

Web Dev Day 2: HTML & CSS

Website Development
UBC | Coding Pals

Tables

The syntax for table is very similar to the syntax for a list, but there are a few more elements.

For an example code for the table element, refer to next slide.

Tag	Description
<u><table></u>	Defines a table
<u><th></u>	Defines a header cell in a table
<u><tr></u>	Defines a row in a table
<u><td></u>	Defines a cell in a table
<u><caption></u>	Defines a table caption
<u><colgroup></u>	Specifies a group of one or more columns in a table for formatting
<u><col></u>	Specifies column properties for each column within a <colgroup> element
<u><thead></u>	Groups the header content in a table
<u><tbody></u>	Groups the body content in a table
<u><tfoot></u>	Groups the footer content in a table

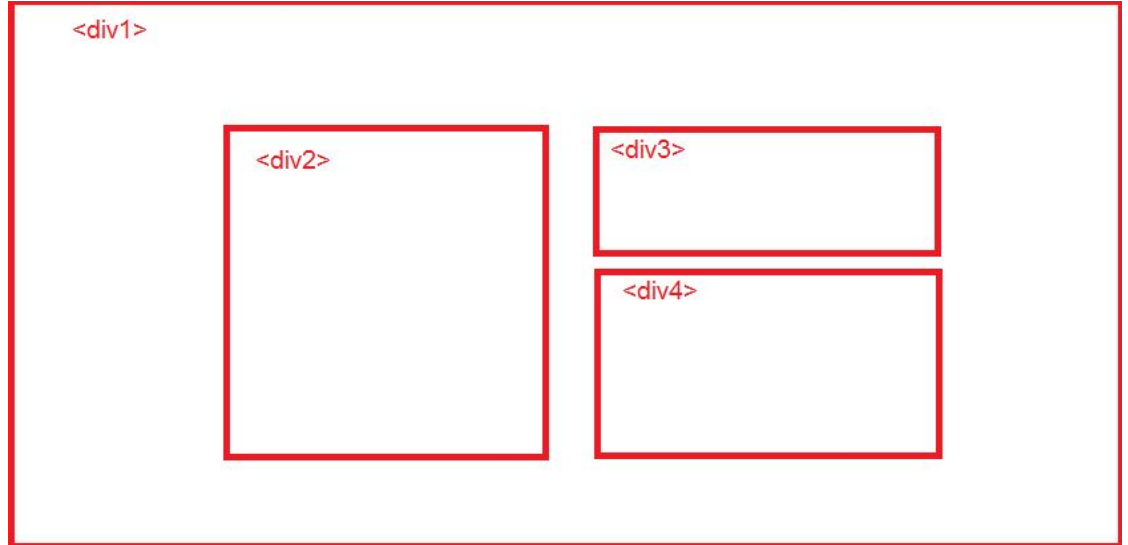
Table Syntax

```
<table>
  <thead>
    <tr>
      <th scope="col">Name</th>
      <th scope="col">Age</th>
      <th scope="col">Occupation</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <td>Gary</td>
      <td>17</td>
      <td>Student</td>
    </tr>
    <tr>
      <td>Davis</td>
      <td>17</td>
      <td>Student</td>
    </tr>
  </tbody>
</table>
```

Name	Age	Occupation
Gary	17	Student
Davis	17	Student

Divs and Spans

- Inline vs Block elements
 - `<a>` is an example of inline, `<p>` is an example of block
- Divs are for block elements while Spans are for inline elements
- Divs and spans define “sections” of HTML to group it all under one category



IDs and Classes

IDs and Classes are used to assign an identifier to an HTML tag

Referenced when styling specific elements

- IDs can only be used once
- Classes can be used for multiple tags

Will go more into detail for CSS

Styling HTML

We can style HTML directly in the tag as an attribute:

```
<p style="color:red; font-size: 24px">Demo  
Text</p>
```

Will go more into detail for CSS

Demo Text

Character Entities

Lots of character are used in the HTML language, but they can be easily misinterpreted by the computer! For example, a “<” will often be read as a part of a tag instead of a part of a paragraph. This is why we have HTML character entities.

	non-breaking space	 	
<	less than	<	<
>	greater than	>	>
&	ampersand	&	&
"	double quotation mark	"	"
'	single quotation mark (apostrophe)	'	'
¢	cent	¢	¢
£	pound	£	£
¥	yen	¥	¥
€	euro	€	€
©	copyright	©	©
®	registered trademark	®	®

What is CSS?

Cascading Style Sheets is a style sheet language (not actually a programming language) that describes the presentation of an HTML page

It can be incorporated directly into the HTML file, or separately in another file

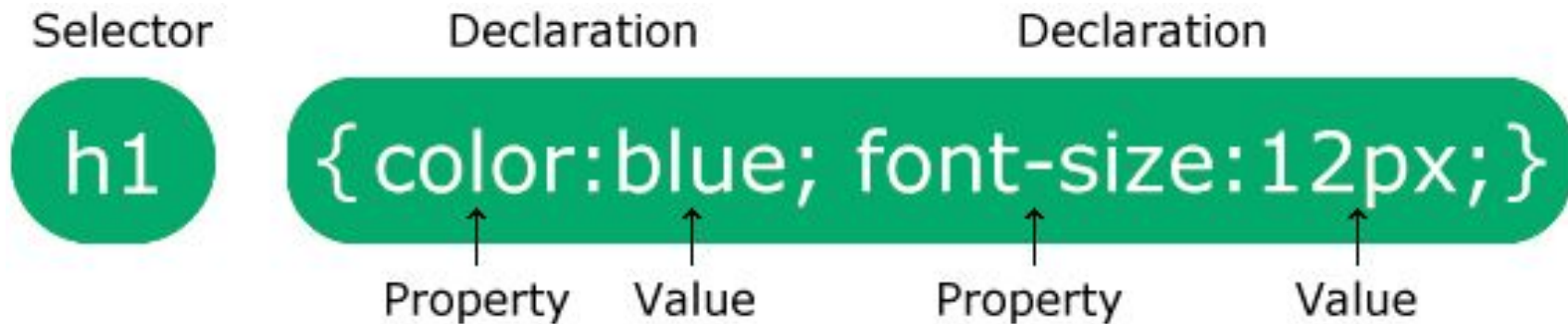
Inline Styling vs External Styling

Writing the style attribute on the HTML tag like we did earlier is known as *Inline Styling*

- While maybe easy for one or two tags, this can become a big hassle for large documents

External styling allows us to control all the styles on a separate file for accessibility and ease of use

CSS Syntax



Creating the styles.css file

In the same folder as your html file, create a new file names “styles.css”

Add the following line to your head section back in your html document:

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

(There is a VSCode shortcut: type “link” and then press tab for autofill)

```
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>
  <link rel="stylesheet" href="styles.css">
</head>
```

Element Selector

In CSS, you can select an entire type of elements to make changes on. For example, you can select the <p> element and change its attributes, which will cause all <p> elements in that webpage to be altered.

```
p{  
    color: blue;  
}
```

Note: the “*” selector targets all elements in the HTML document

Today, I learned about **HTML**

I love coding!

Today is a good day. I got a big fish and a small turtle.

Link without _blank

Link with _blank

Water is also called H₂O

Targeting Specific IDs and Classes

We can also target IDs and Classes to style

- This is why it is important to give HTML tags IDs and Classes

. and #

- Use a “.” before the class name to target a class
- Use a “#” before the ID name to target an ID

It is also possible to assign one element to multiple classes.

Break

Colours

```
.classname {  
    /* the three following lines do the same thing */  
    color: red;  
    color: rgb(255,0,0);  
    color: #ff0000  
}
```

Can also define background colours with “background-color”

Use RGBA to define an opacity value at the end [ex: rgba(255,0,0,0.5)]

Text

Text can be customized in many different way using CSS!

Some example of properties that we can play around with are:

- color
- background-color
- text-align
- text-decoration
- text-transform
- letter-spacing

Width and Height

Certain tags need widths and heights defined

We can define width and height in two ways:

1. Absolute units: px, cm, etc. (do not add a space between the number and the unit)
2. Relative units:
 - a. Rem: relative to the font size of the root element
 - b. Em: relative to the font size of the element

Comments

Defined by `/*` to open a comment and `*/` to close a comment

VSCoDe shortcut of “`ctrl + /`” on Windows or “`command + /`” on Mac also works for comments

Pseudo-class Selector

You can also select elements based on their current states.

This is achieved by the “:” operator. Some examples include :hover, :focus, :visited, :disabled, :active.

The pseudo selector must come after a properly defined selector (example below targets <a> tag)

```
a:hover {  
    Color: red;  
}
```

Today, I learned about **HTML**

I love coding!

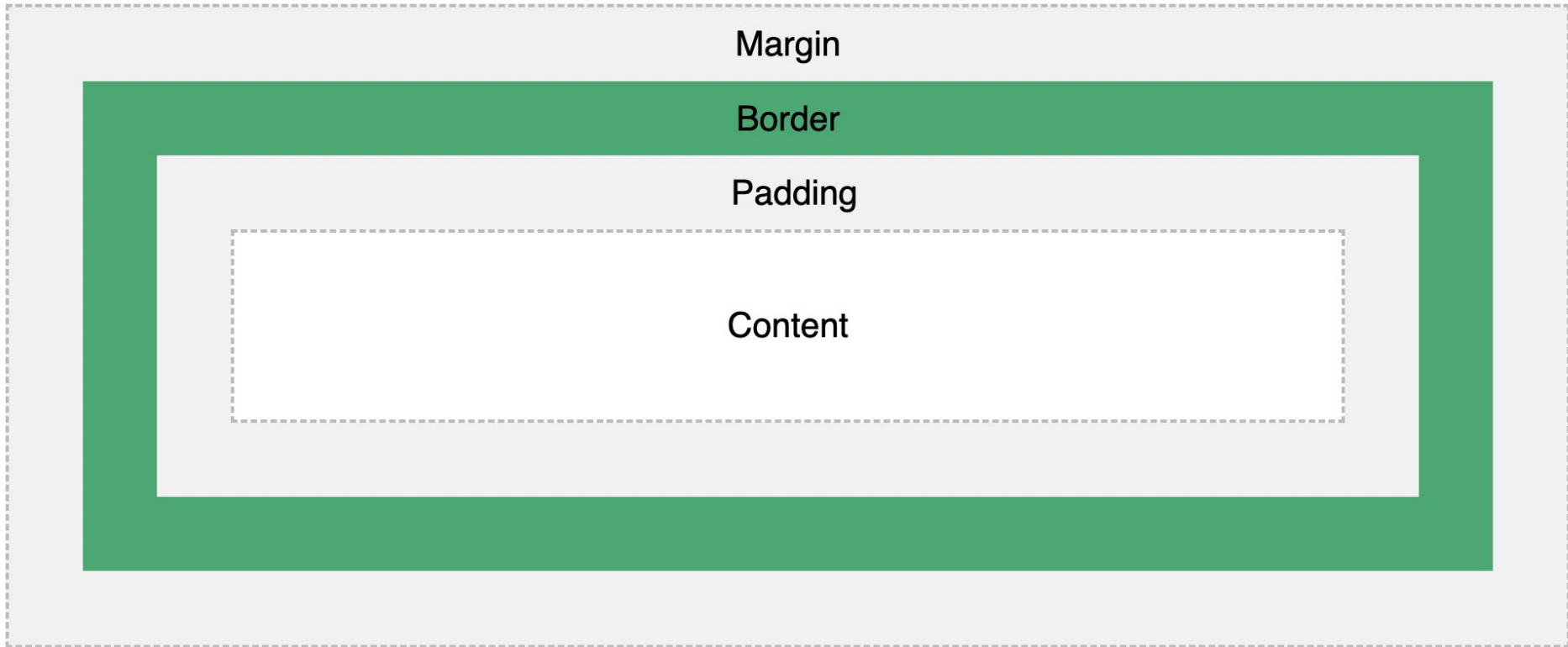
Today is a good day. I got a big fish and a small turtle.

Link without blank

Link with blank

Water is also called H₂O

Box Model (Padding, Border, Margin)



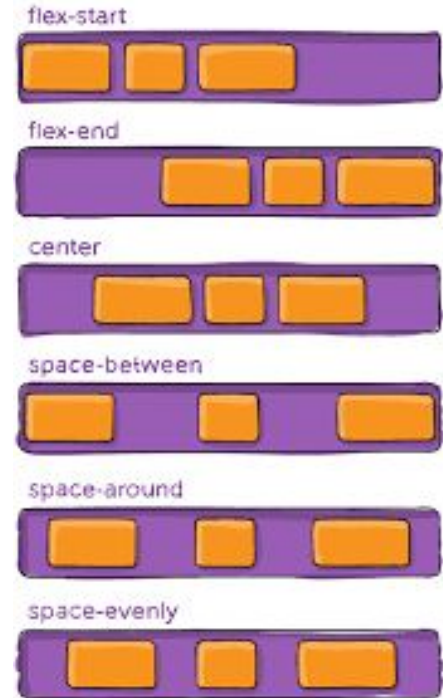
Display

Display lets us manipulate the positioning and placement of elements

We can change inline elements to block and block elements to inline

By using “display: flex” we can change spacing with the “justify-content” property as shown below in the following ways:

- I. **inline/ block/ inline-block** = How items are positioned in relative to other elements
- II. **flex-start** = Default value. Items are positioned at the beginning of the container
- III. **flex-end** = Items are positioned at the end of the container
- IV. **center** = Items are positioned in the center of the container
- V. **space-between** = Items will have space between them
- VI. **space-around** = Items will have space before, between, and after them
- VII. **space-evenly** = Items will have equal space around them



```
p {  
  display: flex;  
  justify-content: center;  
}
```

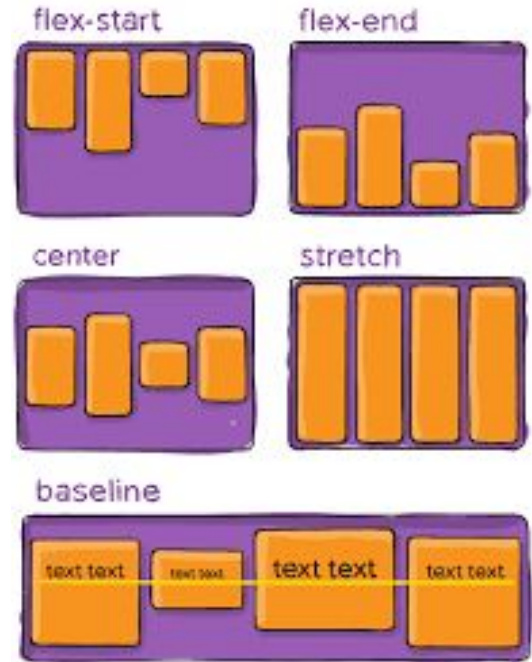
Align Items

Another useful attribute with flex displays is called align-items

For all the elements in your div they need to be aligned

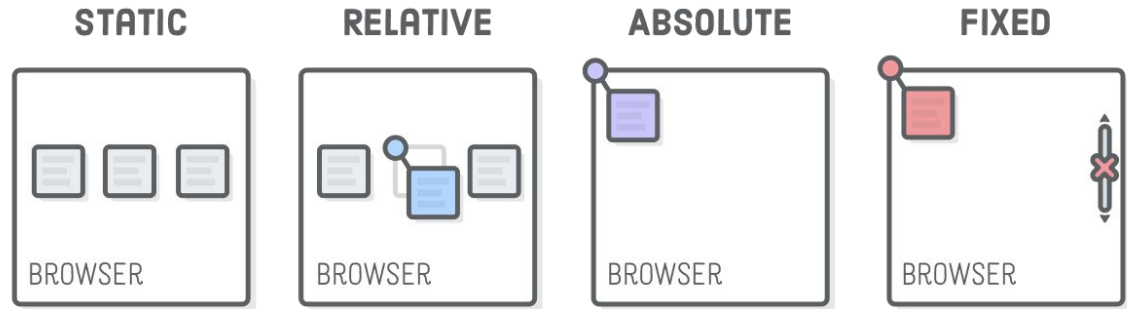
Options:

- I. flex-start = Align everything to the top
- II. flex-end = Align everything to the bottom
- III. center = Align everything in the center
- IV. stretch = Stretches everything to top & bottom
- V. baseline = Align texts to be on the same level



Position

- Static
- Relative
 - top, bottom, left, right
- Absolute
- Fixed
- Sticky
- Z-index



Absolute vs. Fixed vs. Sticky

CSS Battle

<https://cssbattle.dev/>

Create an account and play a few rounds!

Challenge

Challenge

Expand on your “About Me” page made yesterday and add the following elements:

- Images (make a folder called images and reference the path in your src attribute)
- Table with a calendar of the next two weeks on which days you are available, and which days you are not (maybe)
- Favicon for your page (find a logo)
- Optional: YouTube videos

Afterwards, style the elements with varying colours, text, and sizes

- Play around with widths and heights (try to understand the relative units)

Be sure to use divs, spans, IDs, and classes!

Homework

- Finish the Challenge!
- Consider other pages you would like to add to your website (home page, image gallery, contact me, etc.)
- Find images (and also videos, if you have them) to add to your website if you did not already add some in the challenge
- Familiarize yourself with the resources (next slide) as they will be very useful

Resources

MDN Web Docs

“Dictionary” for HTML, CSS, & JS

<https://developer.mozilla.org/>

W3 Schools

Modules that explain all the components

<https://w3schools.com/>

