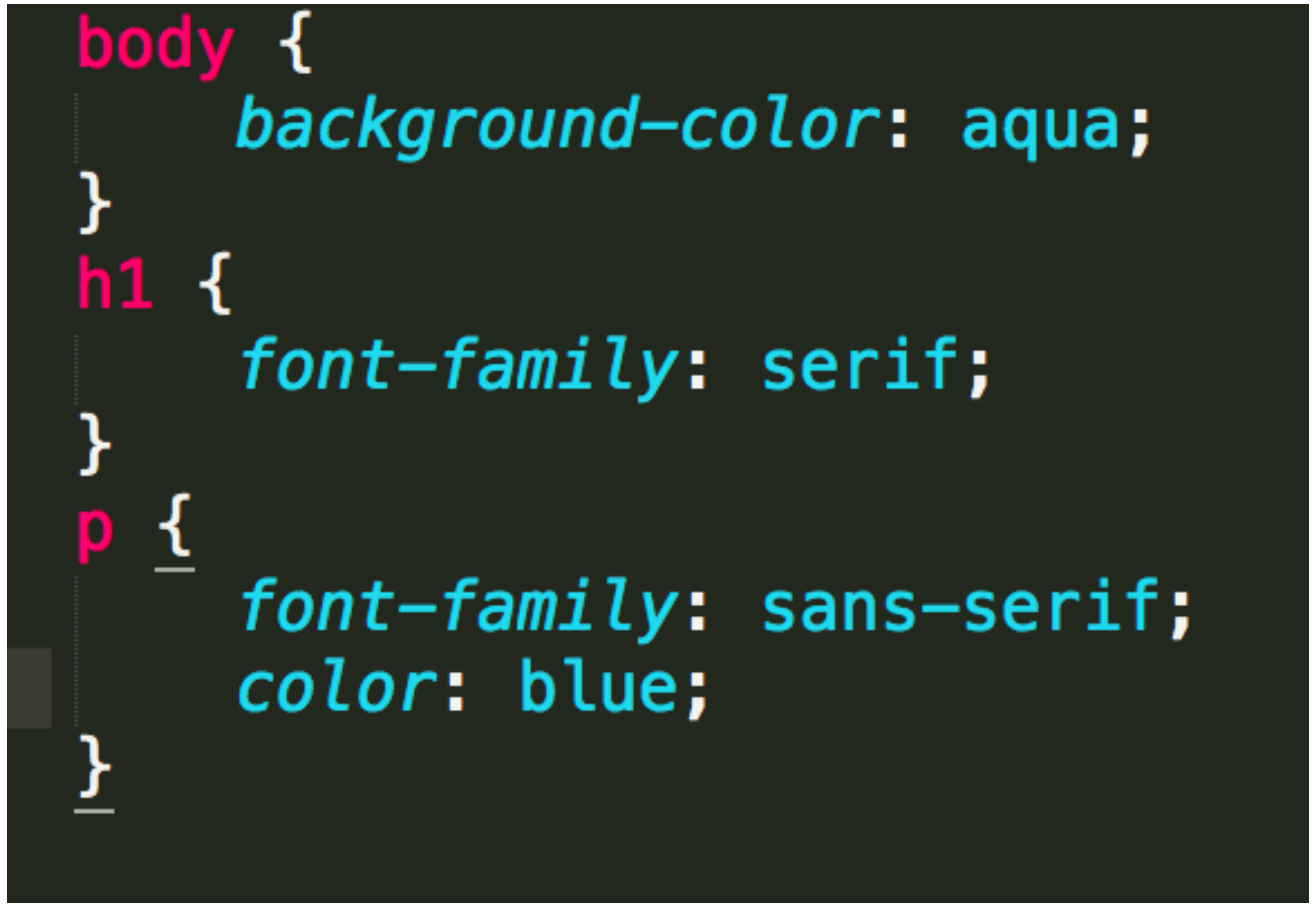
Welcome

This site is going to look so pretty.

Example: external CSS (in a style.css file)



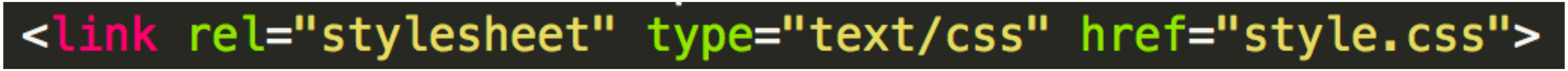
```
<!DOCTYPE html>
<html>
    <head>
        <title>External CSS example</title>
        <link rel="stylesheet" type="text/css" href="style.css">
    </head>
    <body>
        <h1>Welcome</h1>
        <p>This site is going to look so pretty.</p>
    </body>
</html>
```



```
body {
    background-color: aqua;
}
h1 {
    font-family: serif;
}
p {
    font-family: sans-serif;
    color: blue;
}
```

Tells the browser
where to find the
CSS file used to
style the page

Specifies the
path to CSS file



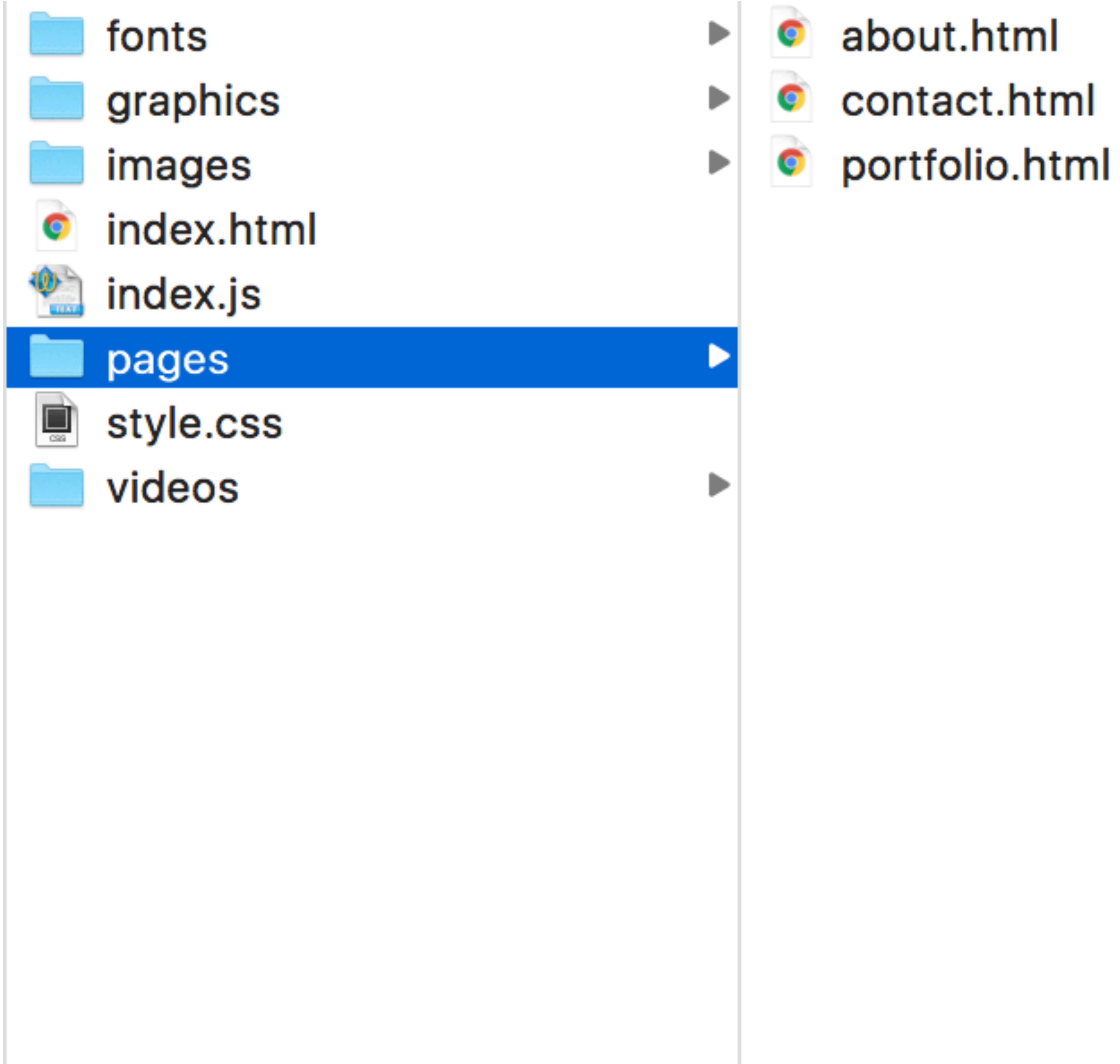
```
<link rel="stylesheet" type="text/css" href="style.css">
```

Specifies the
relationship
between the
HTML page and
the file it is
linked to.

Specifies the
type of
document being
linked to.

Remember file structure

- Remember our file structure?
- Now we also want to have a **style.css** file adjacent to our **index.html** page
- Could also be in its own folder labeled **css**
- All of our pages will reference this one **stylesheet**
- For example, my **about** page's file path would be:



fonts
graphics
images
index.html
index.js
pages
style.css
videos

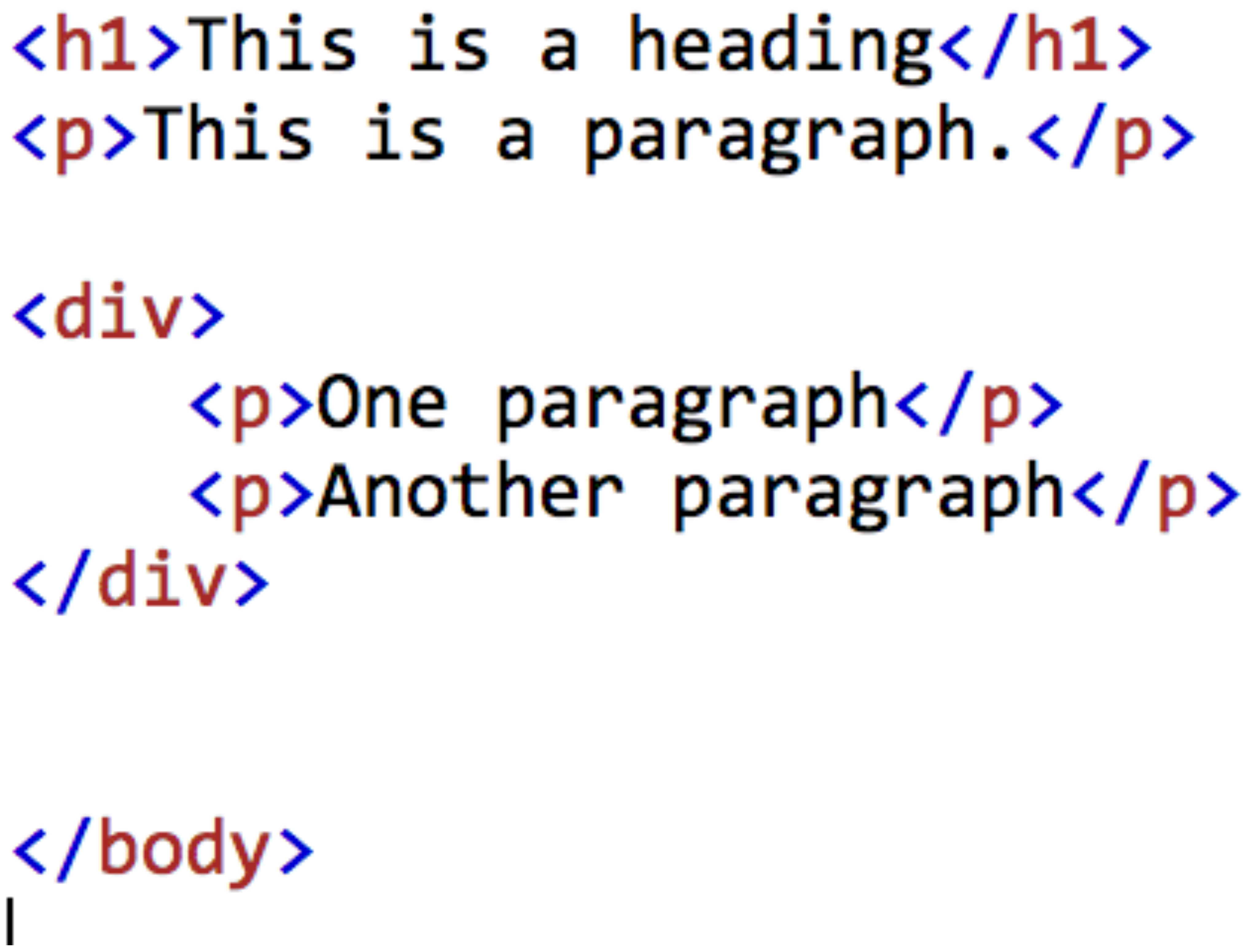
about.html
contact.html
portfolio.html

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

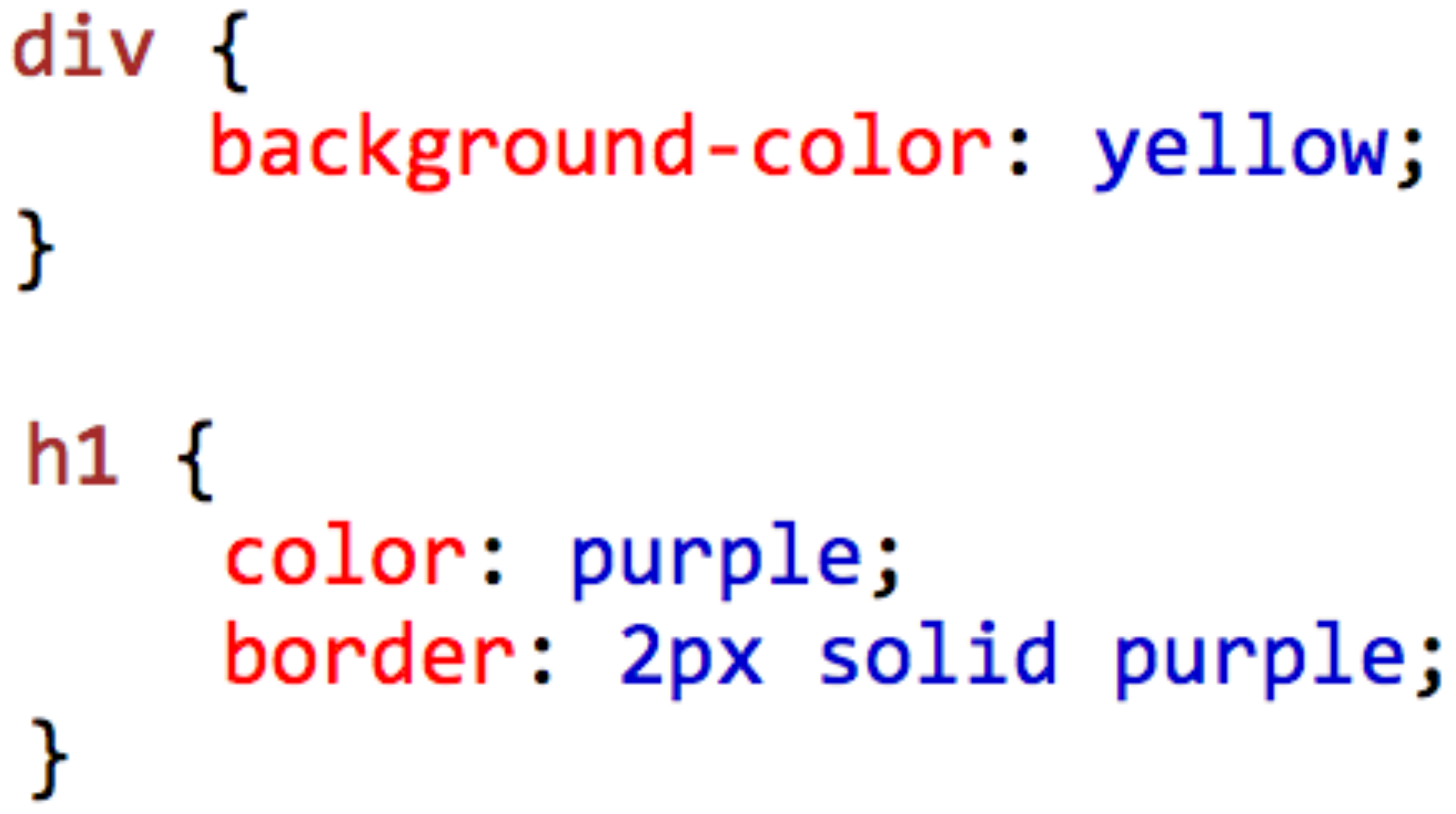
We’ll write both internal and external CSS during this class and will continue only with external later on.

3. CSS Selectors

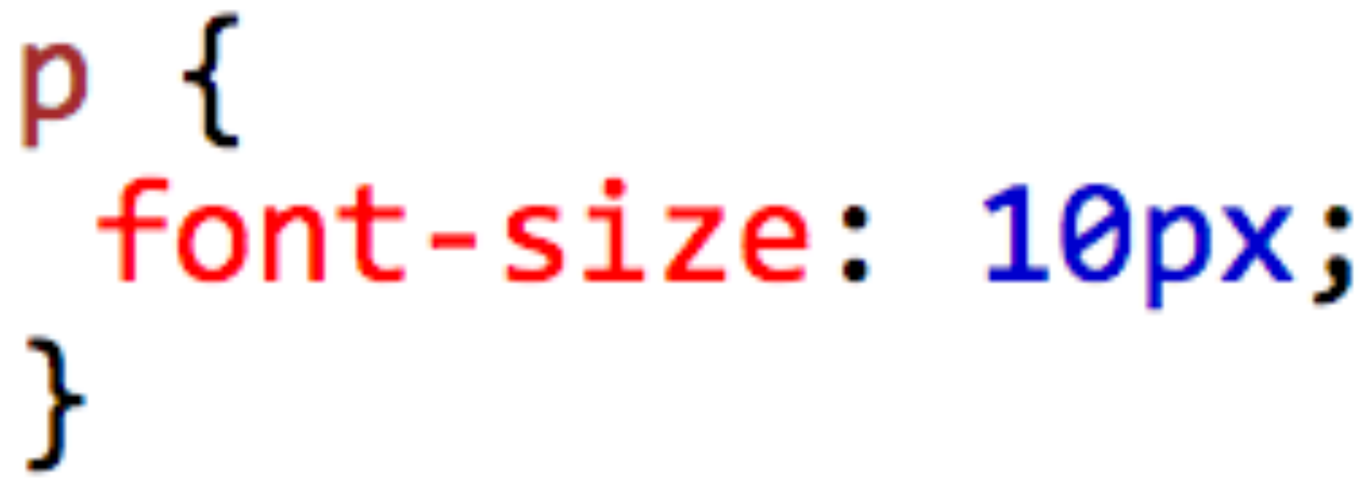
Selector is a term such as p, h1 that identifies the element you want to format or apply a rule to. You can add multiple selectors in a declaration.



```html
<h1>This is a heading</h1>
<p>This is a paragraph.</p>

<div>
    <p>One paragraph</p>
    <p>Another paragraph</p>
</div>


</body>
```



```css
div {
    background-color: yellow;
}

h1 {
    color: purple;
    border: 2px solid purple;
}
```



```css
p {
 font-size: 10px;
}
```

Here are all different selectors

Selector Tye	**Tag Example**	**Meaning**
Universal Selectors	*{}	Applies to full document
Type Selectors	h1, h2, h3 {}	Targets individual elements
Class Selector	.note {}	Targets all classes with value **note**
ID Selector	#name {}	Targets only one ID with value **name**
Child Selector	li>a {}	Targets all anchor tags within lists
Descendent Selector	pa {}	Targets all anchor tags in paragraphs, even if there’s other elements between them

CSS Properties

There're so many different properties: **http://www.htmldog.com/references/css/properties/**

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

What are the “properties” you can use?

- We’ve talked about font-size, color and background-color
- There’s a whole list for you to look at the different properties: **it’s almost infinite!**
- This week we’ll focus on some of the basic properties:
  - background-color
  - border
  - text-align (and other text-properties)
  - text-decoration (for underlines and strikethroughs)
  - font-size
  - color to change text color

What does Cascading mean?

- CSS Cascading has to do with how the styles apply when you have two or more rules that apply to the same element? Which one takes precedence?
- CSS Rules will cascade towards specificity and then downwards on the page. For example, if you set text to be a certain size in your body { } tag, that rule will cascade all the way down to all of the elements within body
- This all comes back to the tree structure of HTML, remember?
- But: if you do something to a body tag, then do something to lists within the body, the latter will take precedence
- If you do two things to those lists, the last one on your CSS page will take precedence

More on Cascading

From John Duckett book:

Last rule
If the two selectors are identical, the latter of the two will take precedence. For example, if there were two **i** elements in style sheet, the second one would take precedence over the first.

Specificity
If one selector is more specific than the others, the more specific rule will take precedence over more general ones. For example, h1 is more specific than body tag and so one.

Important
You can add **!important** after any property value to indicate that it should be considered more important than other rules that apply to the same element.

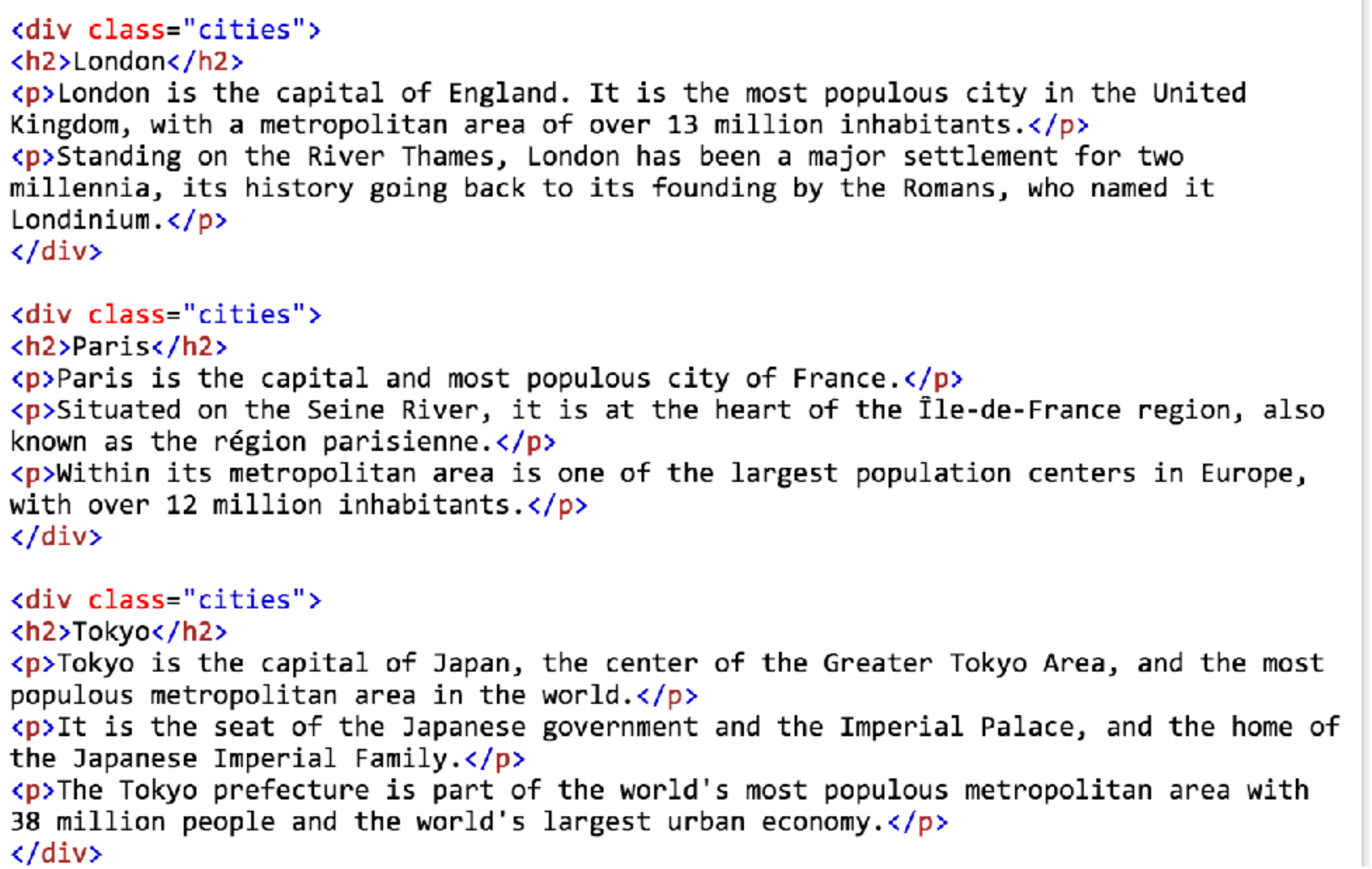
4. Classes and IDs

Classes and IDs

- Two common attributes used to single out certain HTML elements are "class" and "id", both of which are used to identify particular elements when adding **CSS rules**
- Class and id names can be anything you like, they have no particular meaning in themselves. The are used more like labels or selectors in CSS . Use a class when you have more than one element you want to share the same class (maybe you want to style them all the same, for example).
- Use an id when there is only one element on the page with that id, for example id=”header
- With a class you can have as many as you like
- An element can have more than one class, but not more than one id. When there is more than one class, the class names are separated by spaces.

Classes

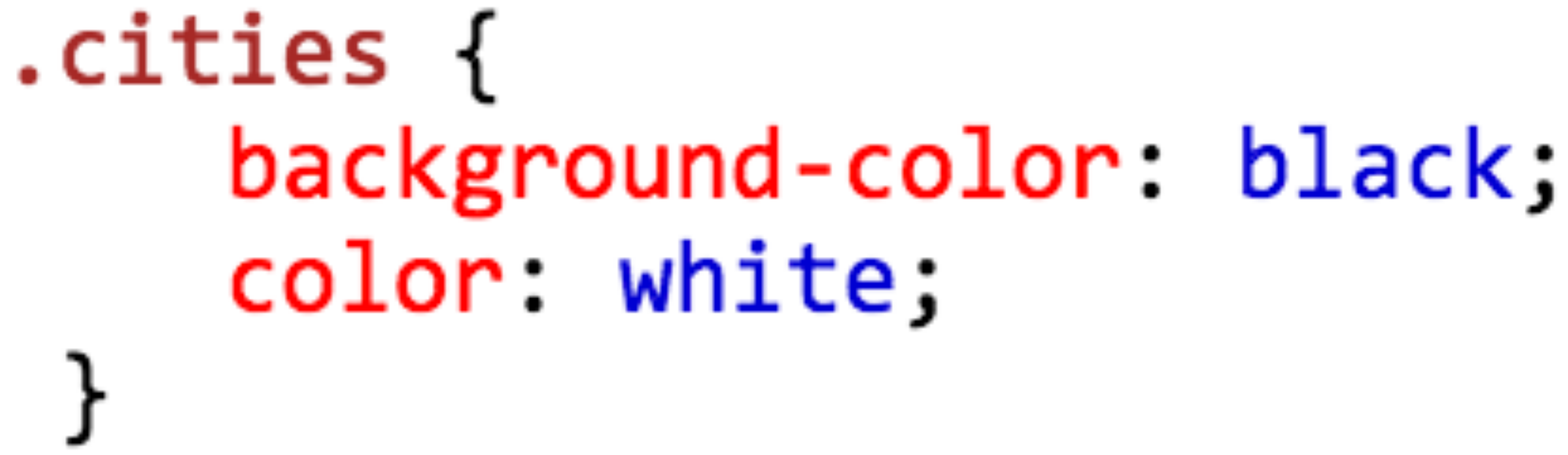
Every HTML element can also carry a class attribute. Sometimes, rather than uniquely identifying one element within a document, you will want a way to identify several elements as being different from the other elements on the page.



```html
<div class="cities">
<h2>London</h2>
<p>London is the capital of England. It is the most populous city in the United
Kingdom, with a metropolitan area of over 13 million inhabitants.</p>
<p>Standing on the River Thames, London has been a major settlement for two
millennia, its history going back to its founding by the Romans, who named it
Londinium.</p>
</div>

<div class="cities">
<h2>Paris</h2>
<p>Paris is the capital and most populous city of France.</p>
<p>Situated on the Seine River, it is at the heart of the Île-de-France region, also
known as the région parisienne.</p>
<p>Within its metropolitan area is one of the largest population centers in Europe,
with over 12 million inhabitants.</p>
</div>

<div class="cities">
<h2>Tokyo</h2>
<p>Tokyo is the capital of Japan, the center of the Greater Tokyo Area, and the most
populous metropolitan area in the world.</p>
<p>It is the seat of the Japanese government and the Imperial Palace, and the home of
the Japanese Imperial Family.</p>
<p>The Tokyo prefecture is part of the world's most populous metropolitan area with
38 million people and the world's largest urban economy.</p>
</div>
```

Classes

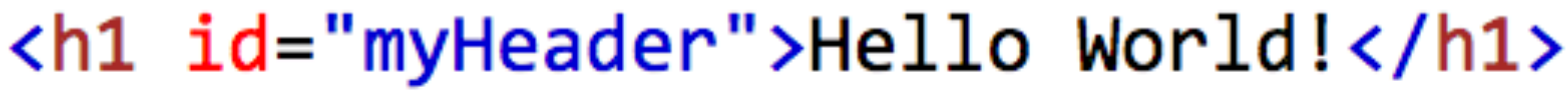
To select these classes in CSS you would do so with .cities syntax



```
.cities {
    background-color: black;
    color: white;
 }
```

IDs

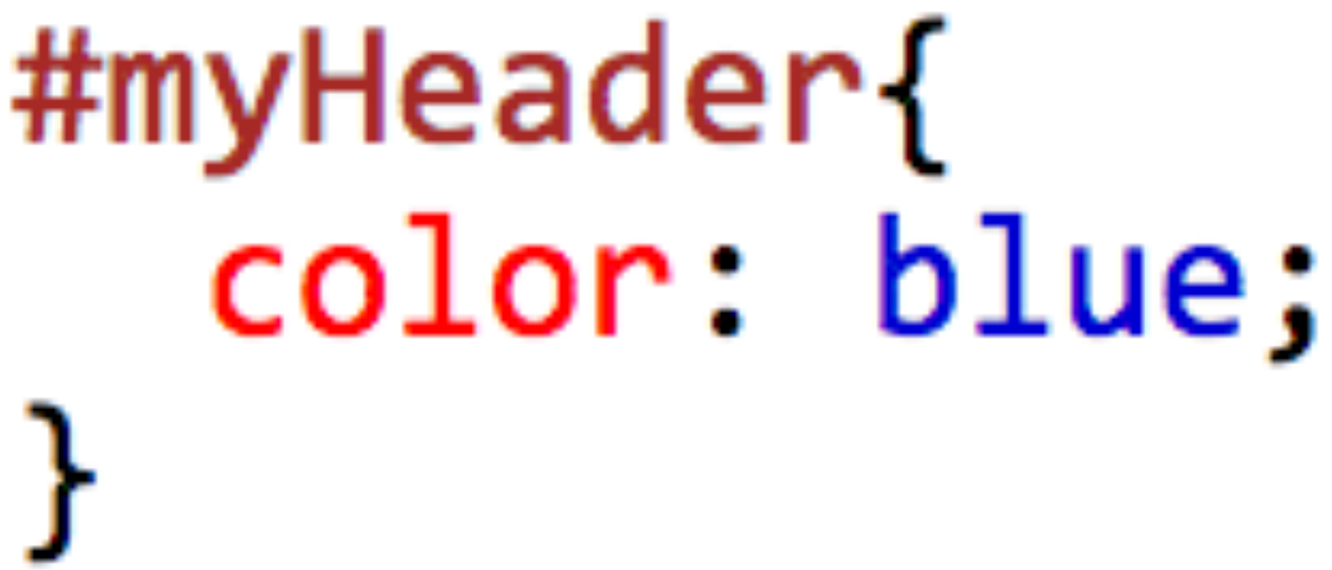
Every HTML element can carry the id attribute. It is used to uniquely identify that element from other elements on the page. Its value should start with a letter or an underscore (not a number or any other character). It is important that no two elements on the same page have the same value for their id attributes (otherwise the value is no longer unique).
More to read on ID naming: https://mathiasbynens.be/notes/css-escapes



```
<h1 id="myHeader">Hello World!</h1>
```

IDs

To select these IDs in CSS you would do so with #myHeader syntax



```
#myHeader{
  color: blue;
}
```

Classes and IDs Summary

- We touched on classes and IDs briefly early on with our HTML elements (linking)
- Now that we can pinpoint what we want to style, they become useful because we can put anything in them and style them how we want.
- <div> is the most common to use with them, but can be other elements
- Classes can be used many times
- IDs are only used once

In-class exercise

Adding a bit of style and color to the html file on github.

Introduction to midterm

Midterm Assignment (due 10/12)

Your midterm project will be a multi-page (at least 3) website responding to one of the following assignment prompts:

- An conceptual, interactive self portrait. Make a digital interpretation of a portrait that has come before you, from painting to photography. This project is not intended to be a portfolio but some kind of creative representation of yourself and who you are that cannot be experienced in another way.

- A piece of hypertext narrative or art, similar to one of these examples **https://rhizome.org/art/artbase/collections/collection-hypertext/**

- Invent a fictional or futuristic product (a machine that could record your dreams, a sneaker that plays music, get creative!) and create a product website for it. Look here for some inspiration **http://www.webdesign-inspiration.com/web-designs/type/product**

It's an individual project.

Midterm Assignment 2

Your midterm project will be a multi-page (at least 3) website responding to one of the 3 assignment prompts (or one of your own, approved by me).

You will give a five minute presentation on your midterm to the class, explaining your idea, your code, your visual choices and your process. (More questions to guide your presentation will be provided).

You will be graded according to a rubric that I will distribute next week on concept, execution, visual design and effective communication.

Midterm presentations: Thursday, October 12. There might be guest critics.

Midterm Assignment 3

Your midterm project will include the following:

- Must be posted to Github pages in proper file structure.
- Your code should be neat and well organized, as well as well commented out
- The HTML and CSS code is efficient (no duplications), with CSS in an external style.css file

Required web elements include:

- Minimum of three HTML pages
- Media elements (video, audio, iframes)
- Appropriate usage of HTML elements: text (h, p, ul, ol, li), lists, links, images, div, span
- Use of Classes and IDs in your HTML and as CSS selectors
- CSS color selections
- CSS font selections

Homework assignment

Homework

Due Wednesday by 6pm

1. **Read:**
   - This post about Classes and IDs
2. **CSS tutorial:** do lesson 1 of *Learn CSS* course on Codecademy (CSS Selectors and Visual Rules -> CSS Setup and Selectors & CSS Visual Rules)
   - Post what new you learned and any questions you may have on #general channel on Slack
3. **Come up with a concept for your midterm assignment** (based on the prompts on the previous page). Each of you will have 1min to present your idea during next class. For this post on #general channel on Slack:
   - A paragraph on what you are planning to make for the assignment. This can include a short description, links to references or things that inspired you, etc.
   - One slide that layouts 1min of your presentation (can be text, image, etc.). I'll load them all from my computer, so it's faster to present.
4. **Apply some CSS to your multiple pages homework from last week as follows:**
   1. Create a new folder called Week5-1
   2. **Copy over** your files from last week into that folder
   3. Create a new CSS file and link it in the head of your HTML, following the instructions
   4. Use at least 3 selectors in your homework. For example, you can change the background colour of the page, make text a different color, add a border around some elements. Feel free to experiment.
5. **After you're done, post two links on #general channel on Slack by 6pm on Wednesday:**
   - link to your project repository on Github
   - link to Github page for it
6. **Office hours:** today 6:45-7:45pm. Sign up **here**.