More HTML Tags and Attributes / Intro to CSS

Introduction to Web Design and Development

CSCI 1210

*Based upon notes created by Stephen Hendrix

HTML

A Little More HTML

Let's talk about HTML a little more

Then, we'll jump into Cascading Stylesheets (CSS)

There are a couple of important attributes we need to be aware of:

`id`

&

`class`

A Little More HTML

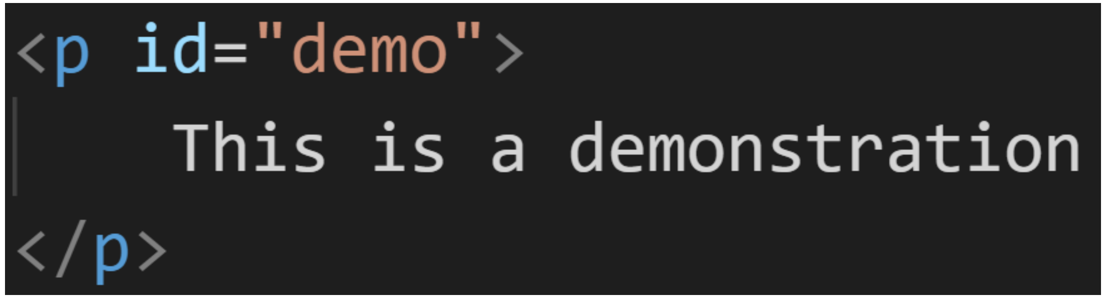
They are similar, but there are important differences

id

As its name implies, assigns an identification value to an element

It can be accessed later for styling (CSS) or scripting (JavaScript)

Value must be unique on a given page



```html
<p id="demo">
    This is a demonstration
</p>
```

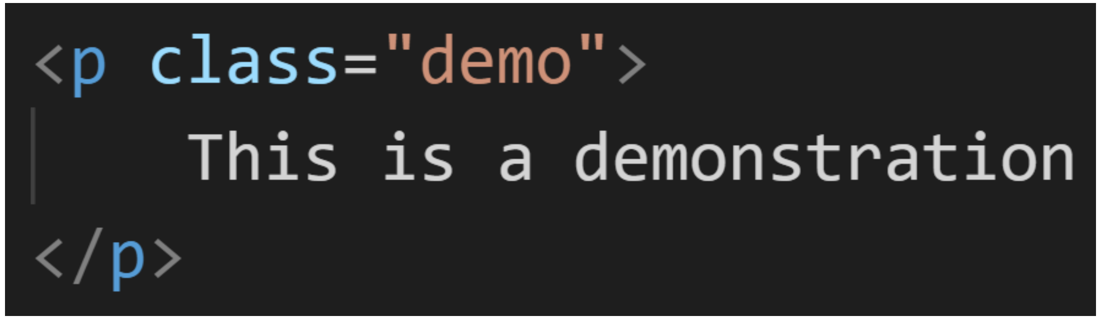
A Little More HTML

They are similar, but there are important differences

`class`

Allows us to identify a group of similar elements so that we can manipulate those elements selectively

Assigns a value to one or more elements that can be accessed later



```
<p class="demo">
    This is a demonstration
</p>
```

A Little More HTML

Similarities

The value you assign to an **`id`** or a **`class`** can be most anything, as long as it starts with a letter or an underscore

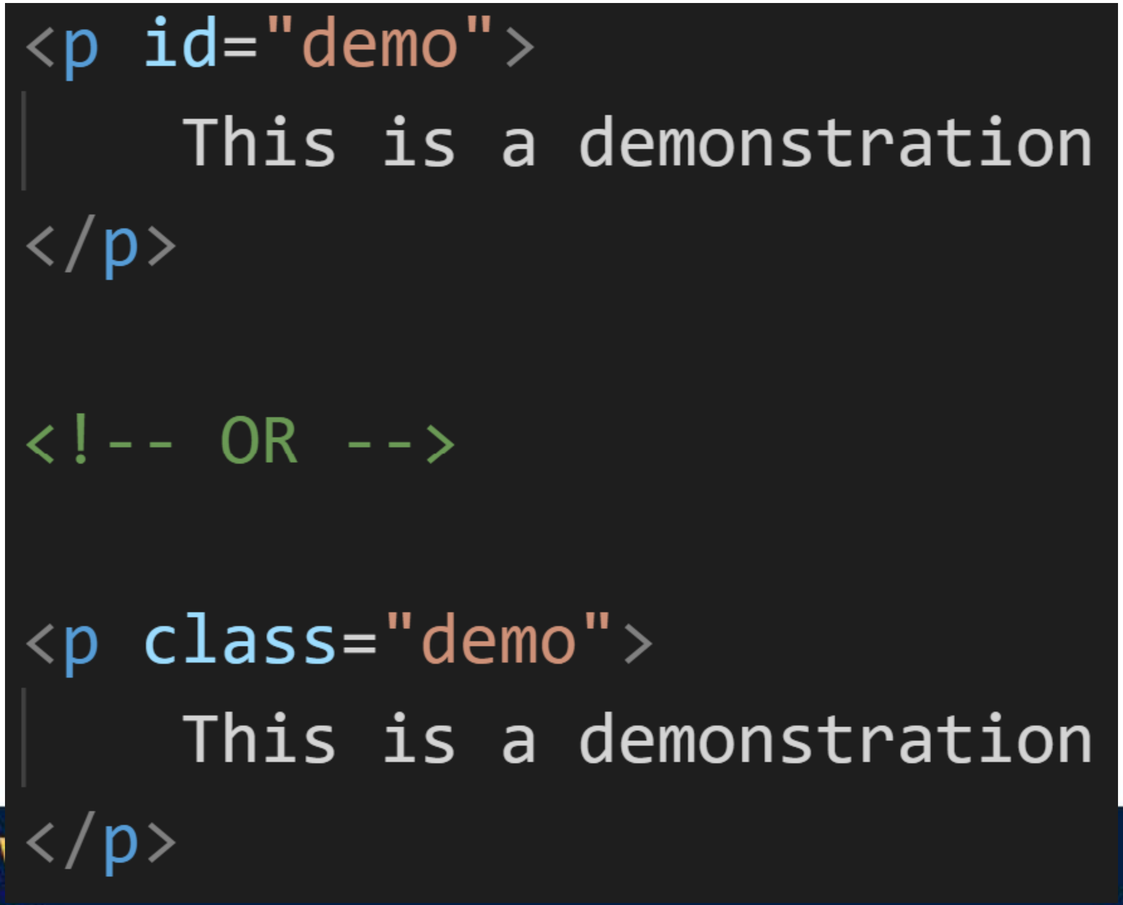
Other Errata

We'll see later how we can use classes to customize our styling

We can use ids also, but we use them more with JavaScript...which is a bit beyond the scope of this class (but, feel free to enroll in CSCI 1720 - Intermediate Web!)

Syntax / Use

Assigning a class or id to an element simply requires adding the appropriate attribute to the element’s opening tag

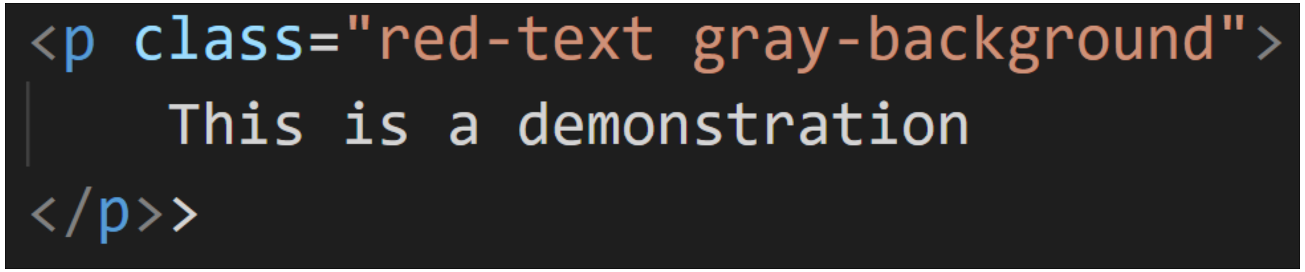


```html
<p id="demo">
    This is a demonstration
</p>

<!-- OR -->

<p class="demo">
    This is a demonstration
</p>
```

Multiple Classes

We can assign multiple classes to an element to further specify its presentation

For example, let’s say we have classes defined to display text as red (named `red-text`) and to change the element’s background color to light gray (named `gray-background`)



```
<p class="red-text gray-background">
    This is a demonstration
</p>>
```

👍 Notice that the list of classes is space-separated. We can use as many classes as we need (not just two)

👍 We’ll discuss the mechanics of ‘defining’ classes soon

The div Element

Allows you to group a set of elements together in one block level box (container)

Typically used to define a web page's structure, or layout

Often nested

Page layout is the part of graphic design that deals in the arrangement of visual elements on a page

It generally involves organizational principles of composition to achieve specific communication objectives. (More on that later)

Layout

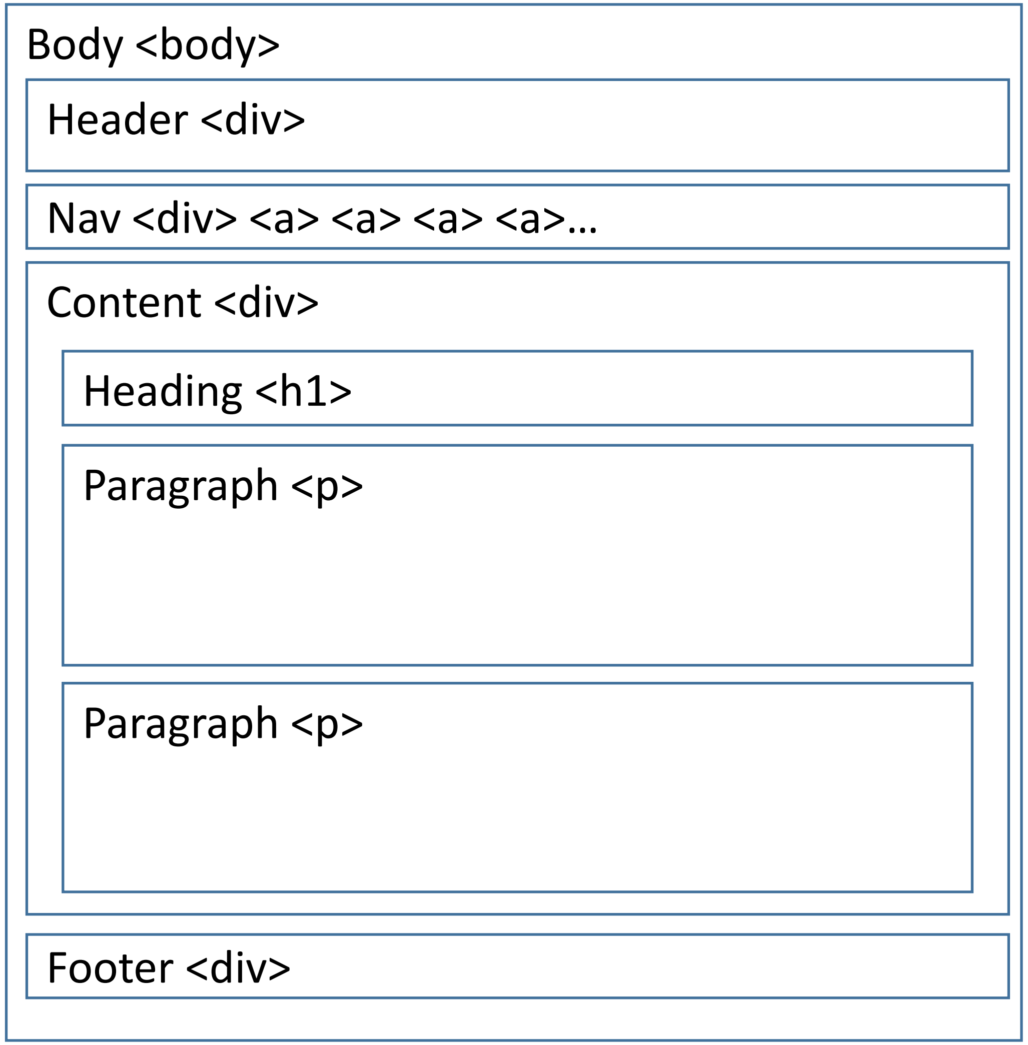
Layout

There are many page layout options

This is a basic, one column layout

Each page still has a header, navigation bar, content section, & footer

The content section is subdivided into its own sections, based on the page's actual content



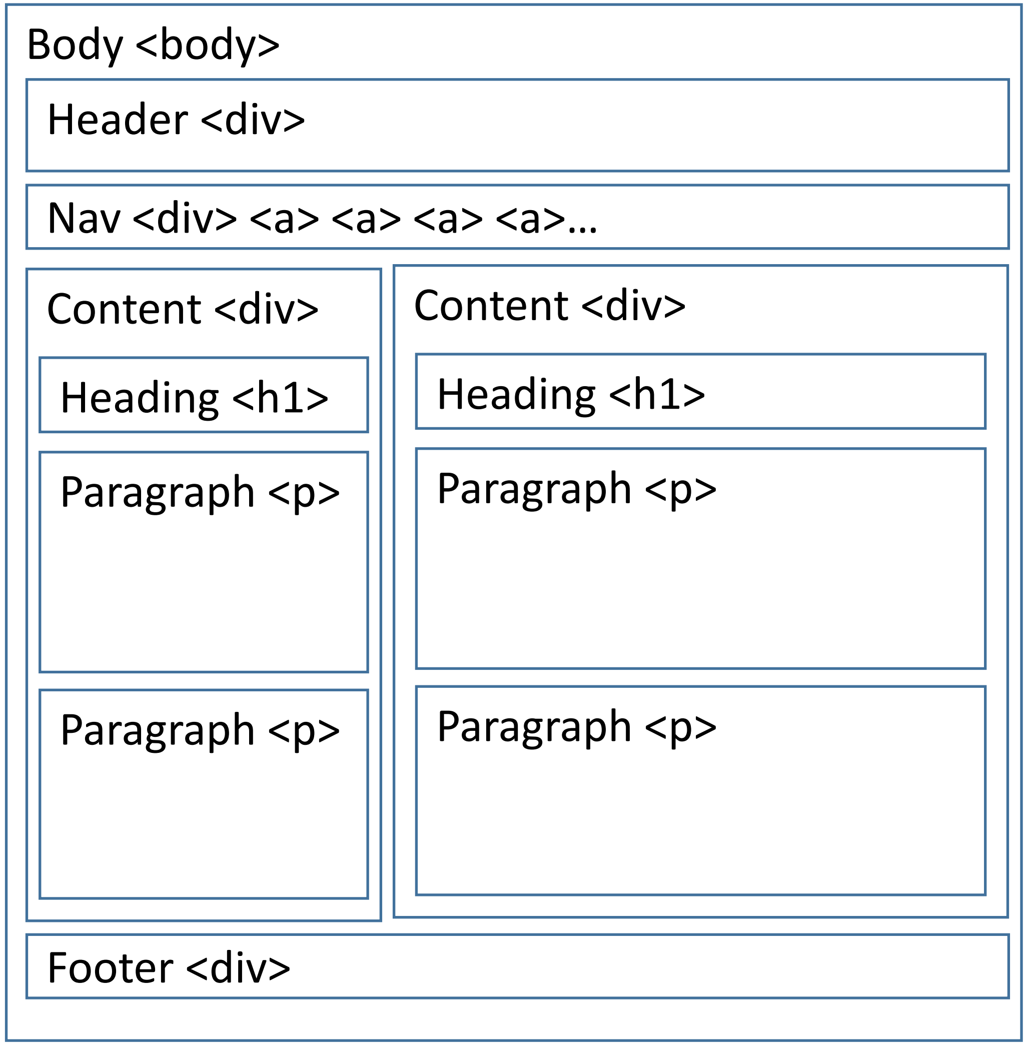
Layout

There are many page layout options

This is a basic, two column layout

Each page still has a header, nav bar, & footer

But here we have two content sections which display as two columns. One could be a "main content" column and the other an "aside" column



Body <body>

Header <div>

Nav <div> <a> <a> <a> <a>…

Content <div>

Heading <h1>

Paragraph <p>

Paragraph <p>

Content <div>

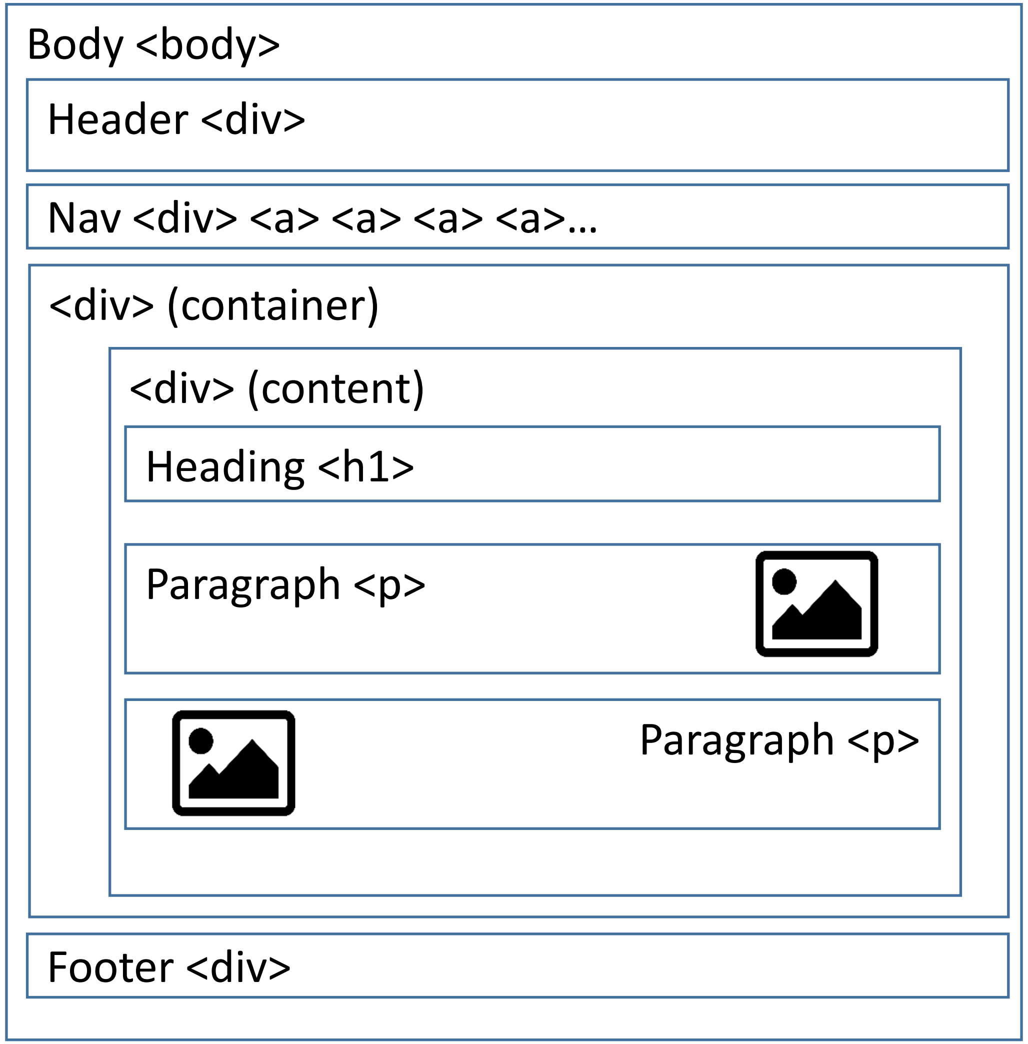
Heading <h1>

Paragraph <p>

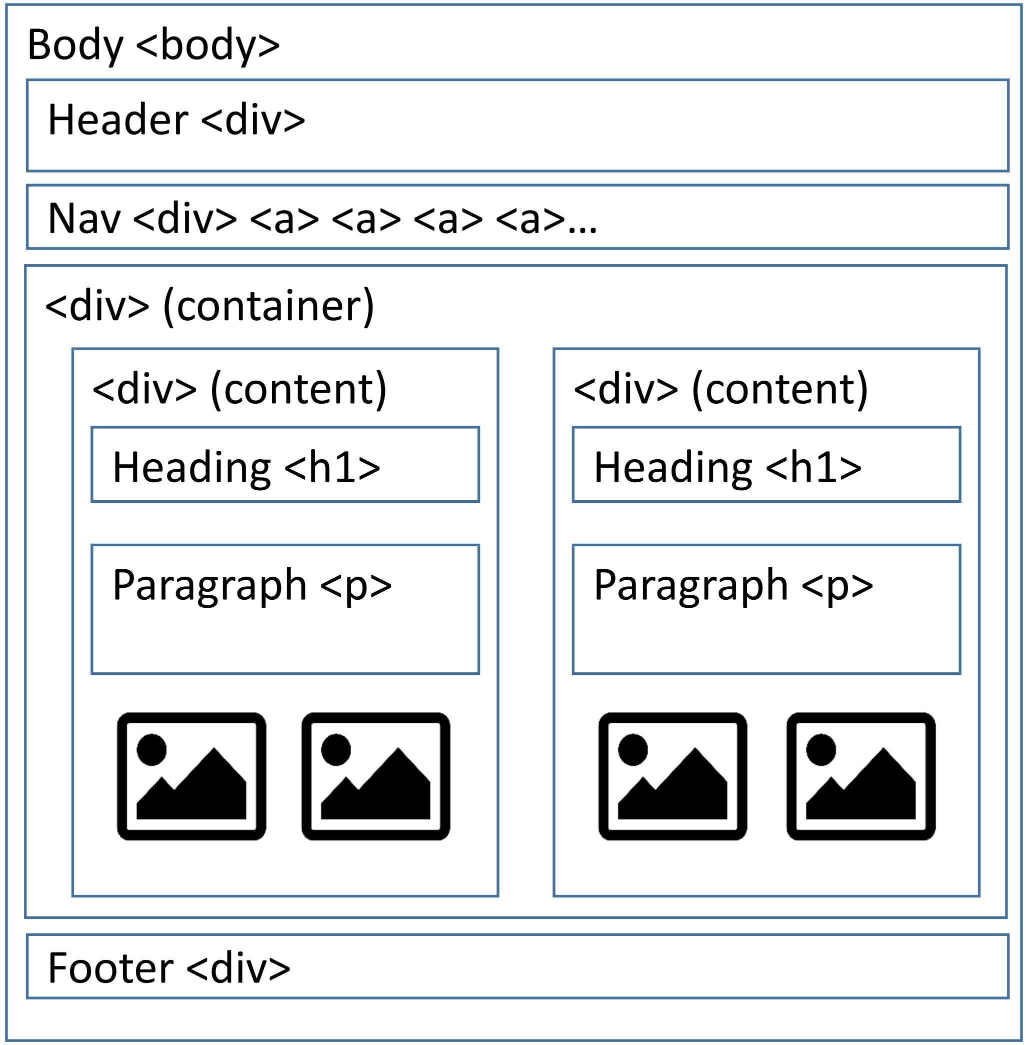
Paragraph <p>

Footer <div>

Another Example Layout



Another Example Layout



Key Issues with Layout

- ✓Layout is a critical element that makes a website to be a success. Or a failure
- ✓A website layout is a pattern (or framework) that defines a website’s structure
- ✓It has the role of structuring the information present on a site both for the website’s owner and for users
- ✓It provides clear paths for navigation within web pages and puts the most important elements of a website front and center

Key Issues with Layout

- ✓A good layout keeps users on the site because it makes important information easily accessible and intuitive to find
- ✓A bad layout frustrates users which then quickly leave the site because they can't find what they are looking for
- ✓There's a strong relationship between the layout and the engagement of users with the website
- ✓It determines how long users dwell on the website, how many pages they browse, and how often they come back to the website

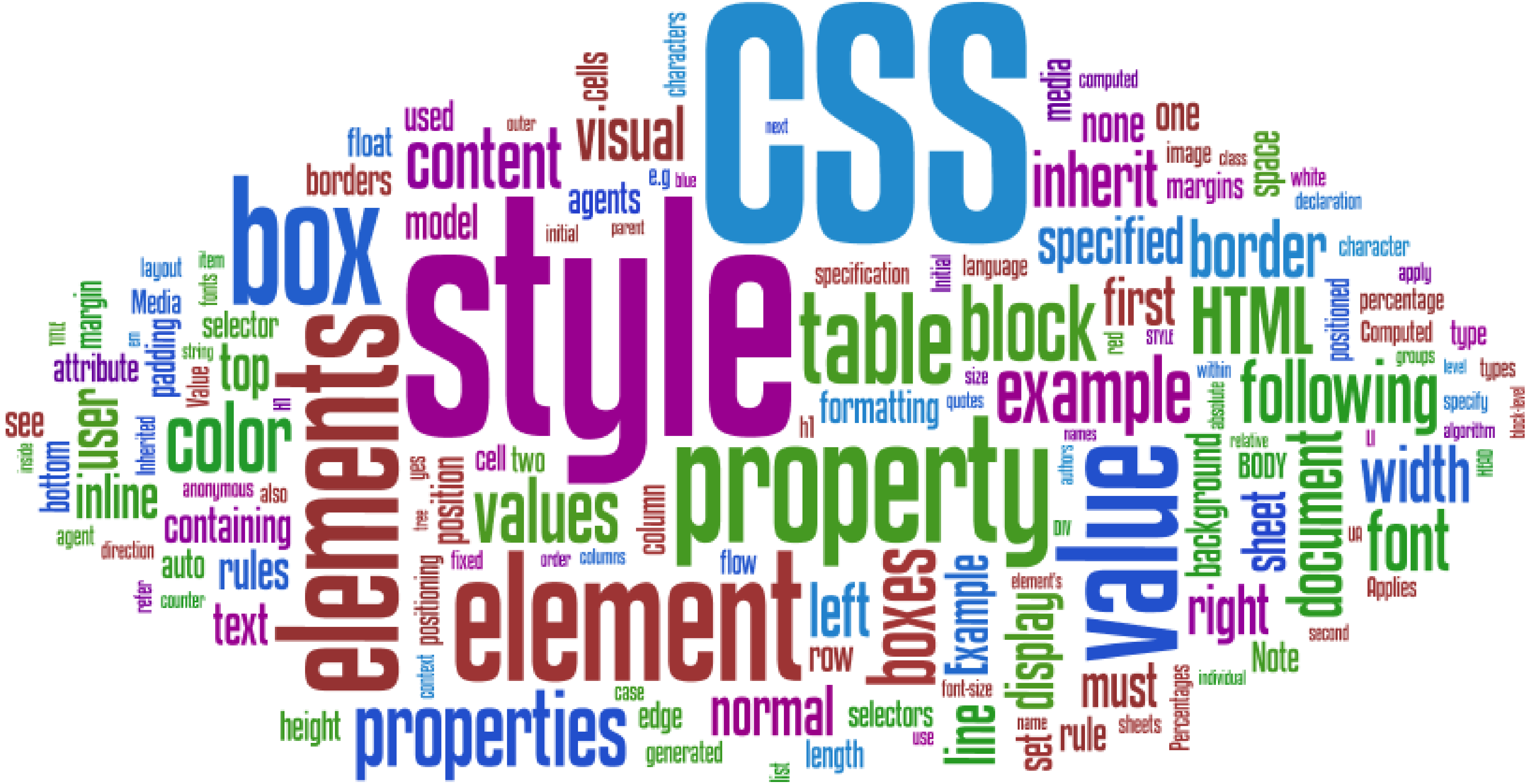
Our Future with HTML

We'll be doing more with HTML in the future

- Tables
- Semantics
- Forms
- And various other tags/elements
- But now…

Introduction to CSS

(Cascading Style Sheets)



East Tennessee State University
Department of Computing

CSCI 1210
Essentials of Web Design

The Power of CSS

What is CSS?

Allow users to apply typographic and design styles for elements on a page

Done by using established HTML elements and specifying a rule for how that element should be displayed

Creates rules that specify how the content of an element should appear

Once you have learned how to write a CSS rule, learning CSS mostly involves learning the different properties you can use

- W3Schools is an excellent resource for this

Advantages

- Greater page layout and style controls
  - Size, color, line spacing, placement, margins, etc.
- Separates style from structure in the HTML document
- Easier site maintenance
  - Make style changes to many pages (entire site) at once

Disadvantages

CSS not uniformly implemented in browsers

- Older browsers don't support. Newer browsers have variances in display

Different versions of CSS are implemented (CSS2 vs CSS3)

HTML5 became standard on October 28, 2014 (hence all the reworking of lecture notes)

Css3 is still not an official standard but is well supported by modern browsers

Versions

CSS 1 (1996) - No longer maintained

CSS 2 (1998) - No longer maintained

CSS 2.1 (2011) - Planned as the first & final revision of level 2, but work on CSS 2.2 began in 2015. This is the current "official" standard

CSS 3 (2012 - 2021) - Things got a lot more complicated here. CSS 3 is broken down into several distinct "modules," which are in varying stages of development. Still hasn't been "officially" adopted as the standard, though modern browsers mostly support it. (* big topic in CSCI 1720)

CSS 4 (?) - Proposed updates to CSS 3. Again, split into modules. It seems unlikely at this point if there will ever be a single, CSS 4 standard

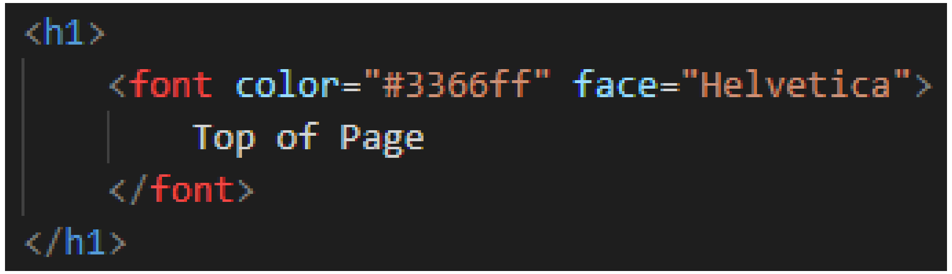
The Power of CSS

HTML defines structure

CSS defines style

Separating the two concerns makes each easier to write and maintain

Back in the day, what would have been



```
<h1>
    <font color="#3366ff" face="Helvetica">
        Top of Page
    </font>
</h1>
```

...and would have to be applied to each **`<h1>`** in the document

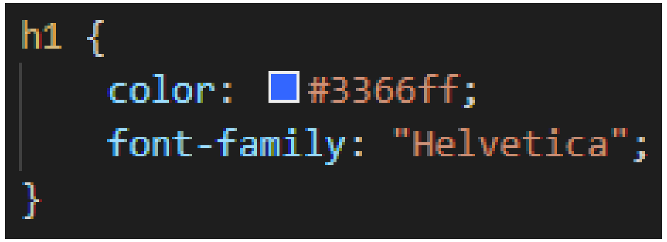
The Power of CSS

HTML defines structure

CSS defines style

Separating the two concerns makes each easier to write and maintain

Is now



```
h1 {
    color: #3366ff;
    font-family: "Helvetica";
}
```

...and can now be applied to all `<h1>` elements at a time

Ways to Define Styles

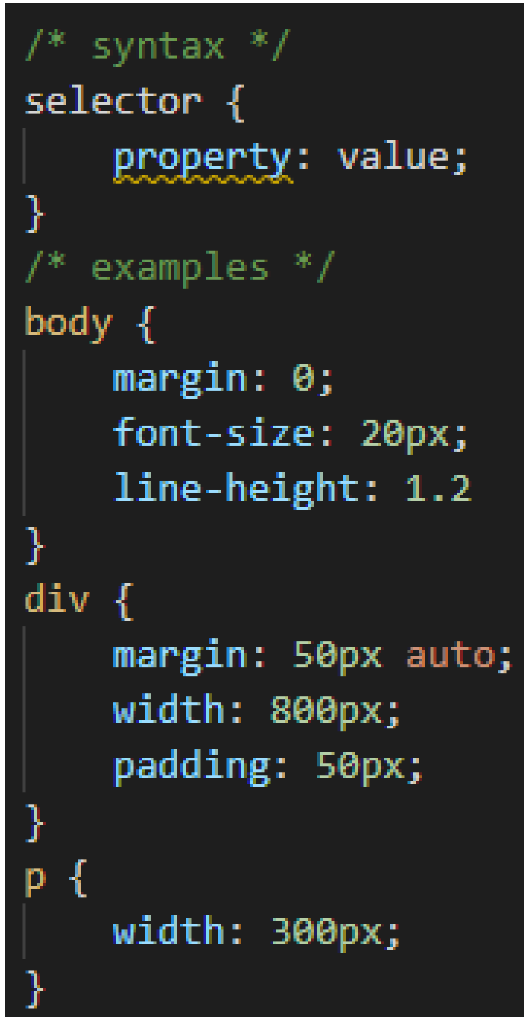
One or more rules

- contained in an external style sheet (linked)
- contained in an internal style sheet (embedded)
- applied directly to a tag (inline)

Ways to Define Styles

Style sheet rule syntax

Property/value pairs separated by semicolon



```
/* syntax */
selector {
    property: value;
}
/* examples */
body {
    margin: 0;
    font-size: 20px;
    line-height: 1.2
}
div {
    margin: 50px auto;
    width: 800px;
    padding: 50px;
}
p {
    width: 300px;
}
```

External Style Sheets

One way we can implement CSS is through external (linked) style sheets

With external style sheets we create a second document and name it with a .css extension, e.g., **`style.css`**

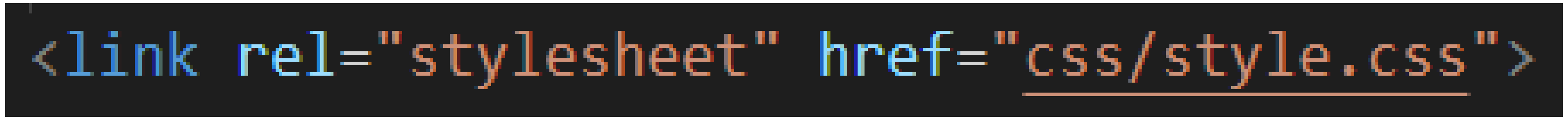
This second file contains only the CSS expressions

Preferred method for employing CSS

External stylesheets allow for modifying style across multiple HTML documents from a single location

External Style Sheets

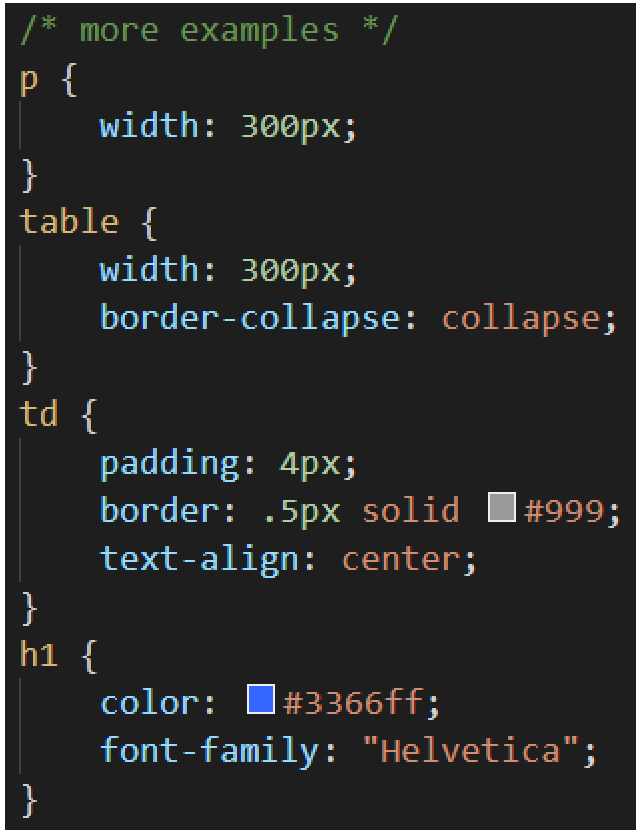
Within every page that needs to use the external style sheet, we add the following tag to the head section of the document:



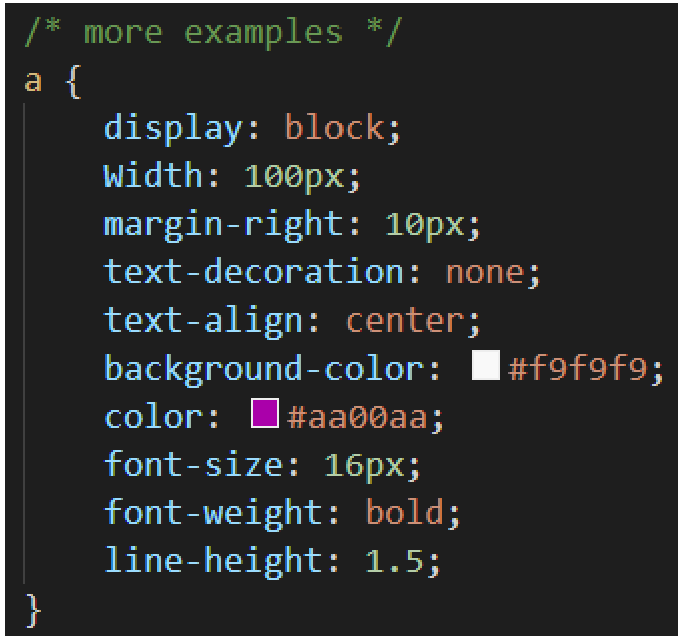
```
<link rel="stylesheet" href="css/style.css">
```

The advantage to external style sheets is that we can create one (or several) css file(s) and our entire website can use it/them

External Style Sheets



```css
/* more examples */
p {
    width: 300px;
}
table {
    width: 300px;
    border-collapse: collapse;
}
td {
    padding: 4px;
    border: .5px solid #999;
    text-align: center;
}
h1 {
    color: #3366ff;
    font-family: "Helvetica";
}
```



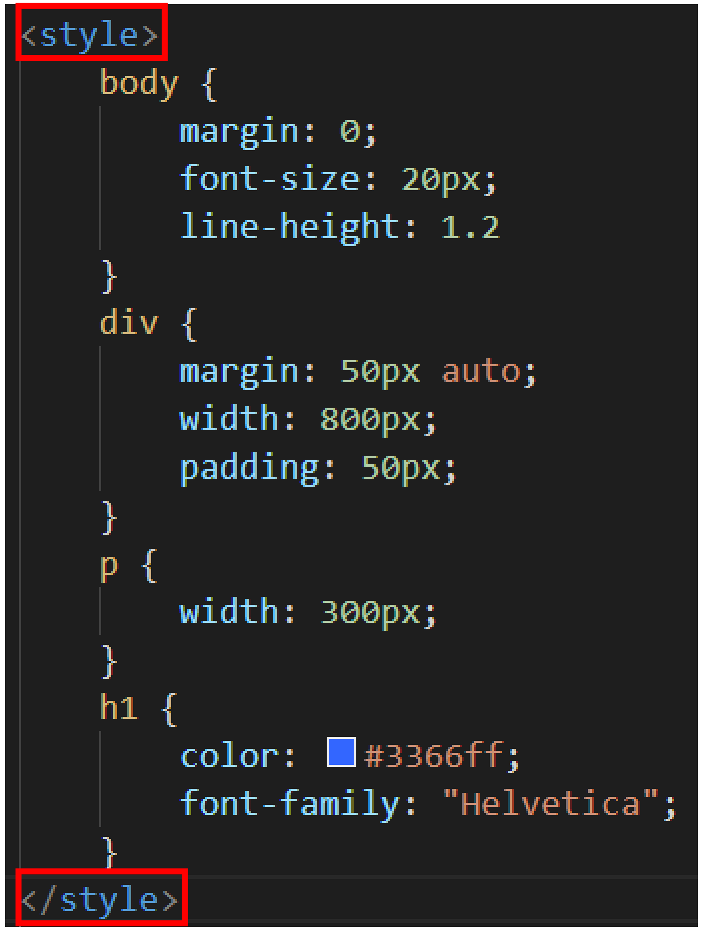
```css
/* more examples */
a {
    display: block;
    Width: 100px;
    margin-right: 10px;
    text-decoration: none;
    text-align: center;
    background-color: #f9f9f9;
    color: #aa00aa;
    font-size: 16px;
    font-weight: bold;
    line-height: 1.5;
}
```

Embedded Style Sheets

Embedded style sheets are defined in the **<head>** section of the document

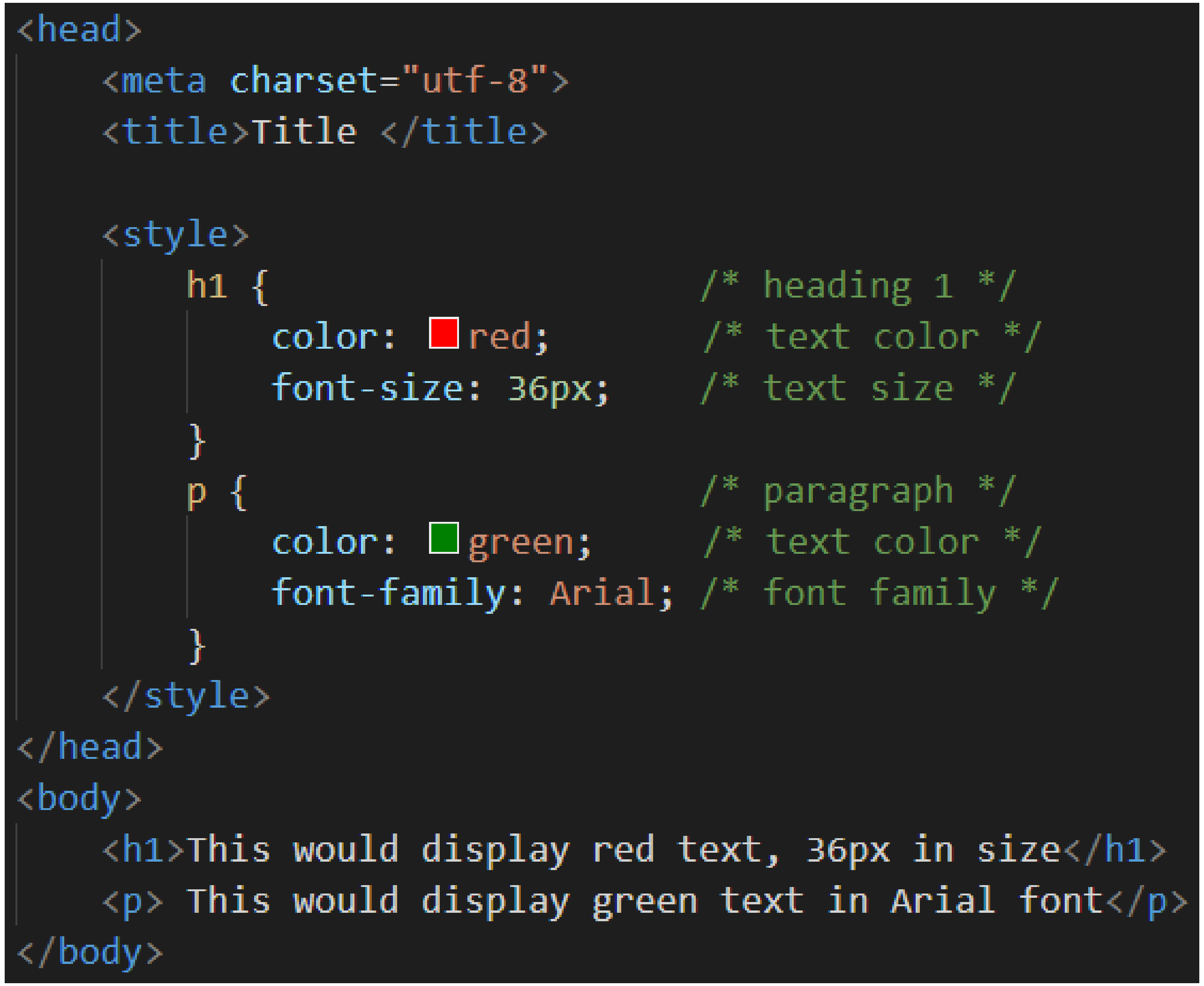
Only apply to the page in which they're included

Embedded style sheets are used to apply style to a single web page



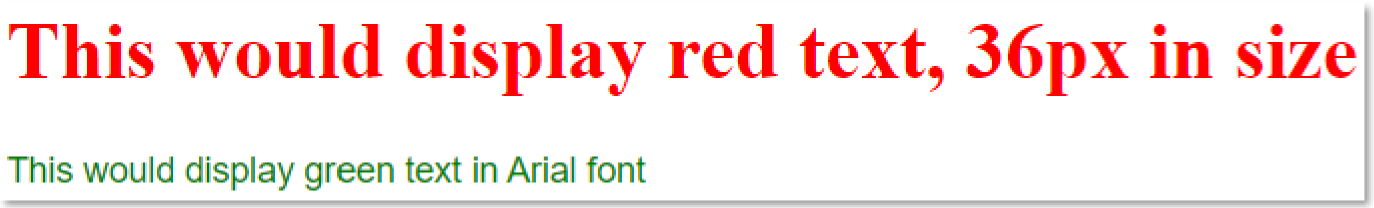
```
<style>
    body {
        margin: 0;
        font-size: 20px;
        line-height: 1.2
    }
    div {
        margin: 50px auto;
        width: 800px;
        padding: 50px;
    }
    p {
        width: 300px;
    }
    h1 {
        color: #3366ff;
        font-family: "Helvetica";
    }
</style>
```

Embedded Style Sheets



```
<head>
    <meta charset="utf-8">
    <title>Title </title>

    <style>
        h1 {                        /* heading 1 */
            color: 🟥red;            /* text color */
            font-size: 36px;        /* text size */
        }
        p {                         /* paragraph */
            color: 🟩green;          /* text color */
            font-family: Arial; /* font family */
        }
    </style>
</head>
<body>
    <h1>This would display red text, 36px in size</h1>
    <p> This would display green text in Arial font</p>
</body>
```

Embedded Style Sheets



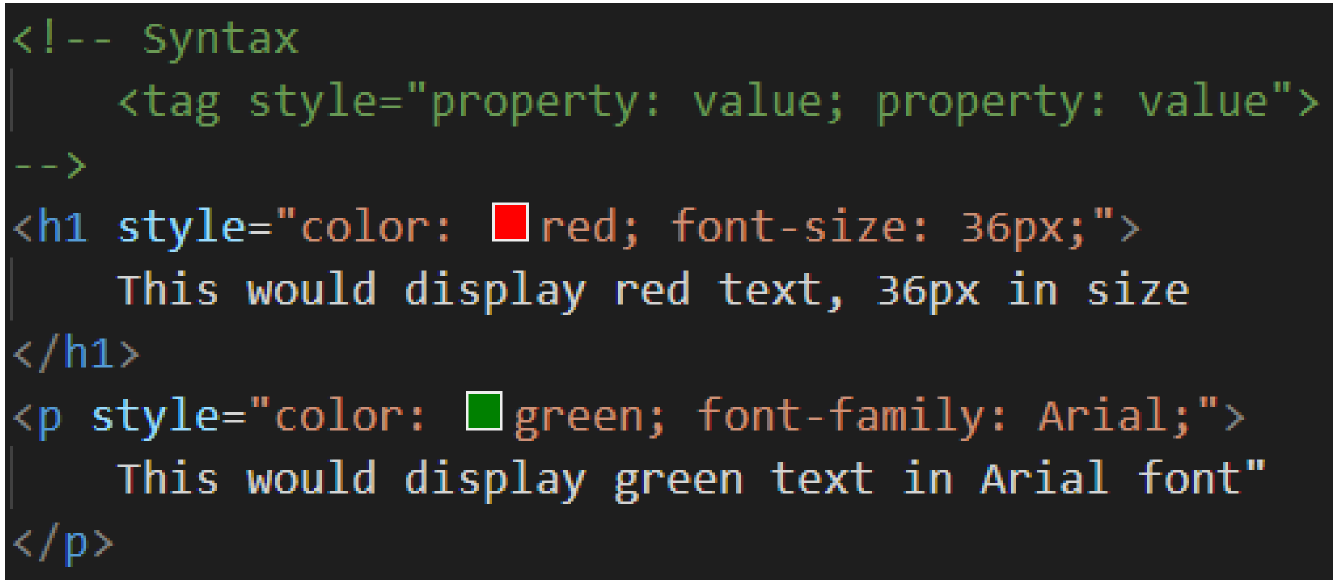
This would display red text, 36px in size

This would display green text in Arial font

Inline Style

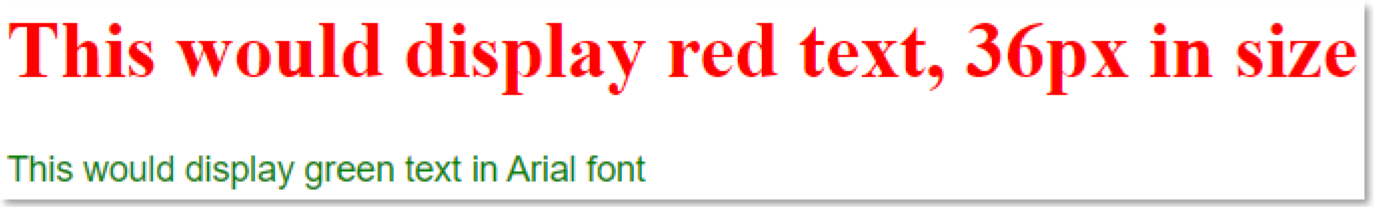
Option three is declaring CSS inline

Inline allows us to add CSS to one specific tag within a document



```
<!-- Syntax
    <tag style="property: value; property: value">
-->
<h1 style="color: 🟥red; font-size: 36px;">
    This would display red text, 36px in size
</h1>
<p style="color: 🟩green; font-family: Arial;">
    This would display green text in Arial font"
</p>
```

Inline Style



This would display red text, 36px in size

This would display green text in Arial font

Order of Operations

Order of Operation

What happens if more than one style is applied to the same property for a tag?

The more specific rule is applied

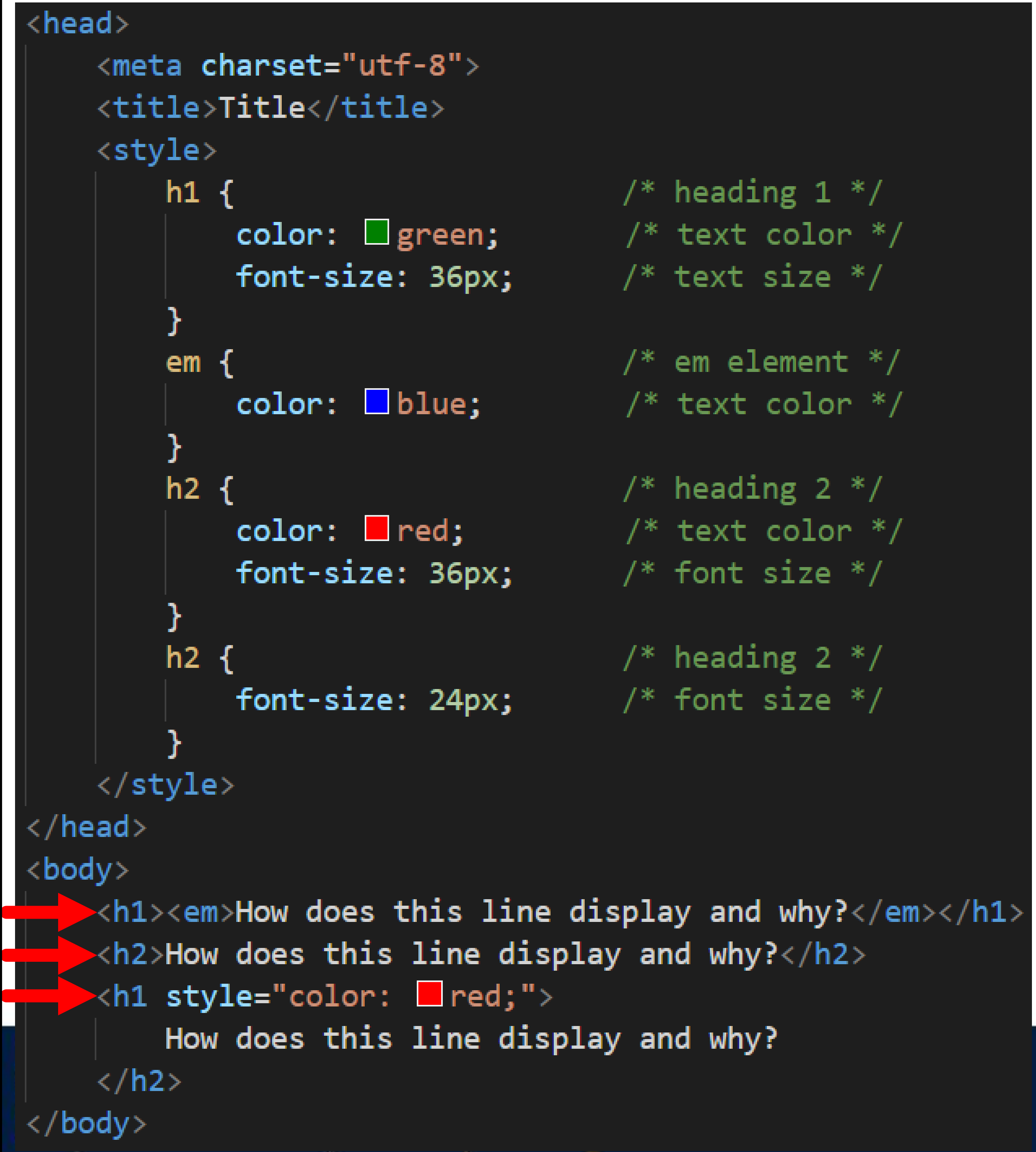
In order of highest priority to lowest priority, we would have:

Inline style

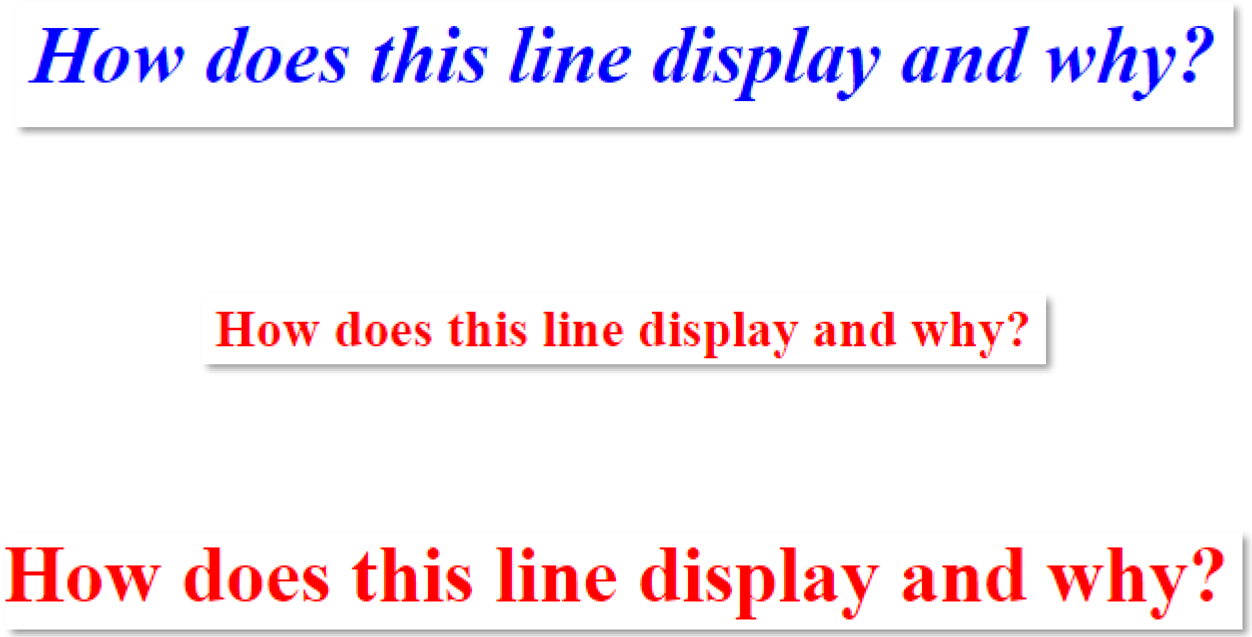
Embedded style

Linked Style

Browser default



```
<head>
    <meta charset="utf-8">
    <title>Title</title>
    <style>
        h1 {                          /* heading 1 */
            color: 🟩green;           /* text color */
            font-size: 36px;          /* text size */
        }
        em {                          /* em element */
            color: 🟦blue;            /* text color */
        }
        h2 {                          /* heading 2 */
            color: 🟥red;             /* text color */
            font-size: 36px;          /* font size */
        }
        h2 {                          /* heading 2 */
            font-size: 24px;          /* font size */
        }
    </style>
</head>
<body>
    <h1><em>How does this line display and why?</em></h1>
    <h2>How does this line display and why?</h2>
    <h1 style="color: 🟥red;">
        How does this line display and why?
    </h2>
</body>
```



How does this line display and why?

How does this line display and why?

How does this line display and why?

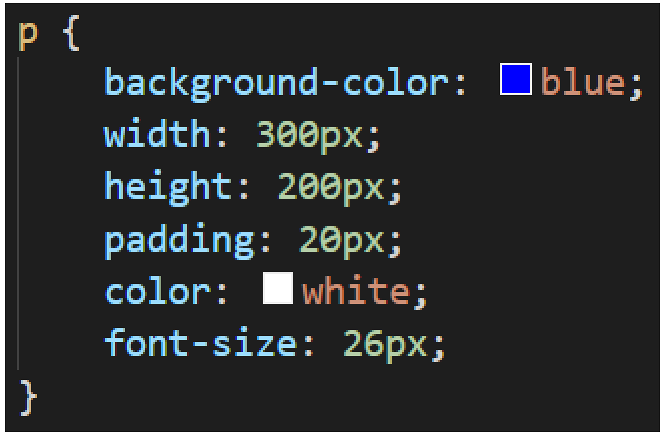
Order of Operation

What happens if more than one property's value is applied to an element using different forms of CSS (External, Embedded, or Inline)?

The more recent the definition has the higher priority (i.e., the rule that is **closest** to the element in the HTML that it affects)

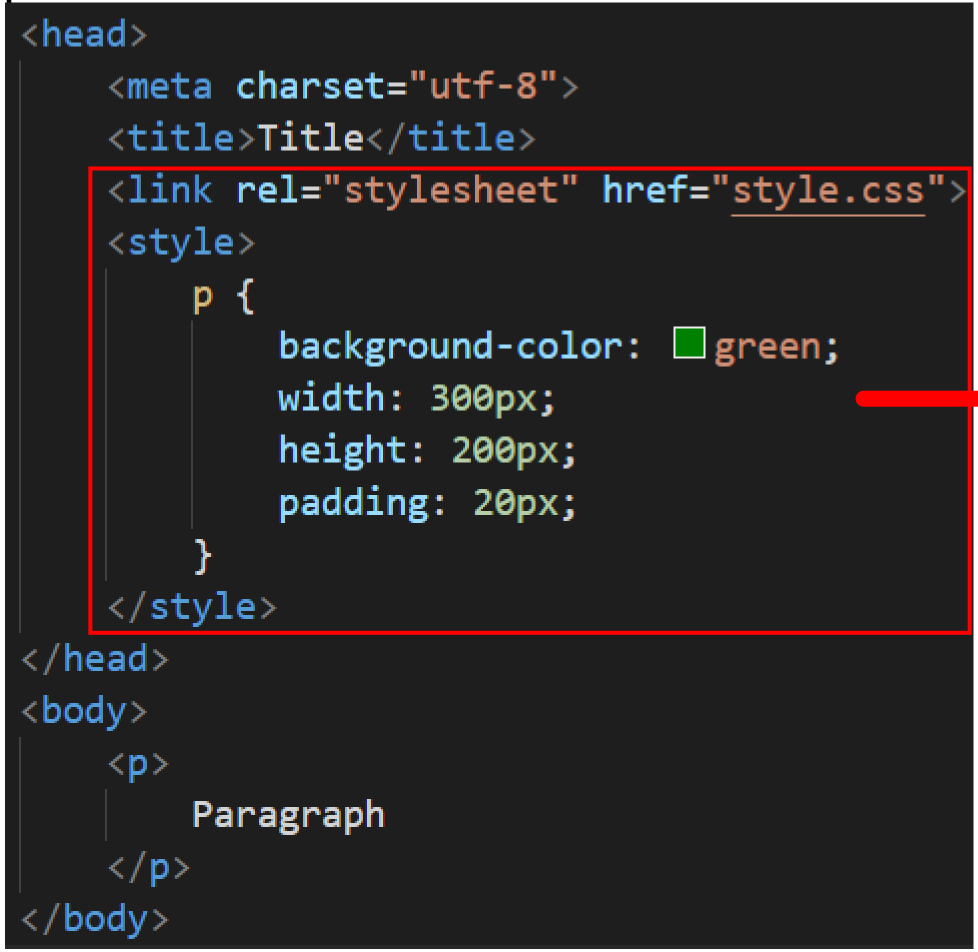
Let's say in the below example, **`style.css`** has a rule that changes all paragraphs to a background color of blue

Order of Operation



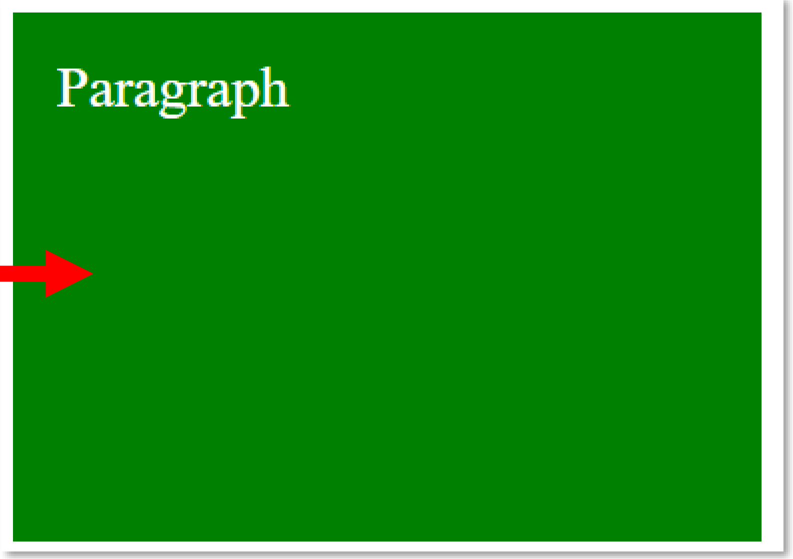
```css
p {
    background-color: 🟦blue;
    width: 300px;
    height: 200px;
    padding: 20px;
    color: ⬜white;
    font-size: 26px;
}
```

style.css



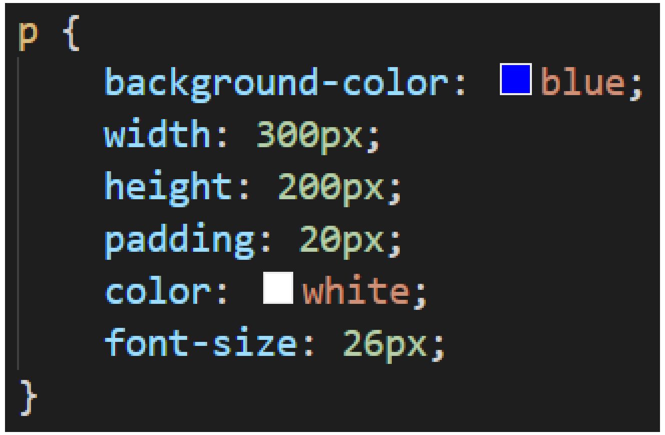
```html
<head>
    <meta charset="utf-8">
    <title>Title</title>
    <link rel="stylesheet" href="style.css">
    <style>
        p {
            background-color: 🟩green;
            width: 300px;
            height: 200px;
            padding: 20px;
        }
    </style>
</head>
<body>
    <p>
        Paragraph
    </p>
</body>
```

index.html



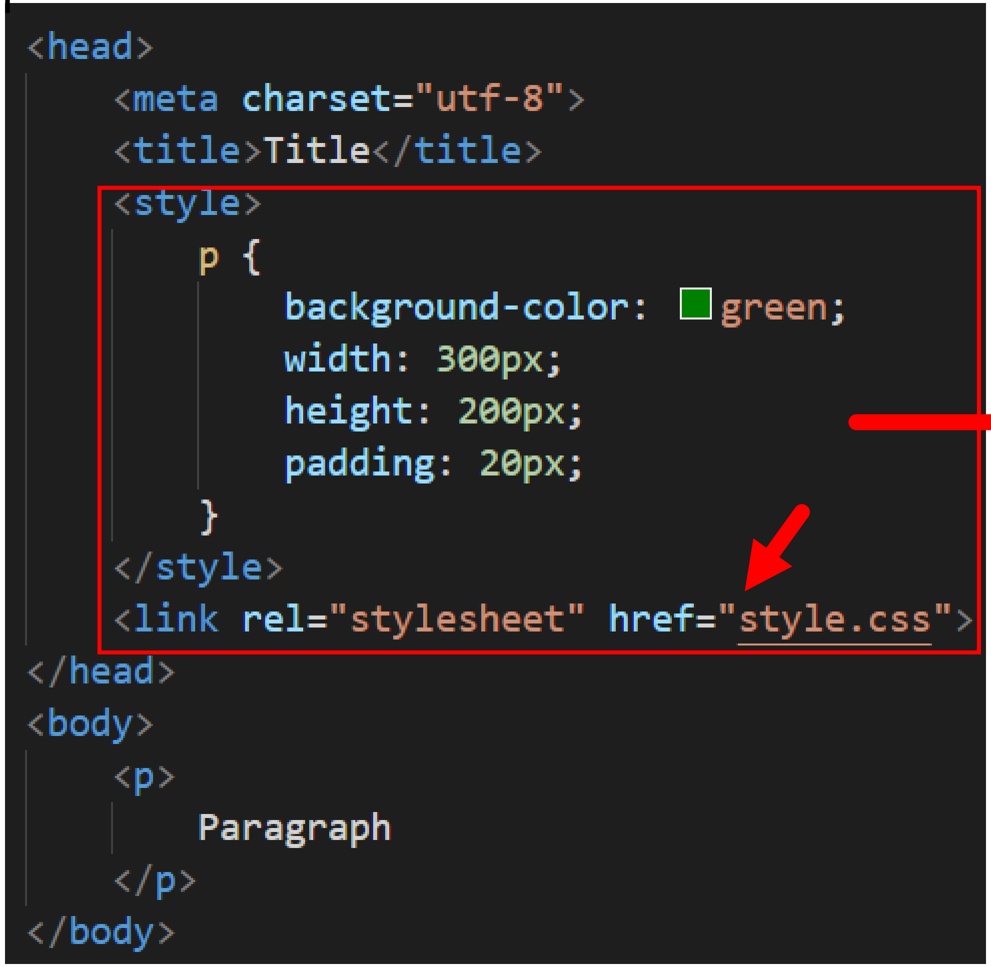
Paragraph

Order of Operation



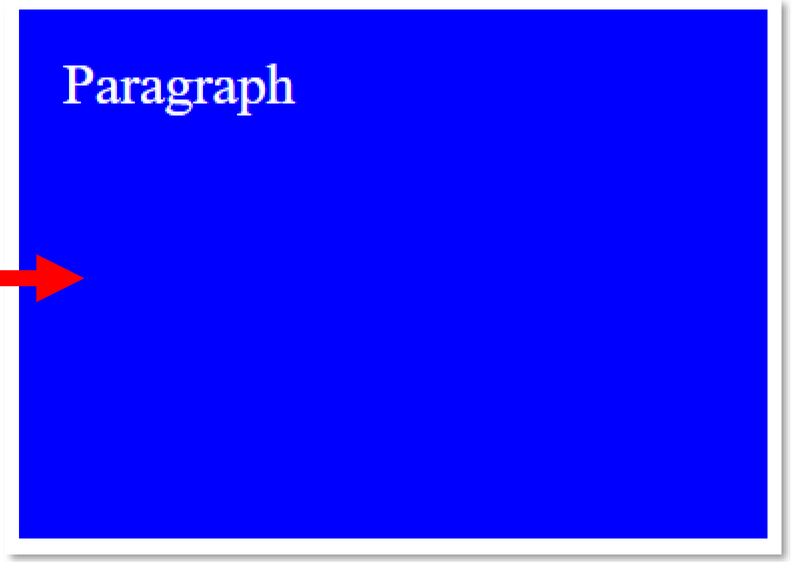
```css
p {
    background-color: 🟦blue;
    width: 300px;
    height: 200px;
    padding: 20px;
    color: ⬜white;
    font-size: 26px;
}
```

style.css



```html
<head>
    <meta charset="utf-8">
    <title>Title</title>
    <style>
        p {
            background-color: 🟩green;
            width: 300px;
            height: 200px;
            padding: 20px;
        }
    </style>
    <link rel="stylesheet" href="style.css">
</head>
<body>
    <p>
        Paragraph
    </p>
</body>
```

index.html



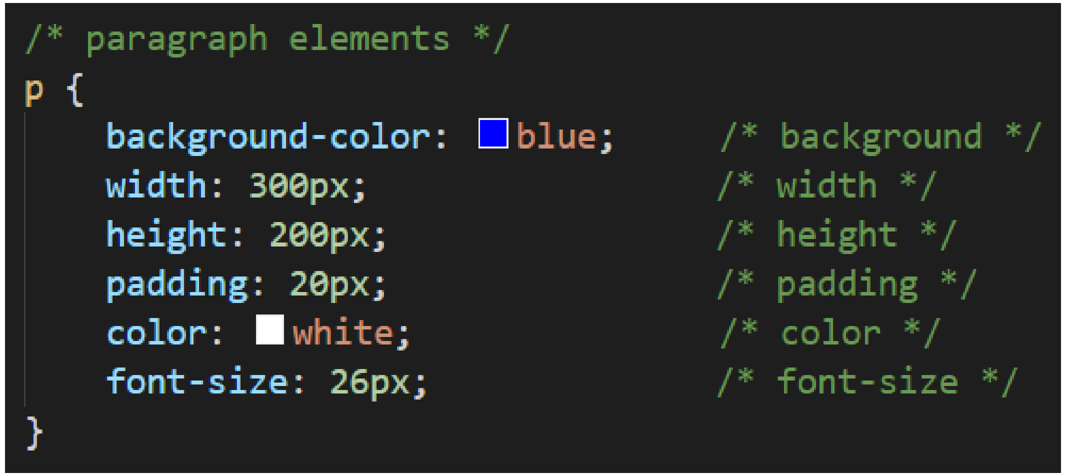
Paragraph

Commenting

Commenting

In CSS, we can apply comments to embedded and linked CSS

To place a comment we use the syntax -- /* COMMENT */



```css
/* paragraph elements */
p {
    background-color: blue;       /* background */
    width: 300px;                 /* width */
    height: 200px;                /* height */
    padding: 20px;                /* padding */
    color: white;                 /* color */
    font-size: 26px;              /* font-size */
}
```

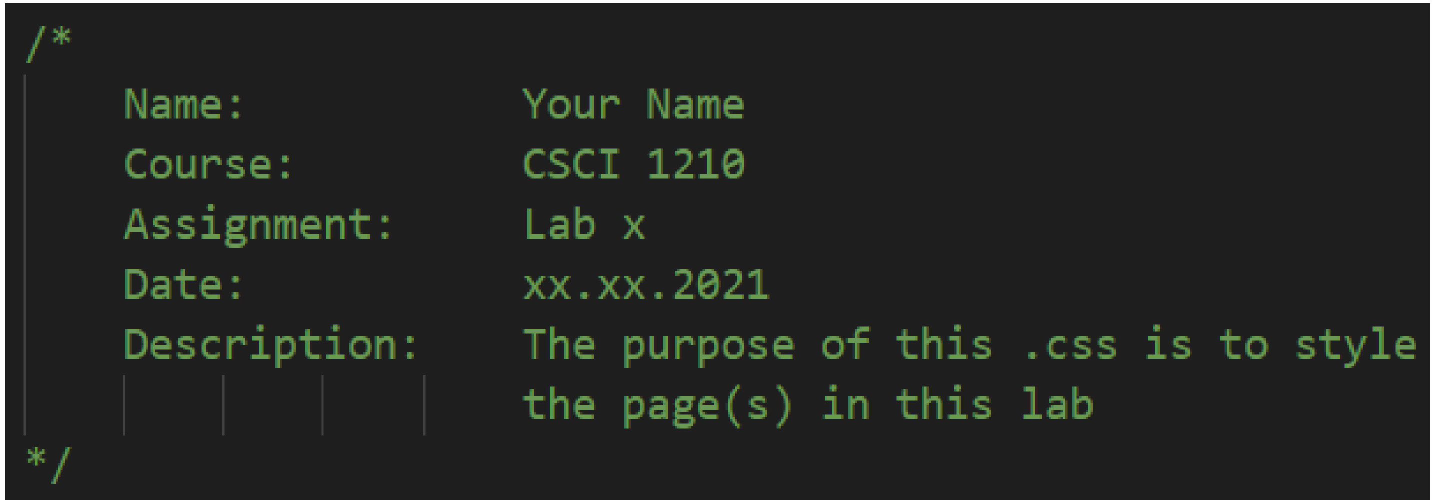
Generally, you don’t have to be this detailed with your comments (this is mostly for illustration)

But comments should be included in your code to explain what’s going on

You might not be the person who has to update it later!

Commenting

For all .css files created, you must have the following comment block:



```
/*
    Name:               Your Name
    Course:             CSCI 1210
    Assignment:         Lab x
    Date:               xx.xx.2021
    Description:        The purpose of this .css is to style
                        the page(s) in this lab
*/
```

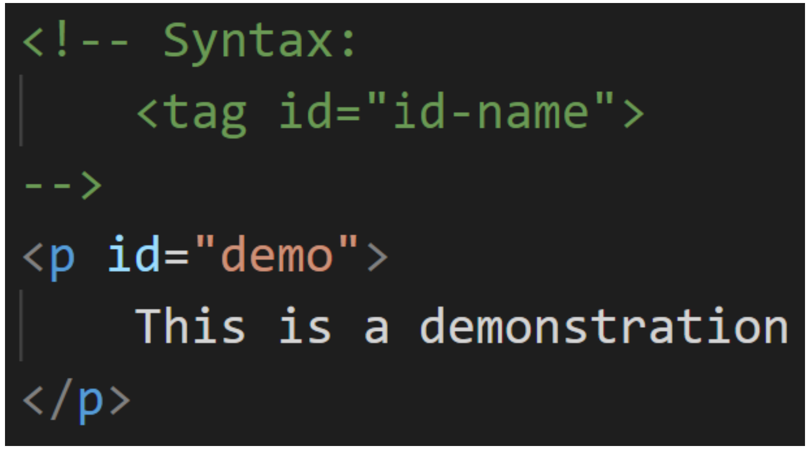
IDs & Classes

ID Attribute

Every element on a web page can be assigned a **unique** ID so that it can be identified by other applications such as CSS or Javascript

A given ID can appear only once on a given page

Value should start with a letter or an underscore (_), not a number or any other character



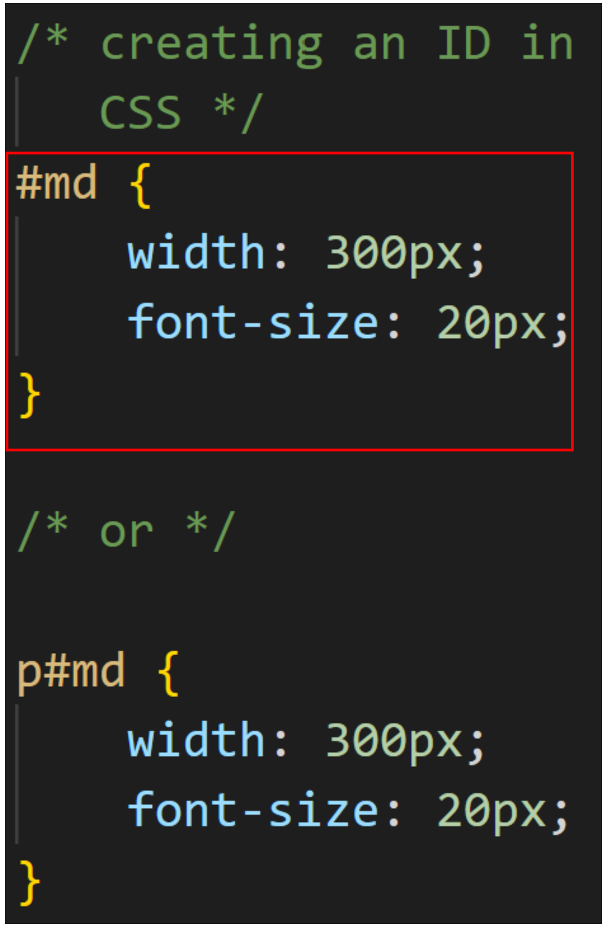
```
<!-- Syntax:
    <tag id="id-name">
-->
<p id="demo">
    This is a demonstration
</p>
```

ID Attribute

Since an ID is unique to a page, this is how we usually reference it in CSS

The octothorpe ("#") will apply the CSS rules to that element only

The second example is valid, but it isn't necessary to include the element name (see above)



```
/* creating an ID in
   CSS */
#md {
    width: 300px;
    font-size: 20px;
}

/* or */

p#md {
    width: 300px;
    font-size: 20px;
}
```

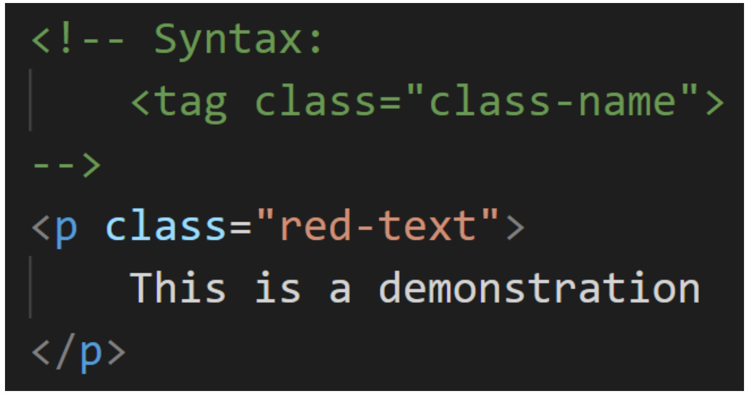
Class Attribute

Every element on a web page can also have one or more class attributes

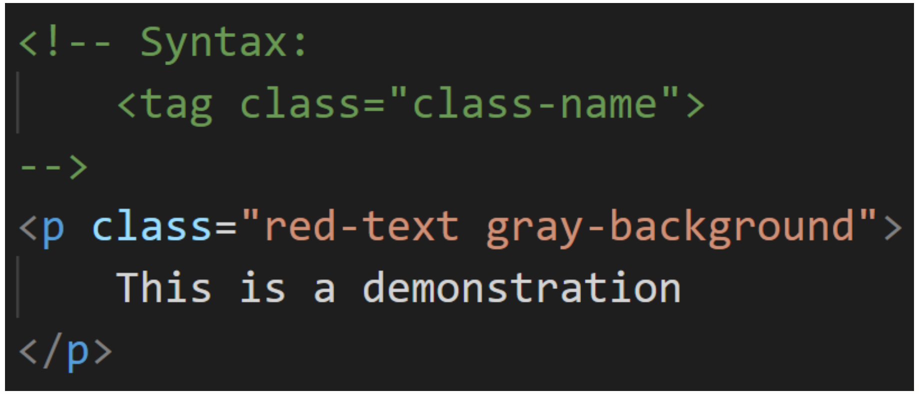
Classes can be assigned to multiple elements on a page

Value must start with a letter or an underscore (_), not a number or any other character

Can assign multiple classes to an element, separated by spaces



```
<!-- Syntax:
    <tag class="class-name">
-->
<p class="red-text">
    This is a demonstration
</p>
```

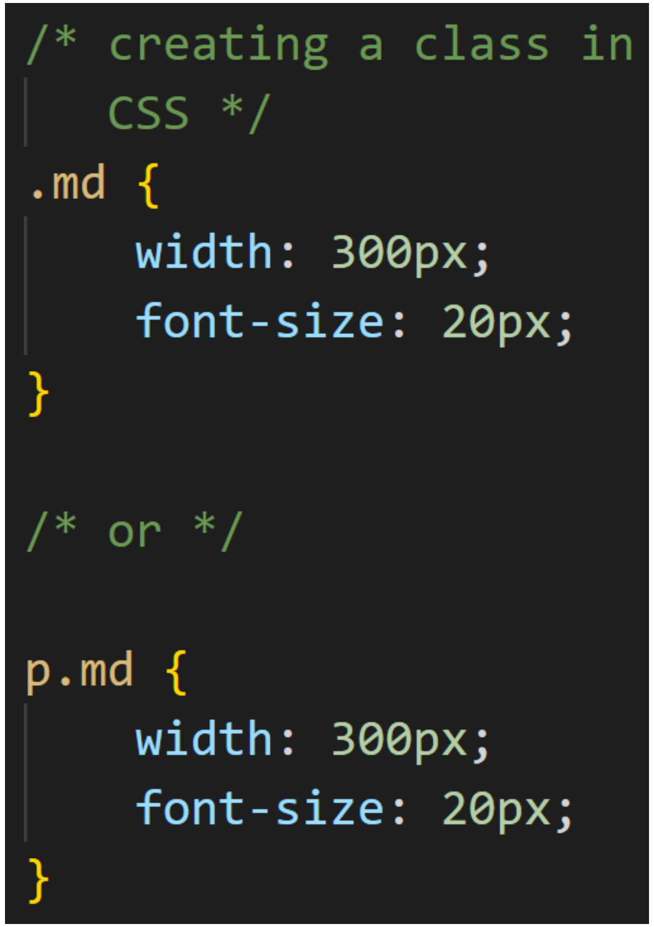


```
<!-- Syntax:
    <tag class="class-name">
-->
<p class="red-text gray-background">
    This is a demonstration
</p>
```

Class Attribute

Again, classes can be used to modify an element's presentation, in multiple elements on a given page. These are the building-blocks used by CSS frameworks

To create a class, we use a dot (".")

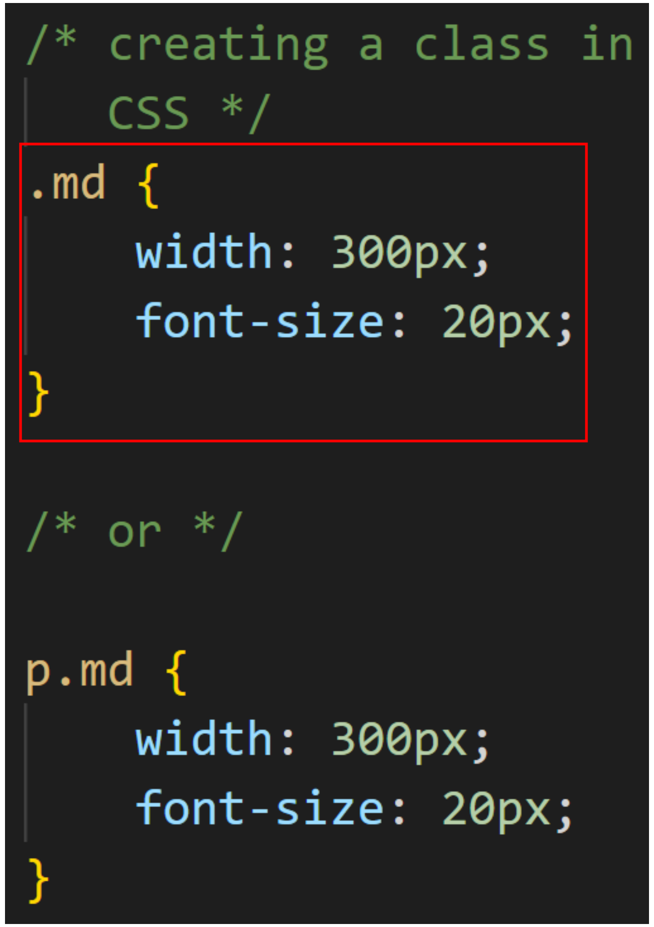


```
/* creating a class in
   CSS */
.md {
    width: 300px;
    font-size: 20px;
}

/* or */

p.md {
    width: 300px;
    font-size: 20px;
}
```

Class Attribute

Will be applied to any element that includes this class

More generic

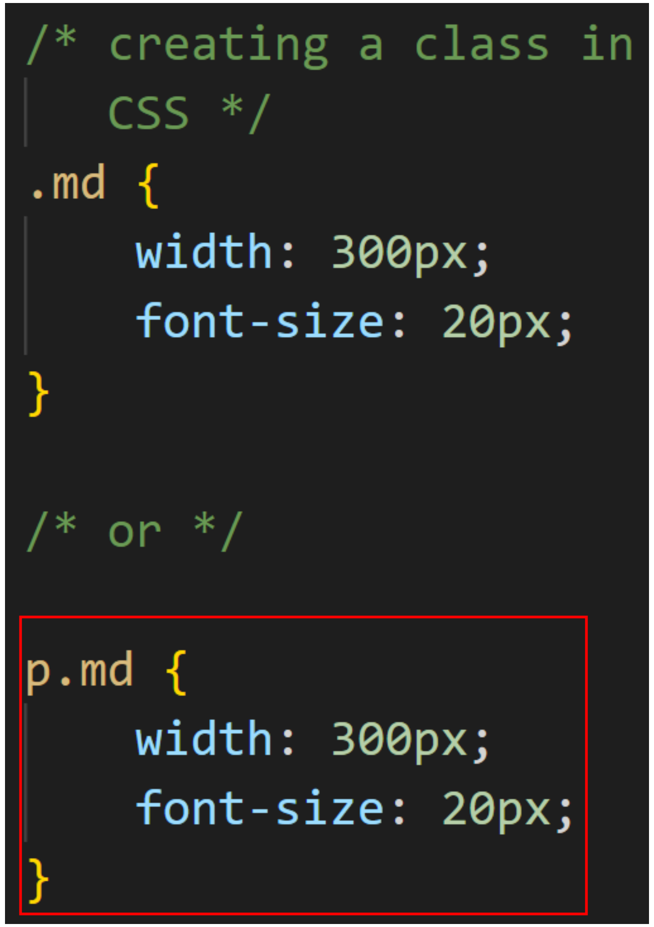


```css
/* creating a class in
   CSS */
.md {
    width: 300px;
    font-size: 20px;
}

/* or */

p.md {
    width: 300px;
    font-size: 20px;
}
```

Class Attribute

Will be applied only to paragraph elements

More specific



```
/* creating a class in
   CSS */
.md {
    width: 300px;
    font-size: 20px;
}

/* or */

p.md {
    width: 300px;
    font-size: 20px;
}
```

CSS Text Properties

CSS Text Properties

`text-align:` (left, right, center, justified) -- **`p{text-align:center;}`**

`text-indent:` (pixels or percentage) -- **`p{text-indent: 5%;}`**

`text-decoration:` (none, overline, underline, line-through) -- **`p{text-decoration: underline;}`**

`word-spacing:` (pixels) -- **`p{word-spacing: 5px;}`**

`letter-spacing:` (pixels) -- **`p{letter-spacing: 5px;}`**

CSS Text Properties

`line-height:` (pixels, percentage, number) --
`p{line-height: 2;}` /* this is double space */

`color:` (name, hex, rgb, rgba) -- **`p{color: red;}`**

`font-family:` (specific,to,general) --
`p{font-family:”Times New Roman”, Georgia, Serif;}`

CSS Text Properties

`font-size:` (px, pt, em, or %) -- **`p{font-size: 10px;}`**

`font-weight:` (lighter, normal, bold, bolder, or value 100-900) -- **`p{font-weight:bold;}`**

`font-style:` (italic, oblique) -- **`p{font-style:italic;}`**

CSS Background

`background-color:` (name, hex, rgb, rgba) -- **`p{background-color: green;}`**

`background-image:` url(imageLocation relative or abs) -- **`p{background-image: url(“images/bg.jpg”);}`**

`background-repeat:` (no-repeat, repeat-x, repeat-y, repeat) -- **`p{background-repeat: no-repeat;}`**

`background-size:` (width and height in px or percent, or the word “cover”) -- **`p{background-size: cover;}`**

CSS Background

background-position: (left, right, center, top, bottom, center)

background-attachment: (fixed, scroll)

```
body{
     background-image: url(“images/bg.gif”);
     background-repeat: no-repeat;
     background-size: 100% 100%;
}
```

CSS Colors

Colors

Effective use of colors can make or break a site

CSS has a number of ways of defining color

- Color keywords (“red,” “green,” “blue,” etc.)
- RGB (Red, Green, Blue)
- Hexadecimal
- RGBa (Red, Green, Blue, Alpha)
- HSL (Hue, Saturation, Luminosity)
- HSLa (Hue, Saturation, Luminosity, Alpha)

We’ll look at the first four

Colors

Option 1 - named colors

One option is to use the named color keywords.

There are 17 standard colors

- aqua, black, blue, fuchsia, gray, green, lime, maroon, navy, olive, orange, purple, red, silver, teal, white, and yellow

#663399Becca

There are 128 additional colors

- BlanchedAlmond, ForestGreen, HoneyDew, WhiteSmoke, RebeccaPurple

Colors

#663399Becca

Purple was the favorite color of Rebecca Alison Meyer who passed away twelve hours into her sixth birthday from brain cancer. Rebecca was the daughter of prolific CSS standards pioneer Eric Meyer. Eric kept his online colleagues informed of the battle his daughter and family were waging through blog posts and brief updates on Twitter.

After hearing the awful news, designer/author Jeffrey Zeldman decided to do something and started a Twitter hashtag campaign. What started as a hashtag charity campaign evidently transformed into a much larger project. Given Eric's prolific work on CSS, it was proposed that the hex-value #663399, a shade of purple, be aliased to "beccapurple."

When informed of the initiative, Eric had one request if the standards body were to adopt the proposal: call it "rebeccapurple" instead. Eric writes that "Rebecca informed us that she was about to be a big girl of six years old, and Becca was a baby name. Once she turned six, she wanted everyone (not just me) to call her Rebecca, not Becca."

Colors

Option 2 is to signify the decimal value that represents the color.

The colors on the screen are comprised of some mixture of Red, Green, and Blue (RGB)

Each color (R,G,B) can have a value from 0 (no color applied) to 255 (full color intensity)

Colors

Express the RGB values in a comma delimited list

This would be accomplished by rgb(RED, GREEN, BLUE);

Example: color: rgb(255, 0, 0);   /* This would be red */

Example: color: rgb(255, 255, 255); /* This would be white */

Note: Do Not place a space between the ‘b’ in rgb and the opening paranthesis (‘(‘)

Colors

	R	G	B
Red	255	0	0
Green	0	255	0
Blue	0	0	255
White	255	255	255
Black	0	0	0

Colors

Option 3 – Hexadecimal

Base 10 (decimal)	Base 16 (hexadecimal)
0	0
10	A
16	10
255	FF

Hexadecimal expresses single digits numbers 0-9, A, B, C, D, E, F. These numbers would be equivalent to decimal numbers 0-15 (with F being equal to 15)

To represent the values 0-255, two hexadecimal digits are needed

Colors

Option 3 – hexadecimal

To convert from decimal to hex, divide decimal number by 16. Quotient (expressed in hex) is left digit, remainder (expressed in hex) is right digit

$175_{10} = 175/16 = 10 \text{ r } 15 = AF_{16}$

$219_{10} = 219/16 = 13 \text{ r } 11 = DB_{16}$

To convert from hex to decimal, convert left digit to decimal and multiply by 16, add to that the right digit converted to decimal

$BC_{16} = 11*16 +12 = 188_{10}$

$AA_{16} = 10*16 + 10 = 176_{10}$

Or, just use a
Scientific calculator

Colors

Option 3 – hexadecimal

Black	#000000
White	#FFFFFF
Red	#FF0000
Green	#00FF00
Blue	#0000FF
BlanchedAlmond	#FFEBCD
DeepSkyBlue	#00BFFF
GreenYellow	#ADFF2F

THINGS
AREN'T ALWAYS
#000000
AND
#FFFFFF

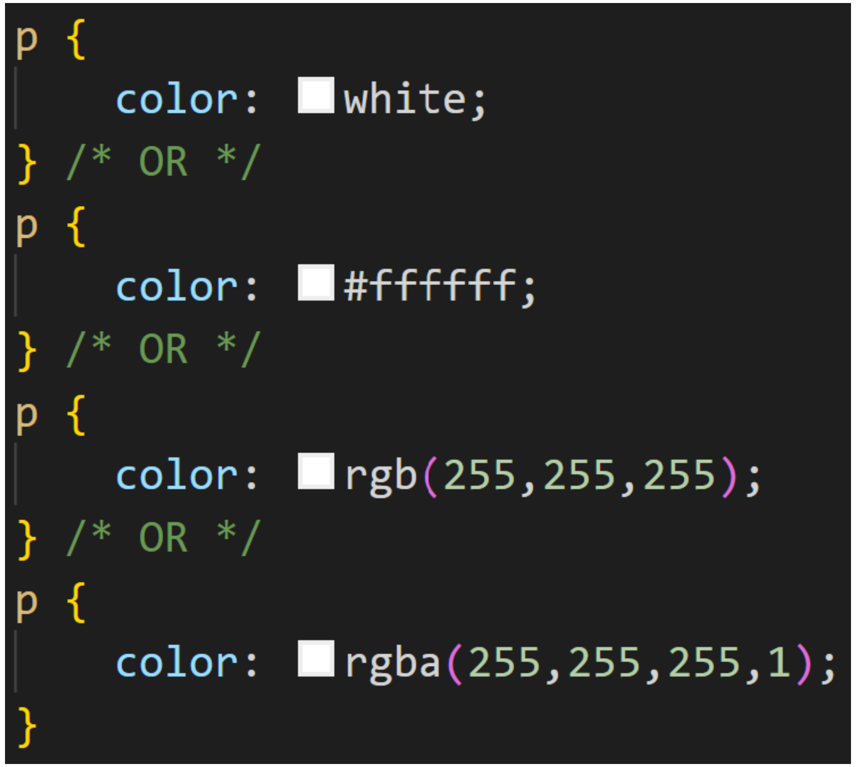
Colors

Option 4 would be to express the RGBA values

- Very similar to option 2, however this allows us to identify the opacity of the object by setting the Alpha parameter. This ranges from 0.0 (fully transparent) to 1.0 (fully opaque)
- This is supported in IE9+, Firefox 3+, Chrome, Safari, Opera 10+
- rgba(RED, GREEN, BLUE, ALPHA);
- Example: color: rgba(255, 0, 0, 0.3); /* red with opacity */

Colors

So to change the font color of a paragraph to white we could do one of the following:



```
p {
    color: □white;
} /* OR */
p {
    color: □#ffffff;
} /* OR */
p {
    color: □rgb(255,255,255);
} /* OR */
p {
    color: □rgba(255,255,255,1);
}
```

Questions?

Lecture Quiz

1. Which of the following is not a valid class or id name?

A. contentTwo

B. _content

C. Content2

D. $content

Lecture Quiz

2. What color is this: **`#00FF00`**?

A. ⬛

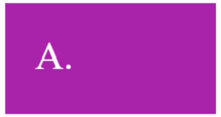
B. 🟥

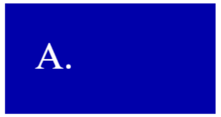
C. 🟦

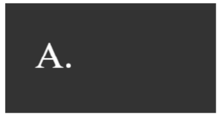
D. 🟩

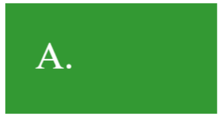
Lecture Quiz

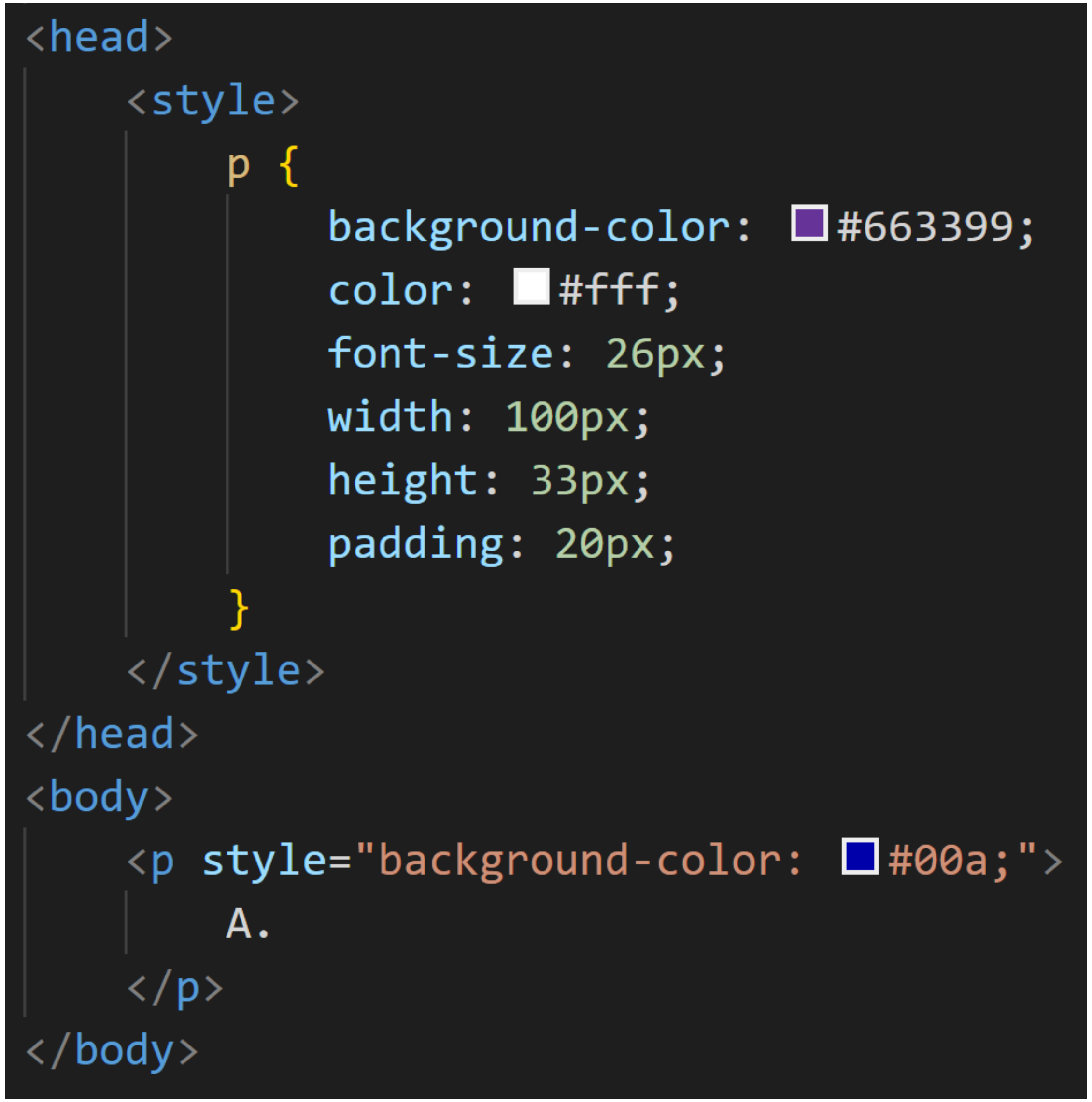
3. Given the following code, what is the background color of the paragraph?

A. 

B. 

C. 

D. 



```
<head>
    <style>
        p {
            background-color: ■#663399;
            color: □#fff;
            font-size: 26px;
            width: 100px;
            height: 33px;
            padding: 20px;
        }
    </style>
</head>
<body>
    <p style="background-color: ■#00a;">
        A.
    </p>
</body>
```

Lecture Quiz

4. What CSS property would we use to modify the display size of text?

A. `font-size`

B. `text-size`

C. `size-font`

D. `text-points`

Lecture Quiz

5. There is a strong relationship between user experience and layout?

A. True

B. False

Lecture Quiz

6. Which is the correct way to assign multiple classes to an element?

A. `<p class="md high-contrast centered">`

B. `<p class="md, high-contrast, centered">`

C. `<p class="md + high-contrast + centered">`

D. `<p class="md/high-contrast/centered">`

Lecture Quiz

7. Which of the following correctly shows how to create a class in CSS?

A. `.image { }`

B. `p.image { }`

C. `div.image { }`

D. All of the above are correct

Lecture Quiz

8. What is the current, “official” version of CSS?

A. 1

B. 2

C. 2.1

D. 3

E. 4

Lecture Quiz

9. What color is this: **`rgb(255,0,0)`** ?

A. ⬛

B. 🟥

C. 🟦

D. 🟩

Lecture Quiz

10. Given the following CSS rule, which is a selector?

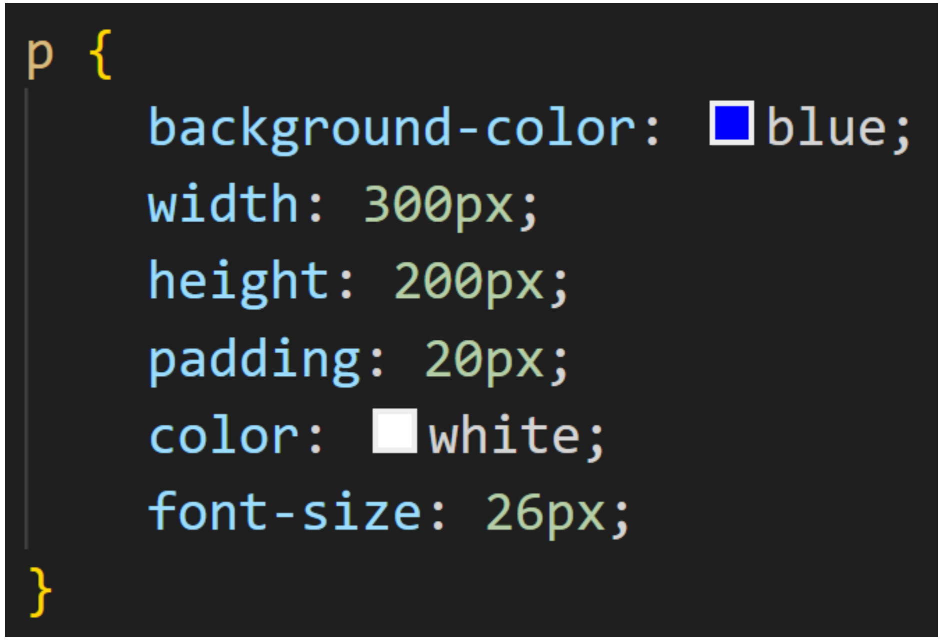
A. p

B. { }

C. width

D. 300px

E. ;



```css
p {
    background-color: blue;
    width: 300px;
    height: 200px;
    padding: 20px;
    color: white;
    font-size: 26px;
}
```

Sources

- “CSS Legal Color Values”, W3Schools, Retrieved from http://www.w3schools.com/cssref/css_colors_legal.asp
- “HTML Color Names”, W3Schools, Retrieved from http://www.w3schools.com/html/html_colornames.asp
- “HTML Reference”, W3Schools, Retrieved from http://www.w3schools.com/tags/default.asp
- “HTML Reference”, W3Schools, Retrieved from http://www.w3schools.com/tags/default.asp
- “HTML Symbols”, W3Schools, Retrieved from http://www.w3schools.com/html/html_symbols.asp
- Kooser, A. “3 percent of American adults still cling to dial-up Internet”, W3Schools, August 2013, Retrieved From http://www.cnet.com/news/3-percent-of-american-adults-still-cling-to-dial-up-internet/
- Duckett, J. (2011). HTML & CSS: Design and build websites. Indianapolis,IN: John Wiley & Sons.

Copyrights

Presentation prepared by and copyright of John Ramsey, East Tennessee State University, Department of Computing . (ramseyjw@etsu.edu)