CIS 371 Web Application Programming

Styles in CSS

GRAND VALLEY STATE UNIVERSITY®

Lecturer: Dr. Yong Zhuang

Based on the original version by Professor Hans Dulimarta.

The Origins of Cascading Style Sheets

- Browser Wars: Major commercial browsers created new HTML tags for Web authors – tags that could only be interpreted with their own browsers. Netscape Navigator and Internet Explorer are used by at least 90% of web users. But they are not compatible with each other.
- A professional Web author must test his web pages against different versions of each of the major browsers and several of the less popular browsers for compatibility.
- However, one idea that helps make the Web useful in the first place is that Web pages should be viewable by all browsers on all platforms.
- This is where the World Wide Web Consortium (W3C) enters the picture.

The W3C and the CSS

- The W3C has convinced major software companies, including Netscape Communications, Microsoft, IBM, Novell, Sun Microsystems, etc., to become members of this standard body.
- The W3C came up with the idea of Cascading Style Sheets (CSS) to head off the need by browser manufacturers to introduce even more HTML tags.
- CSS is a compromise, and provides the page layout features which Web authors want by adding CSS formatting elements to existing HTML tags.

History of CSS

20 Years of CSS

https://www.w3.org/Style/CSS20/

CSS Zen Garden launched in 2003. http://www.csszengarden.com/

Applying CSS to HTML

```
<html>                Option 1: External
  <head>
    <link rel=”stylesheet”
          href=”mystyles.css”>
  </head>
  <body>
    <p>Hello world</p>
  </body>
</html>
```

```
/* in mystyles.css */
p {
  border: 2px
  solid red;
}
```

```
<html>    Option 2: Internal
  <head>
    <style>
      p {
        border: 2px solid red;
      }
    </style>
  </head>
  <body>
    <p>Paragraph 1</p>
    <p>Paragraph 2</p>
  </body>
</html>
```

```
<body>    Option 3: Inline (Not recommended)
  <!-- inline style -->
  <p style=”border: 2px solid red”>......</p>
</body>
```

Cascading Order

If we apply CSS to an HTML element using external, internal, and inline methods simultaneously, and they conflict with each other, which CSS style will take precedence?

Cascading Order

- Situation: A browser is presented with a number of CSS statements, some of which conflict with each other.
- All the styles will "cascade" into a new "virtual“ style sheet by the following rules:

1. Browser Default
2. External Style Sheet
3. Internal Style Sheet
4. Inline Style

Lowest

Highest

- Rule 4 has the highest priority

Specific CSS rules overrule general ones.

How to define styles?

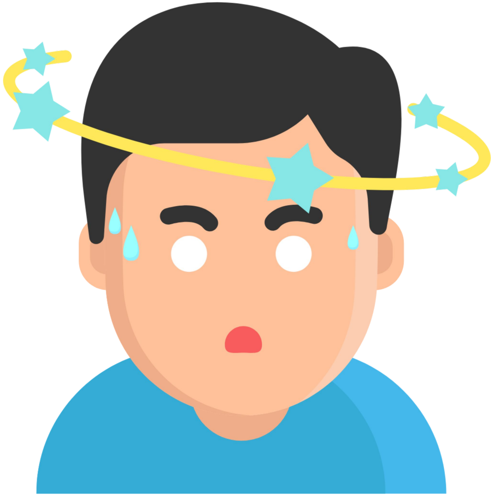
- Styles are defined using a set of rules
- Each rule
  - begins with a selector to select the element(s) onto which the rule is applied.
  - Specifies a group of properties to apply to the element(s).

```
selectorA {                     Rule 1
    property1: value;
    property2: value;
}

selectorB {                     Rule 2
    property1: value;
    property2: value;
}
```

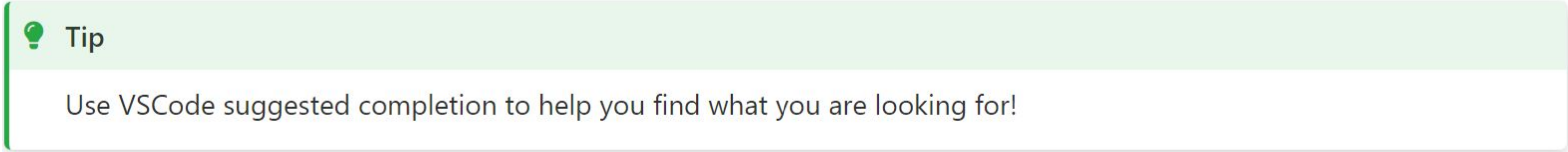
Complete list of CSS properties:

https://www.w3.org/Style/CSS/all-properties.en.html

Too many CSS properties to memorize



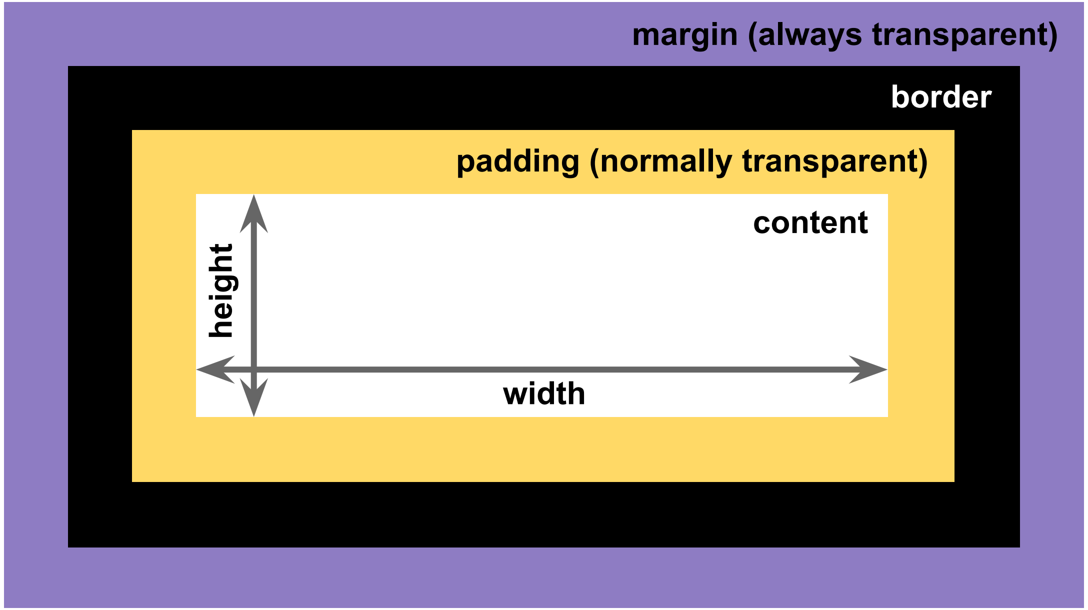
Too many CSS properties to memorize



Tip

Use VSCode suggested completion to help you find what you are looking for!

CSS Box Model

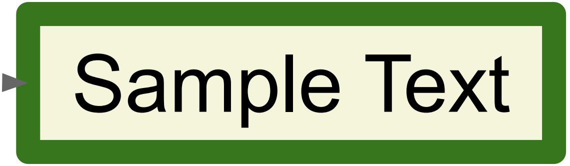


background-color paints the content, padding, and border

CSS Box Model: Padding (inside the border)

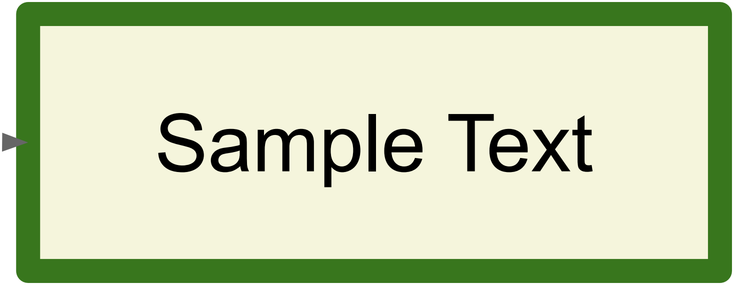
```
<span>Sample Text</span>
```

```
span {
    padding: 4px;
    border: 12px solid green;
    background: beige;
}
```



Sample Text

```
span {
    padding: 16px;
    border: 12px solid green;
    background: beige;
}
```

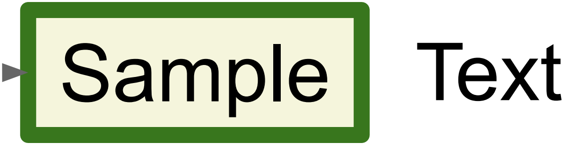


Sample Text

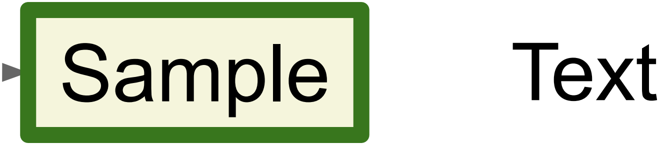
CSS Box Model: Margin (outside the border)

```
<span>Sample</span> Text
```

```
span {
    margin-right: 2px;
    border: 8px solid green;
    background: beige;
}
```



```
span {
    margin-right: 8px;
    border: 4px solid green;
    background: beige;
}
```



CSS Colors

DeepSkyBlue	Gold	DarkRed
Beige	Orange	ForestGreen
HotPink	Olive	YellowGreen

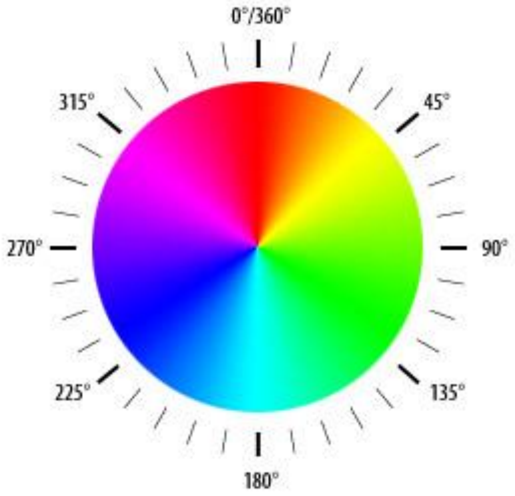
140 standard names

https://www.w3schools.com/colors/colors_names.asp

CSS Colors

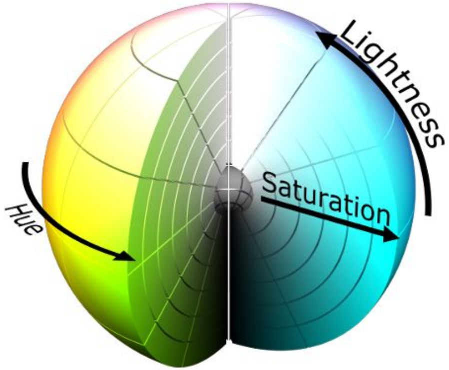
- RGB (0-255 per color) & **Alpha Transparency**
  - rgb(**155**, **138**, **73**)
  - rgba(**155**, **138**, **73**, **0.6**)
- Hex String (00-FF per color)
  - #**C**55 or #**F**CA9
  - #**9B**8A49 or #**9B**8A493F
- HSL
  - **hsl(20, 85%, 30%)** or **hsla(20, 85%, 35%, 0.7)**
  - Benefit: easy to generate shades of a particular color **(in code)**

HSL Colorspace



Hue

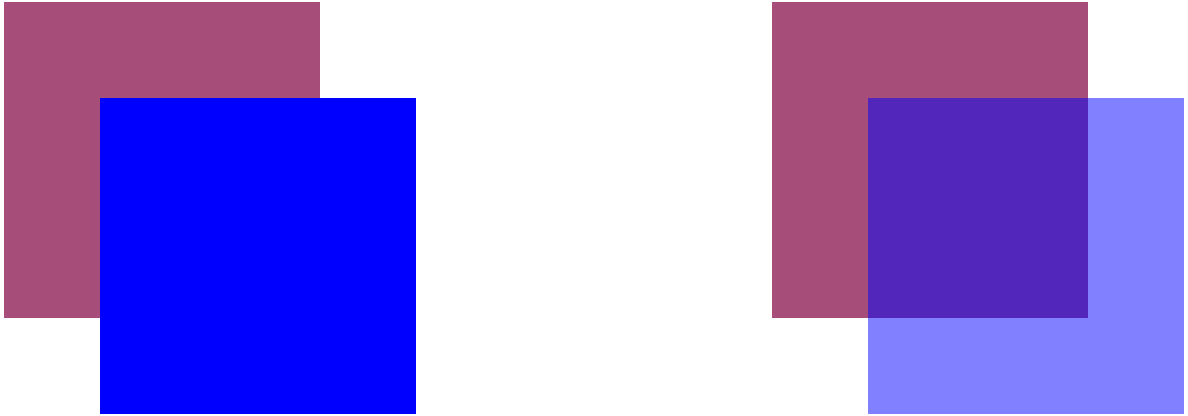
	Description	Range of values
Hue	Color Tone	Red:0, Green:120, Blue:240
Saturation	How much “ink” in your paint	0%: no ink, 100%: max ink
Lightness	How much light available when you are viewing the color	0%: no light 100%: infinite amount of light



HSL Color Picker (CodePen)

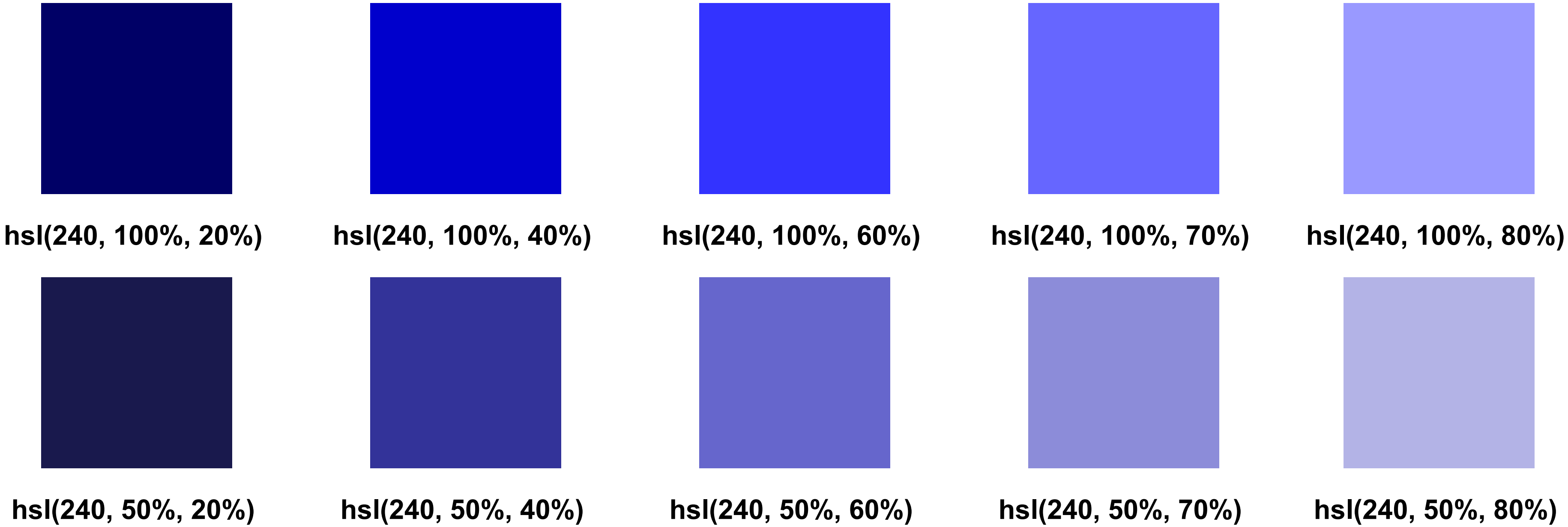
YUI HSL Color Picker

Color Transparency



	Opaque Blue	50% Transparent Blue
RGB	rgb(0, 0, 255, 1.0)	rgb(0, 0, 255, 0.5)
Hex String	#0000FFFF	#0000FF7F
HSL	hsl(240, 100%, 50%, 1.0)	hsl(240, 100%, 50%, 0.5)

HSL Practical Use: shade of color tones



hsl(240, 100%, 20%)

hsl(240, 100%, 40%)

hsl(240, 100%, 60%)

hsl(240, 100%, 70%)

hsl(240, 100%, 80%)

hsl(240, 50%, 20%)

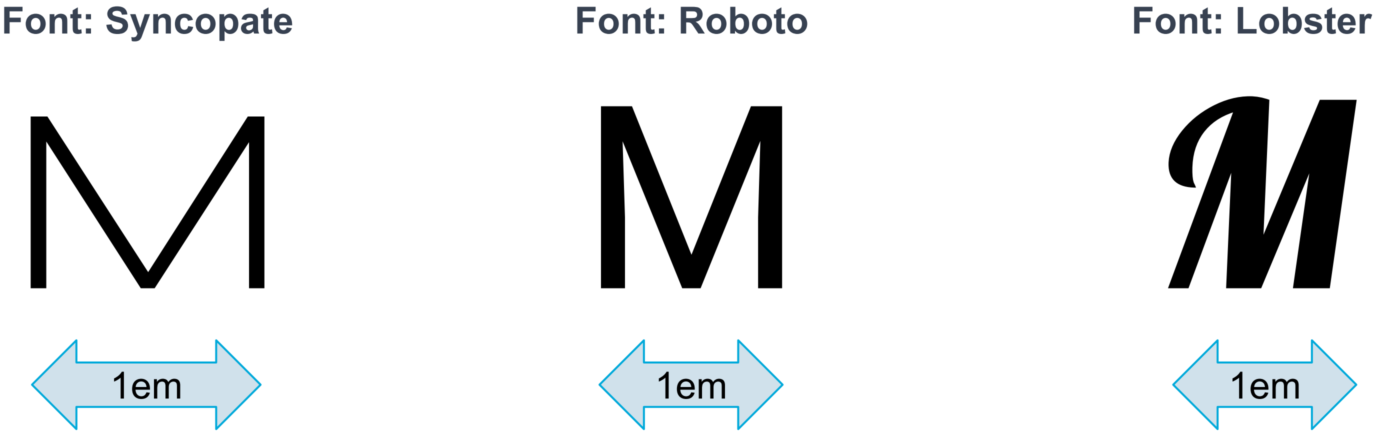
hsl(240, 50%, 40%)

hsl(240, 50%, 60%)

hsl(240, 50%, 70%)

hsl(240, 50%, 80%)

Font Size: 1em



Font: Syncopate

1em

Font: Roboto

1em

Font: Lobster

1em

1em: the width of uppercase M in the current font (traditional interpretation)
1em: the width of the current font (modern typography interpretation)

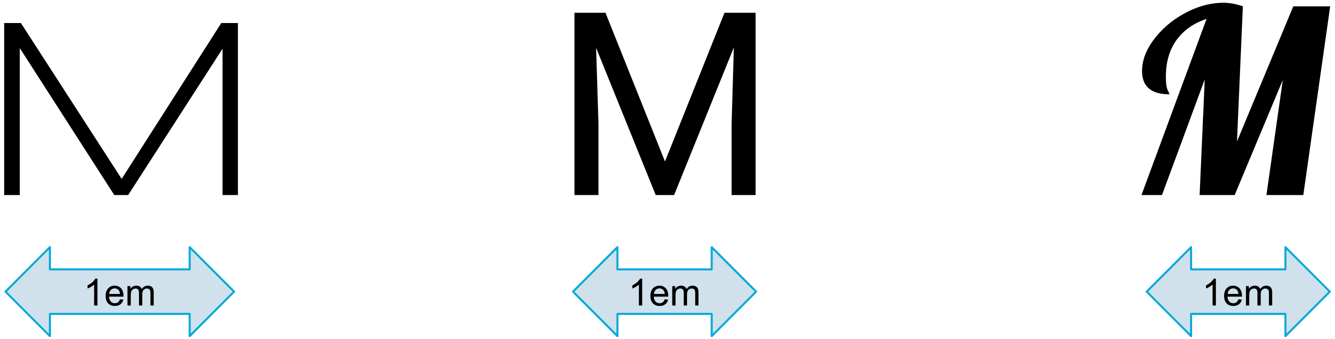
1 em: relative to the nearest parent’s font
1 rem: relative to the root font

Font Size: 1em

Font: Syncopate

Font: Roboto

Font: Lobster



1em

1em

1em

1em: the width of uppercase M in the current font (traditional interpretation)
1em: the width of the current font (modern typography interpretation)

1 em: relative to the nearest parent's font
1 rem: relative to the root font

Use "em" for setting spacing around your text

Applying CSS to HTML

HTML	CSS	Scope of Application

`<!-- by unique id -->`
`<tag id=”ticket”>content</tag>`
	`#ticket {`
	`    padding-left: 2em;`
	`}`
		Only to one element `#ticket`

`<!-- by tag name -->`
`<xyz>content</xyz>`
	`xyz {`
	`    font-weight: bold;`
	`}`
		All `<xyz>` tags in the document

`<!-- by class name -->`
`<xyz class=”weekend”>content</xyz>`
	`.weekend {`
	`    border: 2px solid brown;`
	`}`
		All `.weekend` class in the document

`<!-- by other attributes -->`
`<xyz anyattr=”somevalue”>content</xyz>`
	`[anyattr] {`
	`    background-color: white`
	`}`
		All tags with this attribute `anyattr` in the document

Example: Using CSS Selectors

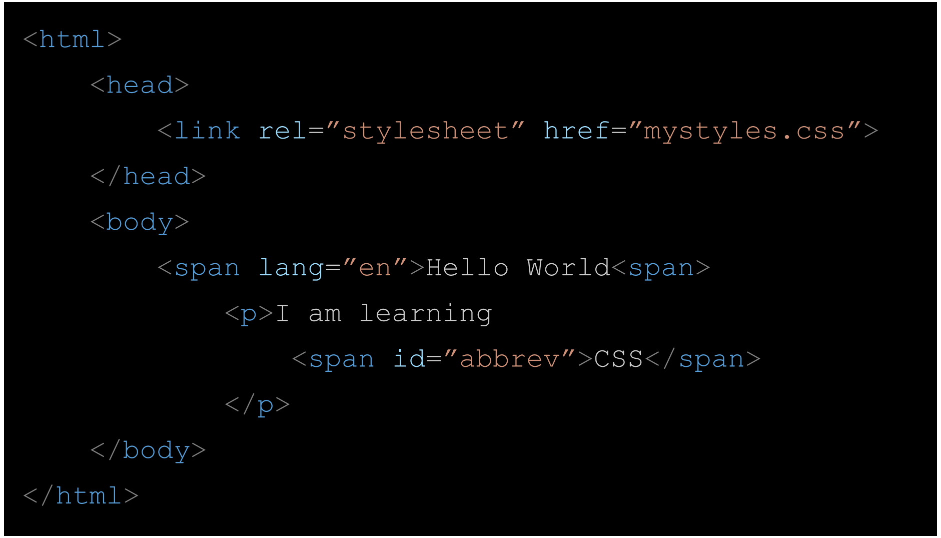
```
<html>
    <head>
        <link rel=”stylesheet” href=”mystyles.css”>
    </head>
    <body>
        <span lang=”en”>Hello World<span>
            <p>I am learning
                <span id=”abbrev”>CSS</span>
            </p>
    </body>
</html>
```

```
#abbrev {
    border-color: red
}
```

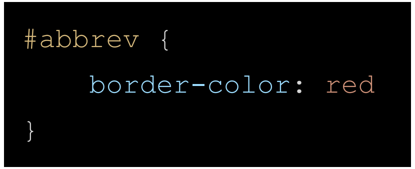
Hello World

I am learning   CSS

Example: Using CSS Selectors



```
<html>
    <head>
        <link rel=”stylesheet” href=”mystyles.css”>
    </head>
    <body>
        <span lang=”en”>Hello World<span>
            <p>I am learning
                <span id=”abbrev”>CSS</span>
            </p>
    </body>
</html>
```

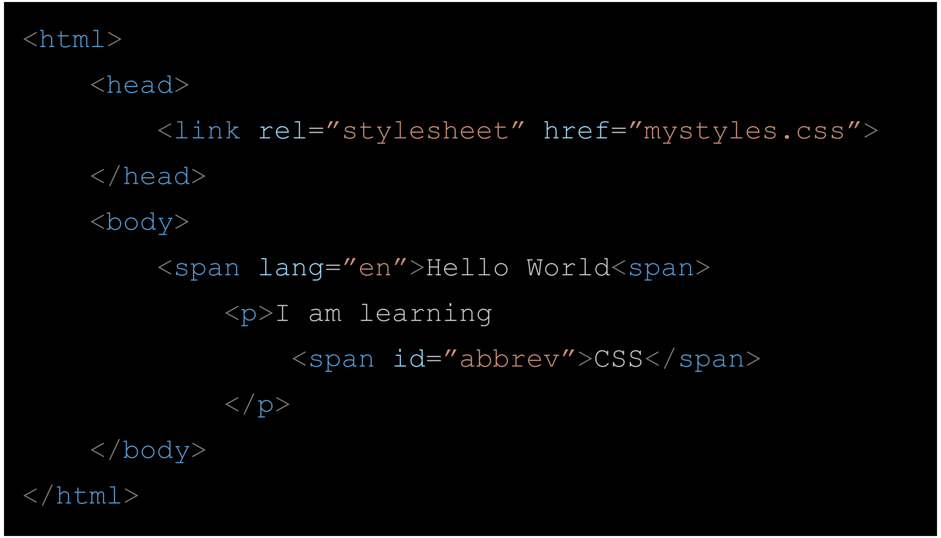


```
#abbrev {
    border-color: red
}
```

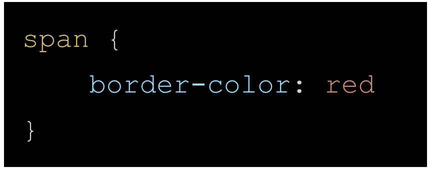
Hello World

I am learning CSS

Example: Using CSS Selectors



```
<html>
    <head>
        <link rel=”stylesheet” href=”mystyles.css”>
    </head>
    <body>
        <span lang=”en”>Hello World<span>
            <p>I am learning
                <span id=”abbrev”>CSS</span>
            </p>
    </body>
</html>
```

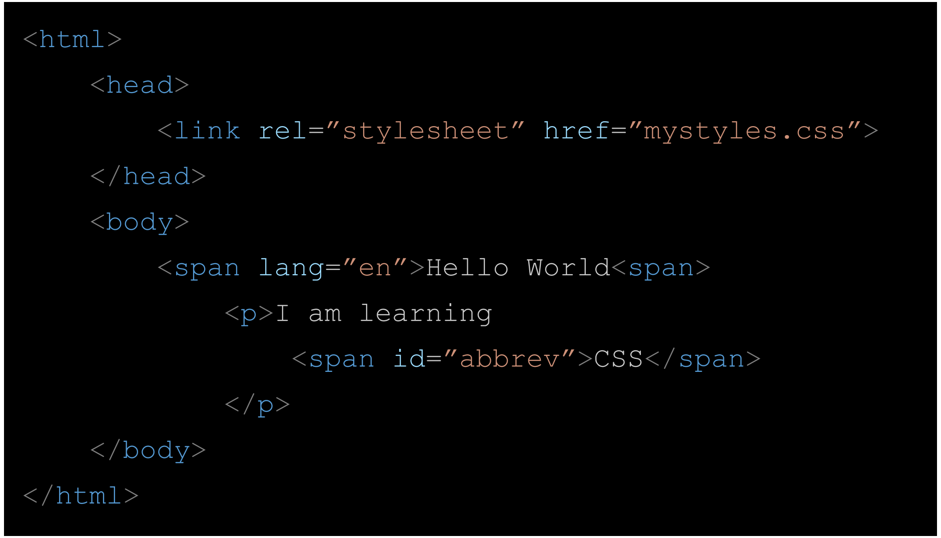


```
span {
    border-color: red
}
```

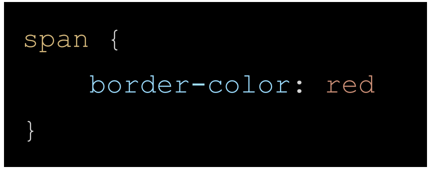
Hello World

I am learning   CSS

Example: Using CSS Selectors



```
<html>
    <head>
        <link rel=”stylesheet” href=”mystyles.css”>
    </head>
    <body>
        <span lang=”en”>Hello World<span>
            <p>I am learning
                <span id=”abbrev”>CSS</span>
            </p>
    </body>
</html>
```



```
span {
    border-color: red
}
```

Hello World

I am learning CSS

Example: Using CSS Selectors

```
<html>
    <head>
        <link rel=”stylesheet” href=”mystyles.css”>
    </head>
    <body>
        <span lang=”en”>Hello World<span>
            <p>I am learning
                <span id=”abbrev”>CSS</span>
            </p>
    </body>
</html>
```

```
[lang] {
    border-color: red
}
```

Hello World

I am learning   CSS

Example: Using CSS Selectors

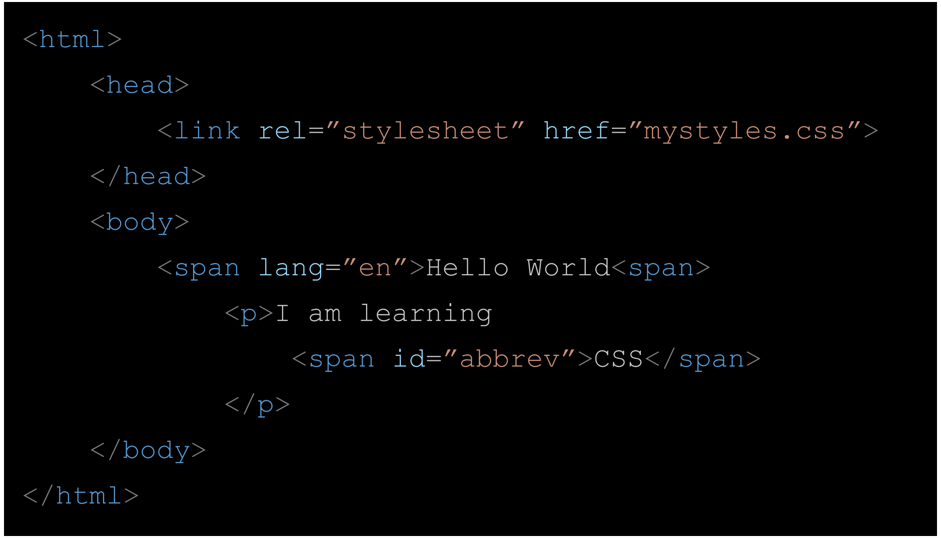
```
<html>
    <head>
        <link rel=”stylesheet” href=”mystyles.css”>
    </head>
    <body>
        <span lang=”en”>Hello World<span>
            <p>I am learning
                <span id=”abbrev”>CSS</span>
            </p>
    </body>
</html>
```

```
[lang] {
    border-color: red
}
```

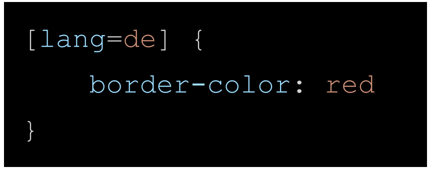
Hello World

I am learning   CSS

Example: Using CSS Selectors



```
<html>
    <head>
        <link rel=”stylesheet” href=”mystyles.css”>
    </head>
    <body>
        <span lang=”en”>Hello World<span>
            <p>I am learning
                <span id=”abbrev”>CSS</span>
            </p>
    </body>
</html>
```



```
[lang=de] {
    border-color: red
}
```

Hello World

I am learning   CSS

Example: Using CSS Selectors

```
<html>
    <head>
        <link rel=”stylesheet” href=”mystyles.css”>
    </head>
    <body>
        <span lang=”en”>Hello World<span>
            <p>I am learning
                <span id=”abbrev”>CSS</span>
            </p>
    </body>
</html>
```

```
[lang=de] {
    border-color: red
}
```

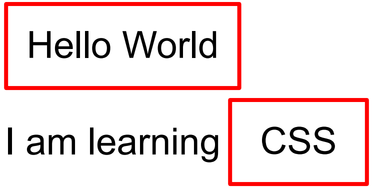
Hello World

I am learning   CSS

Example: Using CSS Selectors

```
<html>
    <head>
        <link rel=”stylesheet” href=”mystyles.css”>
    </head>
    <body>
        <span lang=”en”>Hello World<span>
            <p>I am learning
                <span id=”abbrev”>CSS</span>
            </p>
    </body>
</html>
```

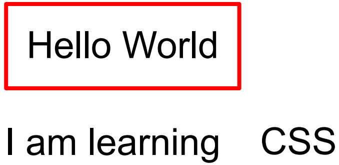
```
span {
    border-color: red
}
```



Hello World

I am learning CSS

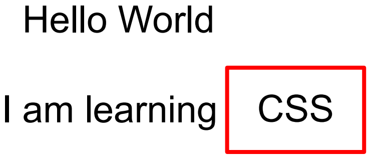
```
[lang] {
    border-color: red
}
```



Hello World

I am learning CSS

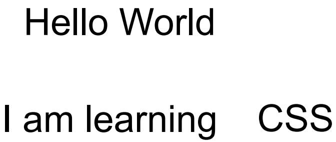
```
#abbrev {
    border-color: red
}
```



Hello World

I am learning CSS

```
[lang=de] {
    border-color: red
}
```



Hello World

I am learning CSS

CSS Selector Specificity

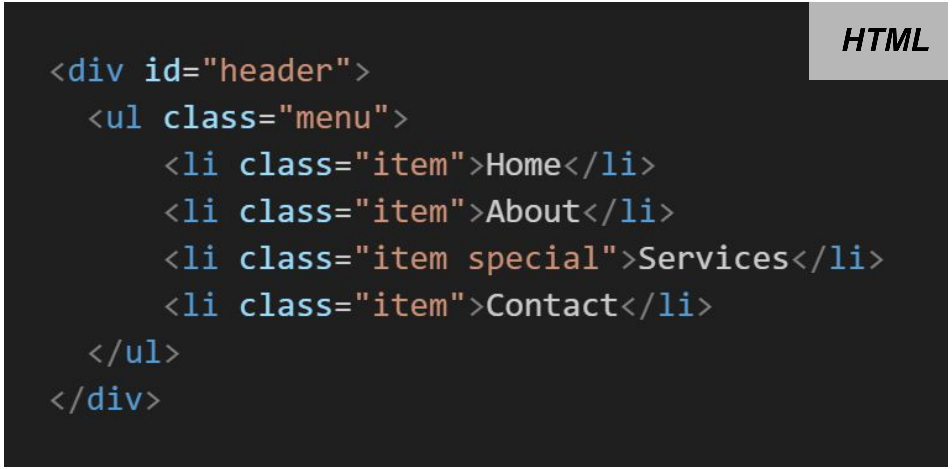
```
/* In mystyles.css  */
div {
    background: red;
}

#top {
    background: green;
}

.warn {
    background: yellow;
}
```

```
<!-- In HTML -->
<div id=”top” class=”warn”>
    Sample
</div>
```

Which selector wins?

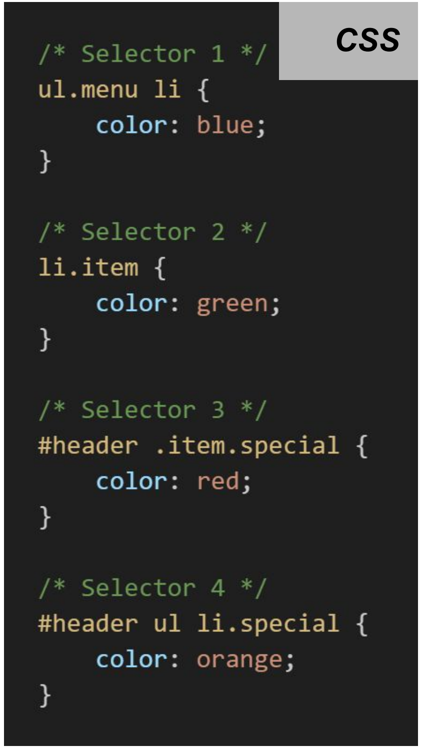
Specificity Calculator
(Higher score wins)

CSS Selector Specificity Exercise



HTML

```html
<div id="header">
  <ul class="menu">
      <li class="item">Home</li>
      <li class="item">About</li>
      <li class="item special">Services</li>
      <li class="item">Contact</li>
  </ul>
</div>
```



CSS

```css
/* Selector 1 */
ul.menu li {
    color: blue;
}

/* Selector 2 */
li.item {
    color: green;
}

/* Selector 3 */
#header .item.special {
    color: red;
}

/* Selector 4 */
#header ul li.special {
    color: orange;
}
```

Practice

[Specificity Calculator]