# Learn Websites:

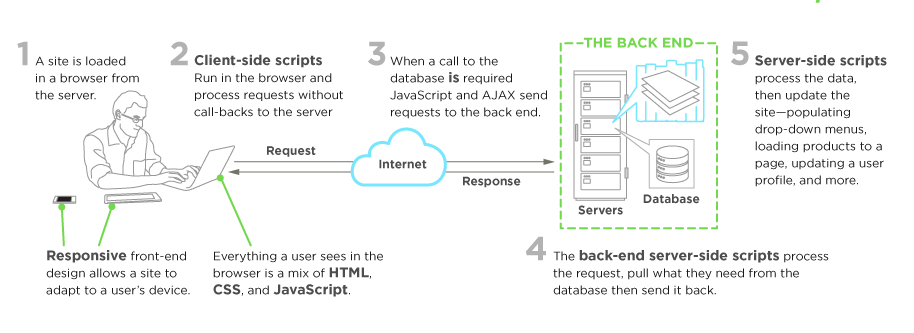
Syllabus My Goal:

<https://www.youtube.com/watch?v=0pThnRneDjw>

<https://github.com/andrews1022/web-development-2020-course-list>

* HTML5 (Semantic elements, attributes)
* CSS fundamentals
* JavaScript
* Responsive to other devices (CSS Framework: Bootstrap)

**Front end developer:** known as client-side development is the practice of producing HTML, CSS and JavaScript for a website or Web Application so that a user can see and interact with them directly.



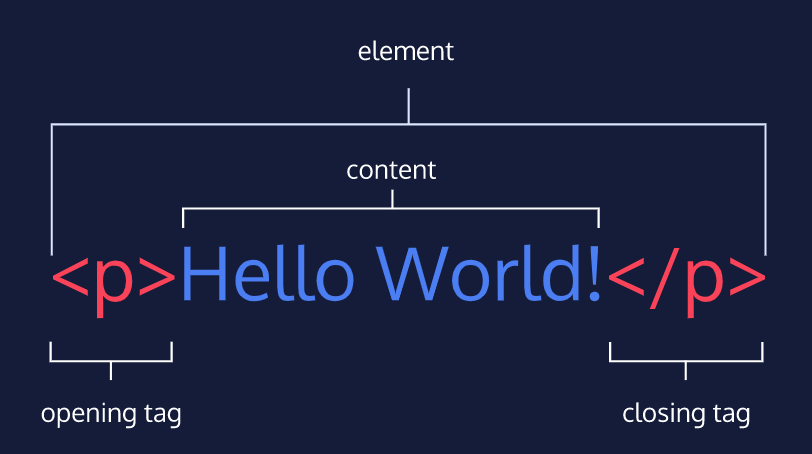
**Front end jobs titles:**

* Web/Front-End User Interface (aka UI) Developer/Engineer
* Front-End Dev. Ops (HTML,CSS, JavaScript)

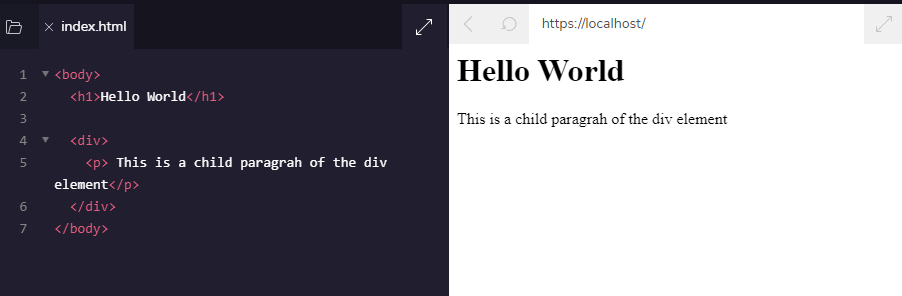
## Part 1: HTML

### HTML Anatomy

* HTML is composed of elements



* Body: Only content inside the opening and closing body tags can be displayed to the screen.



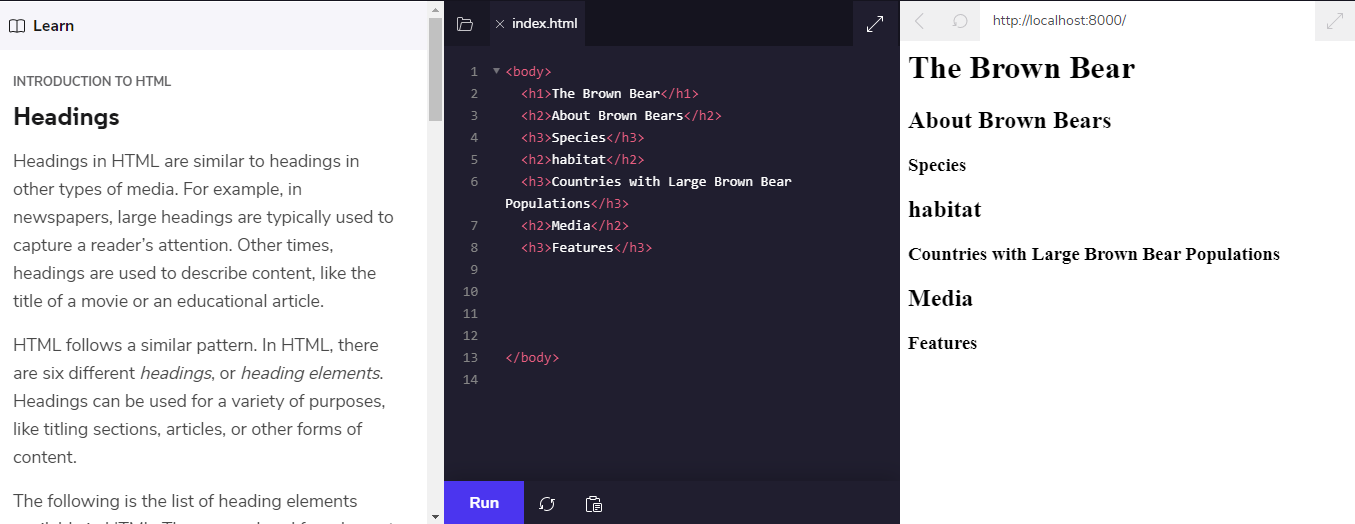
In this example, the <body> element is the parent of the <div> element.

#### **Heading:**

* + Headings in HTML are similar to headings in other types of media.
  + HTML follows a similar pattern. In HTML, there are six different headings, or heading elements.

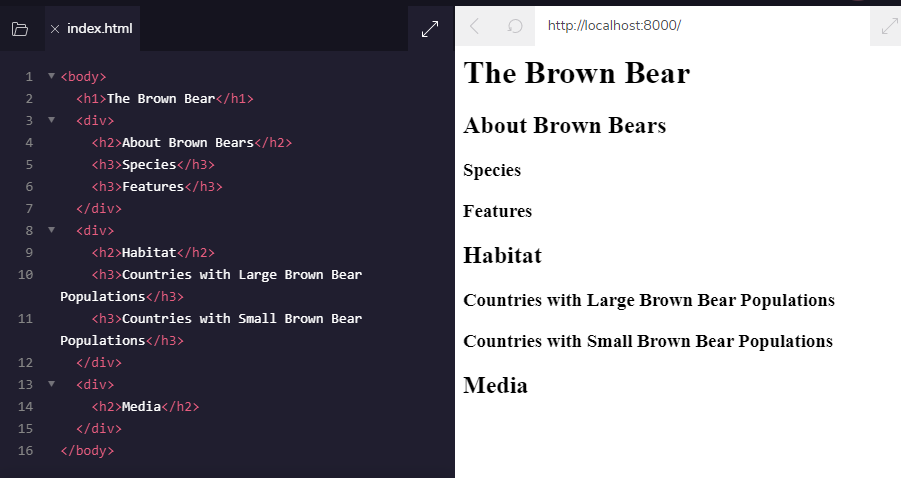
The following is the list of heading elements available in HTML. They are ordered from largest to smallest in size.

1. <h1> — used for main headings. All other smaller headings are used for subheadings.
2. <h2>
3. <h3>
4. <h4>
5. <h5>
6. <h6>



#### **Divs**

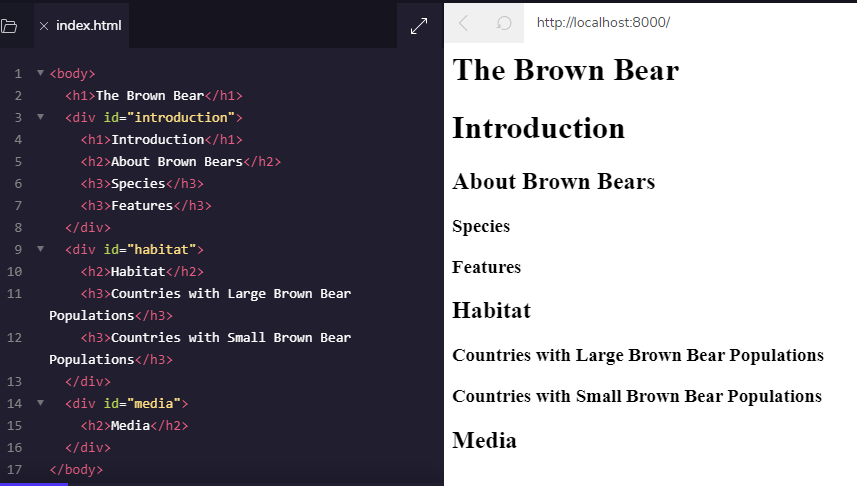
* <div> is short for “division” or a container that divides the page into sections.



#### Attributes

If we want to expand an element’s tag, we can do so using an attribute. Attributes are made up of the following two parts:

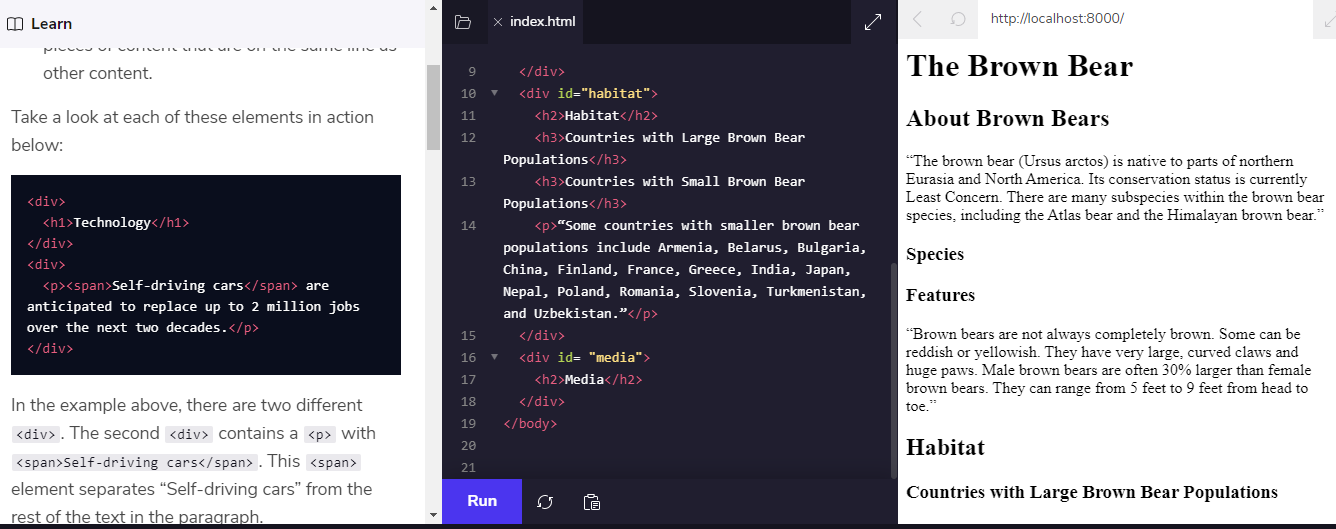
* The name of the attribute
* The value of the attribute
  + One commonly used attribute is the id
  + We can use the id attribute to specify different content (such as <div>s)



### Displaying Text

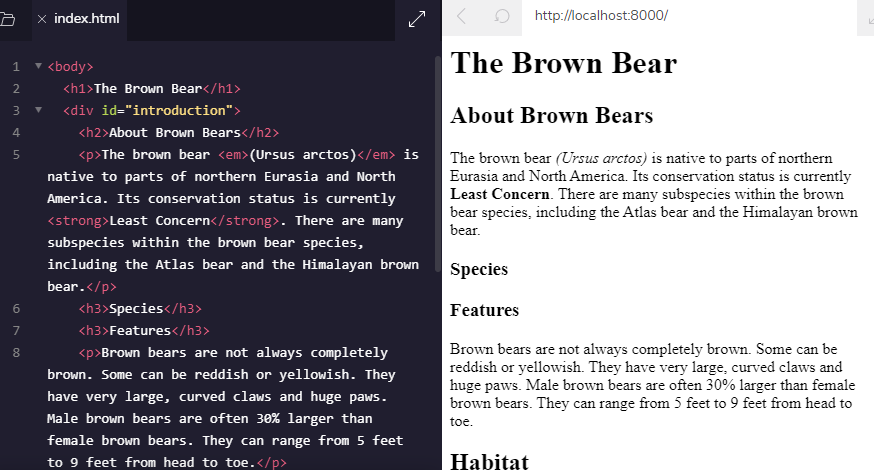
If you want to display text in HTML, you can use a paragraph or span:

* Paragraphs (<p>) contain a block of plain text.
* <span> contains short pieces of text or other HTML. They are used to separate small pieces of content that are on the same line as other content.



#### **Styling Text**

* + The <em> tag emphasizes text (italic)
  + <strong> tag highlights important text (bold)

.

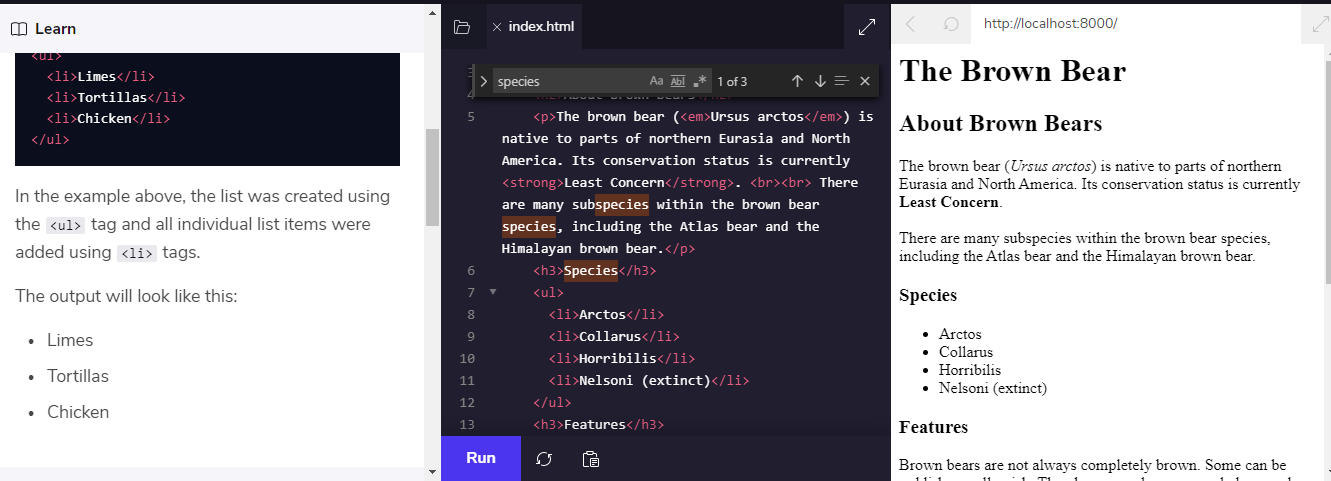
#### Line breaks

* If you are interested in modifying the spacing in the browser, you can use HTML’s line break element: <br>. (Similar to Enter)

### List

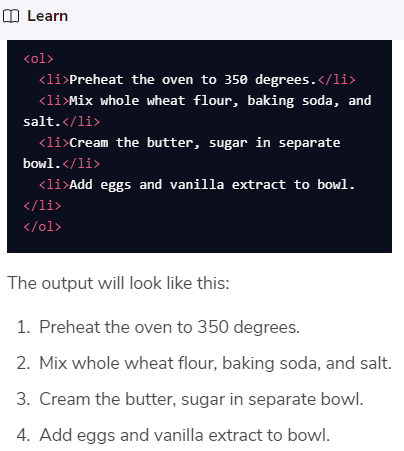
#### Unordered list

* + (<ul>) to create a list of items in no particular order.
  + The <ul> element should not hold raw text and won’t automatically format raw text into an unordered list of items
  + The <li> or list item tag is used to describe an item in a list.



#### Ordered list

* Ordered lists (<ol>) are like unordered lists, except that each list item is numbered.



### Images

* The <img> tag allows you to add an image to a web page.
  + <img src="image-location.jpg" />
  + The src attribute must be set to the image’s source, or the location of the image



#### Images Alt

* : The alt attribute, which means alternative text, brings meaning to the images on our sites.
  + <img src="#" **alt="A field of yellow sunflowers" />.**

The alt attribute also serves the following purposes:

* + If an image fails to load on a web page, a user can mouse over the area originally intended for the image and read a brief description of the image.
  + Visually impaired users often browse the web with the aid of screen reading software. When you include the alt attribute,
  + The alt attribute also plays a role in Search Engine Optimization (SEO), because search engines cannot “see”

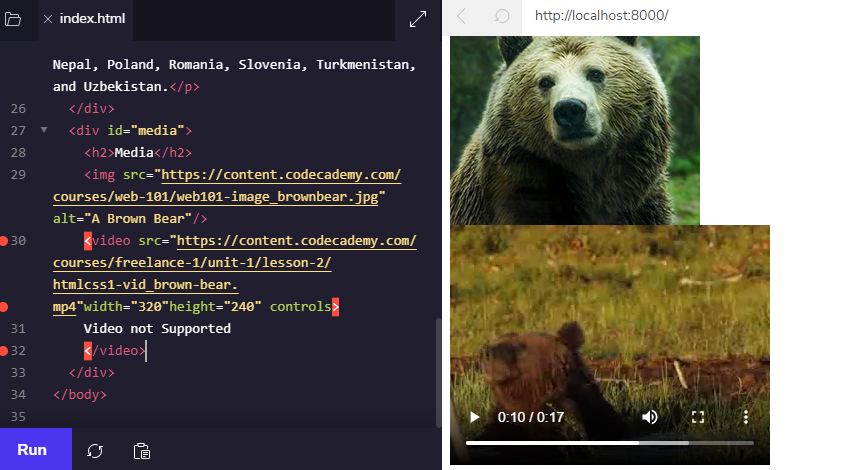
#### Videos

* : <video> tag requires a src attribute with a link to the video source

<video src="myVideo.mp4" width="320" height="240" controls>

Video not supported

</video>



## **HTML Structures**

### <!DOCTYPE html>

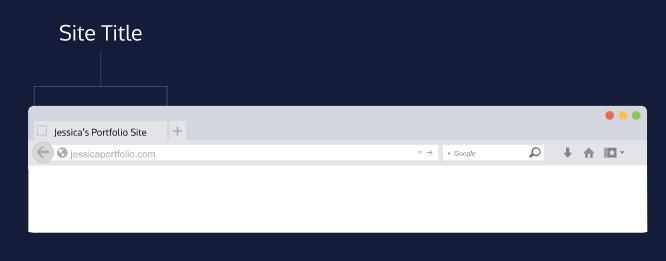
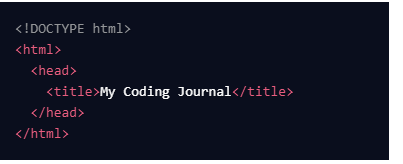
* + HTML files require certain elements to set up the document properly.
  + e using HTML by starting our document with a document type declaration.
  + This declaration is an instruction, and it must be the first line of code in your HTML document
  + For now, the browser will correctly assume that the html in <!DOCTYPE html> is referring to HTML5

### Head

* <head> element is part of this HTML metaphor. It goes above our <body> element. **<head></head>**

### Page Titles

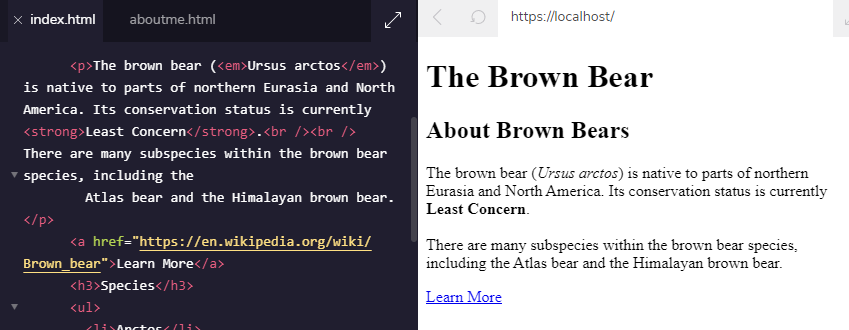
* A browser’s tab displays the title specified in the <title> tag. The <title> tag is always inside of the <head>.



### Linking to Other Web Pages

* + by adding an anchor element <a> and including the text of the link in between the opening and closing tags.
  + The paths provided to the href attribute are often URLs.

**<a href="https://www.wikipedia.org/">This Is A Link To Wikipedia</a>**



#### **Links in a New Window**

* The target attribute can be added directly to the opening tag of the anchor element, just like the href attribute.

**<a href="https://en.wikipedia.org/wiki/Brown\_bear" target="\_blank">The Brown Bear</a>**

#### **Linking to same page**

* In order to link to a *target* on the same page, we must give the target an *id*
* The target link is a string containing the # character and the target element’s id.

Example:

!DOCTYPE html>

<html>

<head>

<title>Brown Bears</title>

</head>

<body>

<a href="./index.html">Brown Bear</a>

<a href="./aboutme.html">About Me</a>

<h1>The Brown Bear</h1>

<ul>

<li><a href="#introduction">Introduction</a></li>

<li><a href="#habitat">Habitat</a></li>

<li><a href="#media">Media</a></li>

</ul>

<div id="introduction">

<h2>About Brown Bears</h2>

<p>The brown bear (<em>Ursus arctos</em>) is native to parts of northern Eurasia and North America. Its conservation status is currently <strong>Least Concern</strong>.<br /><br /> There are many subspecies within the brown bear species, including the

Atlas bear and the Himalayan brown bear.</p>

<a href="https://en.wikipedia.org/wiki/Brown\_bear" target="\_blank">Learn More</a>

<h3>Species</h3>

<ul>

<li>Arctos</li>

<li>Collarus</li>

<li>Horribilis</li>

<li>Nelsoni (extinct)</li>

</ul>

<h3>Features</h3>

<p>Brown bears are not always completely brown. Some can be reddish or yellowish. They have very large, curved claws and huge paws. Male brown bears are often 30% larger than female brown bears. They can range from 5 feet to 9 feet from head to toe.</p>

</div>

<div id="habitat">

<h2>Habitat</h2>

<h3>Countries with Large Brown Bear Populations</h3>

<ol>

<li>Russia</li>

<li>United States</li>

<li>Canada</li>

</ol>

<h3>Countries with Small Brown Bear Populations</h3>

<p>Some countries with smaller brown bear populations include Armenia, Belarus, Bulgaria, China, Finland, France, Greece, India, Japan, Nepal, Poland, Romania, Slovenia, Turkmenistan, and Uzbekistan.</p>

</div>

<div id="media">

<h2>Media</h2>

<a href="https://en.wikipedia.org/wiki/Brown\_bear" target="\_blank"><img src="https://s3.amazonaws.com/codecademy-content/courses/web-101/web101-image\_brownbear.jpg"/></a>

<video src="https://s3.amazonaws.com/codecademy-content/courses/freelance-1/unit-1/lesson-2/htmlcss1-vid\_brown-bear.mp4" height="240" width="320" controls>Video not supported</video>

</div>

</body>

</html>

A screenshot of a cell phone

Description automatically generated

### Whitespace

* As the code in an HTML file grows, it becomes increasingly difficult to keep track of how elements are related. Programmers use two tools to visualize the relationship between elements: *whitespace* and *indentation*.

### Indentation

* The second tool web developers use to make the structure of code easier to read is *indentation*. The spaces are inserted using the space and tab bars on your keyboard.

### Comments

* Comments begin with <!-- and end with -->. Any characters in between will be ignored by your browser.

## **HTML Tables**

* create the table that will contain the data by using the <table> element.

### Table Rows

* The first step in entering data into the table is to add rows using the *table row* element: <tr>.

Example: add two rows of table

<table>

<tr>

</tr>

<tr>

</tr>

</table>

### Table Data

* add data using the *table data* element: <td>.

Example:

<table>

<tr>

</tr>

<tr>

<td>Adam's Greenworks</td>

<td>14</td>

<td>Package Items</td>

</tr>

</table>

Output:

A screenshot of a cell phone

Description automatically generated

### Table Headings

* To add titles to rows and columns, you can use the *table heading* element: <th>.
* Note, also, the use of the scope attribute, which can take one of two values:
  + row - this value makes it clear that the heading is for a row.
  + col - this value makes it clear that the heading is for a column.

Example:

<tr>

<th scope="col">Company name</th>

<th scope="col">Number of Items to Ship</th>

<th scope="col">Next Action</th>

</tr>

A screenshot of a cell phone

Description automatically generated

### Table Borders

* using the border attribute and setting it equal to an integer. This integer would represent the thickness of the border.

A picture containing drawing

Description automatically generated

* can achieve the same table border effect using CSS.

### Spanning Columns

* Data can span columns using the colspan attribute. The attributes accepts an integer

Example:

<tr>

<td colspan="2">Adam's Greenworks</td>

<td>14</td>

<td>Package Items</td>

</tr>

<tr>

A screenshot of a cell phone

Description automatically generated

### Spanning Rows

* The rowspan attribute is used for data that spans multiple rows

### Table Body

* Long tables can be sectioned off using the *table body* element: <tbody>.
* The <tbody> element should contain all of the table’s data, excluding the table headings (more on this in a later exercise).

### Table Head

* <thead> element.
* In the last exercise, the table’s headings were kept inside of the table’s body. When a table’s body is sectioned off, however, it also makes sense to section off the table’s column headings using the <thead> element.

A screenshot of a cell phone

Description automatically generated

Table Footer

* The bottom part of a long table can also be sectioned off using the <tfoot> element.

Example:

<tfoot>

<td>Total</td>

<td>28</td>

<td>Baker's Bike Shop</td>

<td>3</td>

<td>Send Invoice</td>

</tr>

</tfoot>

### Styling with CSS

A screenshot of a cell phone

Description automatically generated

## **HTML FORMS**

### How a Form Works

* Computers need an *HTTP request* to know how to communicate
* The HTTP request instructs the receiving computer how to handle the incoming information. More information can be found in our article about [HTTP requests](https://www.codecademy.com/articles/http-requests).
* The <form> element is a great tool for collecting information, but then we need to send that information somewhere else for processing
* We need to supply the <form> element with both the location of where the <form>‘s information goes and what HTTP request to make. Take a look at the sample <form> below:

A picture containing food, drawing

Description automatically generated

* The <form> element can also contain child elements. For instance

Example: Creating burger menu

<!DOCTYPE html>

<html lang="en" dir="ltr">

<head>

<meta charset="utf-8">

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

<link href="https://fonts.googleapis.com/css?family=Rubik" rel="stylesheet">

<title>HTML Forms</title>

</head>

<body>

<section id="overlay">

<img src="https://s3.amazonaws.com/codecademy-content/courses/web-101/unit-6/htmlcss1-img\_burger-logo.svg" alt="Davie's Burgers Logo" id="logo">

<hr>

<!--Add your code below-->

<form action="practice.html" method="POST">

<h1>Burger menu's</h1>

<p>Chicken burger</p>

</form>

</section>

</body>

</html>

Results:

A screenshot of a cell phone

Description automatically generated

### Text Input

* If we want to create an input field in our <form>, we’ll need the help of the <input> element.
* The <input> element has a type attribute which determines how it renders on the web page and what kind of data it can accept.
* When we create an <input> element with type="text", it renders a text field that users can type into.
* it renders a text field that users can type into. It’s also important that we include a name attribute for the <input> — without the name attribute, information in the <input> won’t be sent when the <form> is submitted.

Example:

A picture containing phone, street

Description automatically generated

Example: Creating a name form inside Davie’s burger

<!DOCTYPE html>

<html lang="en" dir="ltr">

<head>

<meta charset="utf-8">

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

<link href="https://fonts.googleapis.com/css?family=Rubik" rel="stylesheet">

<title>Adding a Text Input</title>

</head>

<body>

<section id="overlay">

<img src="https://s3.amazonaws.com/codecademy-content/courses/web-101/unit-6/htmlcss1-img\_burger-logo.svg" alt="Davie's Burgers Logo" id="logo">

<hr>

<form>

<h1>Login to start creating a burger!</h1>

<!--Add your code below-->

<input type="text" name="username" value="Davie">

</form>

</section>

</body>

</html>

Results

A screenshot of a cell phone

Description automatically generated

### Adding a Label

* For a user to properly identify an <input> we use the appropriately named <label> element.
* to associate a <label> and an <input>, the <input> needs an id attribute

A screenshot of a cell phone

Description automatically generated

Example: add label of text form

<!DOCTYPE html>

<html lang="en" dir="ltr">

<head>

<meta charset="utf-8">

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

<link href="https://fonts.googleapis.com/css?family=Rubik" rel="stylesheet">

<title>Labels</title>

</head>

<body>

<section id="overlay">

<img src="https://s3.amazonaws.com/codecademy-content/courses/web-101/unit-6/htmlcss1-img\_burger-logo.svg" alt="Davie's Burgers Logo" id="logo">

<hr>

<form>

<h1>Login to start creating a burger!</h1>

<!--Add your code below-->

<label for="username">Enter your name

<input type="text" name="username" id="username">

</label>

</form>

</section>

</body>

</html>

Results:

A screenshot of a cell phone

Description automatically generated

### Password Input

* An <input type ="password"> element will replace input text with another character like an asterisk (\*) or a dot (•).

Example:

<form>

<h1>Login to start creating a burger!</h1>

<label for="username">Username:</label>

<input type="text" name="username" id="username">

<br>

<label for="user-pw">Password:</label>

<!--Add your code below-->

<input type="password" id="user-pw" name="user-pw">

</form>

Results:

A screenshot of a cell phone

Description automatically generated

### Number Input

* type="number" for an <input> we can restrict what users type into the input field to just numbers (and a few special characters like -, +, and .)

## **Part 2: CSS**

* language that web developers use to *style* the HTML content
* modifying colors, font types, font sizes, shadows, images, element positioning, and more

### Inline Styles

* To style an HTML element, you can add the style attribute directly to the opening tag

Example: Coloring text

A picture containing drawing

Description automatically generated

* the style attribute is set equal to color: red;, which will set the color of the paragraph text to red within the browser.

Example: Customize font

<p style="font-family:Arial";>Example text.</p>

### The <style> Tag

* CSS can be written between opening and closing <style> tags. To use the <style> element, it must be placed inside of the <head> element.
* it must be placed inside of the <head> element.

Example:

<!DOCTYPE html>

<html>

<head>

<style>

p{

font-family: Arial;

color:red;

font-size:20px;

}

<title>Vacation World</title>

</style>

</head>

<body>

<img src="https://s3.amazonaws.com/codecademy-content/courses/freelance-1/unit-2/explorer.jpeg" />

<h1 class="title">Top Vacation Spots</h1>

<h5>By: Stacy Gray</h5>

<h6>Published: 2 Days Ago</h6>

<p style="font-family: Arial;">The world is full of fascinating places. Planning the perfect vacation involves packing up, leaving home, and experiencing something new.</p>

<h2 class="destination">1. Florence, Italy</h2>

<div class="description">A city-size shrine to the Renaissance, Florence offers frescoes, sculptures, churches, palaces, and other monuments from the richest cultural flowering the world has known. Names from its dazzling historical pastDante, Michelangelo, Galileo, Machiavelliare some of the most resonant of the medieval age. <a href="http://travel.nationalgeographic.com/travel/city-guides/florence-italy/" target="\_blank">Learn More</a>.

<h5>Top Attractions</h5>

<ul>

<li>Museums</li>

<li>Bike Tours</li>

<li>Historical Monuments</li>

</ul>

</div>

<h2 class="destination">2. Beijing, China</h2>

<div class="description">A city in the midst of reinventing itself and continuing to build on the success of the 2008 Summer Olympics, Beijing is a place of frenzied construction. New housing, new roads, and new sports venues seem to spring up overnight. At the same time, the capital of the Peoples Republic of China remains an epicenter of tradition, with the treasures of nearly 2,000 years as the imperial capital still on viewin the famed Forbidden City and in the luxuriant pavilions and gardens of the Summer Palace.

<a href="http://travel.nationalgeographic.com/travel/city-guides/beijing-china/" target="\_blank">Learn More</a>.

<h5>Top Attractions</h5>

<ul>

<li>Biking</li>

<li>Historical Sites</li>

<li>Restaurants and Dining</li>

</ul>

</div>

<h2 class="destination">3. Seoul, South Korea</h2>

<div class="description">The Korean capital is a city of contrasts. Fourteenth-century city gates squat in the shadow of 21st-century skyscrapers, while the broad Han River is back-dropped by granite mountains rising in the city centercomplete with alpine highways speeding around their contours and temples nestling among their crags. Fashionable, gadget-laden youths battle for sidewalk space with fortune-tellers and peddlers, while tiny neighborhoods of traditional cottages contrast with endless ranks of identical apartments.

<a href="http://travel.nationalgeographic.com/travel/city-guides/seoul-south-korea/" target="\_blank">Learn More</a>.

<h5>Top Attractions</h5>

<ul>

<li>Parasailing</li>

<li>Segway Tours</li>

<li>Spas and Resorts</li>

</ul>

</div>

<h2> More Desinations </h2>

<ul>

<li><h4 class="destination">Jackson Hole, Wyoming</h4></li>

<li><h4 class="destination">Cape Town, South Africa</h4></li>

<li><h4 class="destination">La Paz, Bolivia</h4></li>

</ul>

<p>&mdash;Best of luck with your travels, and be sure to send pictures and stories. We"d love to hear them!</p>

</body>

</html>

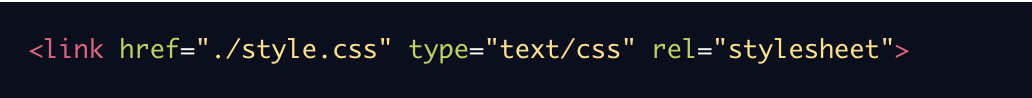
### The .css file

* create a CSS file by using the **.css** file name extension, like so: **style.css**

### Linking the CSS File

* use the <link> element to link HTML and CSS files together.
* <link> element must be placed within the head of the HTML file.
* href — like the anchor element, the value of this attribute must be the address, or path, to the CSS file.
* type — this attribute describes the type of document that you are linking to (in this case, a CSS file). The value of this attribute should be set to text/css.
* rel — this attribute describes the relationship between the HTML file and the CSS file. Because you are linking to a stylesheet, the value should be set to stylesheet.

If the CSS file is stored in the same [directory](https://en.wikipedia.org/wiki/Directory_(computing)) as your HTML file, then you can specify a [relative path](https://en.wikipedia.org/wiki/Path_(computing)#Absolute_and_relative_paths) instead of a URL, like so:



Example: Linking css file with html

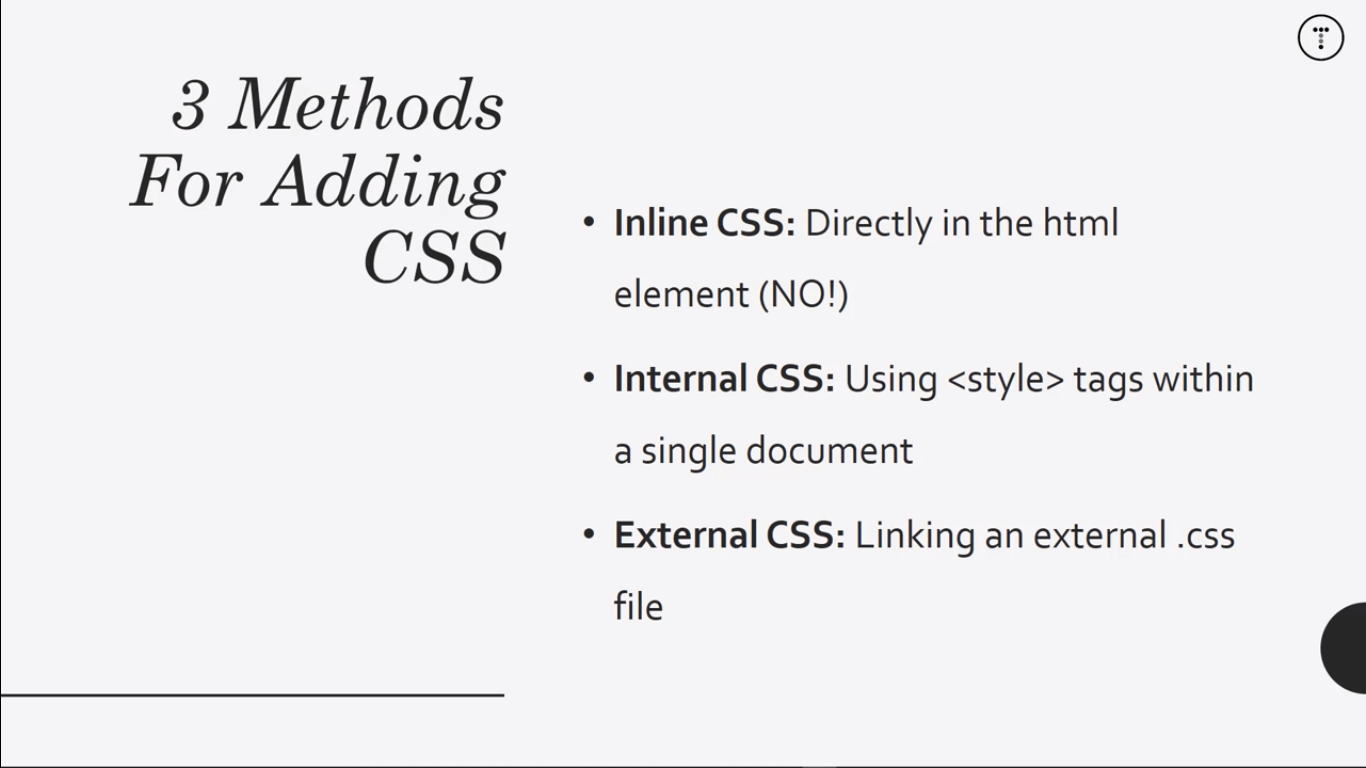
<html>

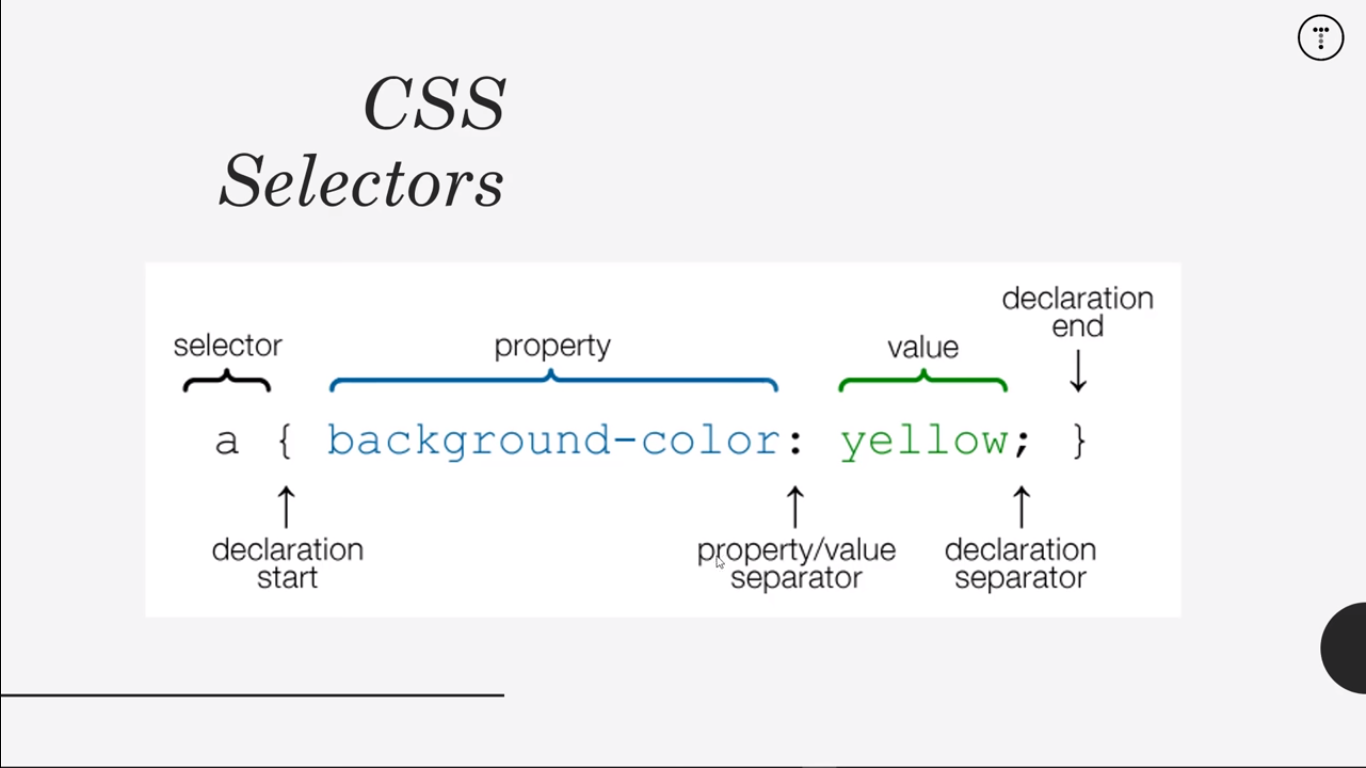
<link href="./style.css" type="text/css" rel="stylesheet">

<head>

<title>Vacation World</title>

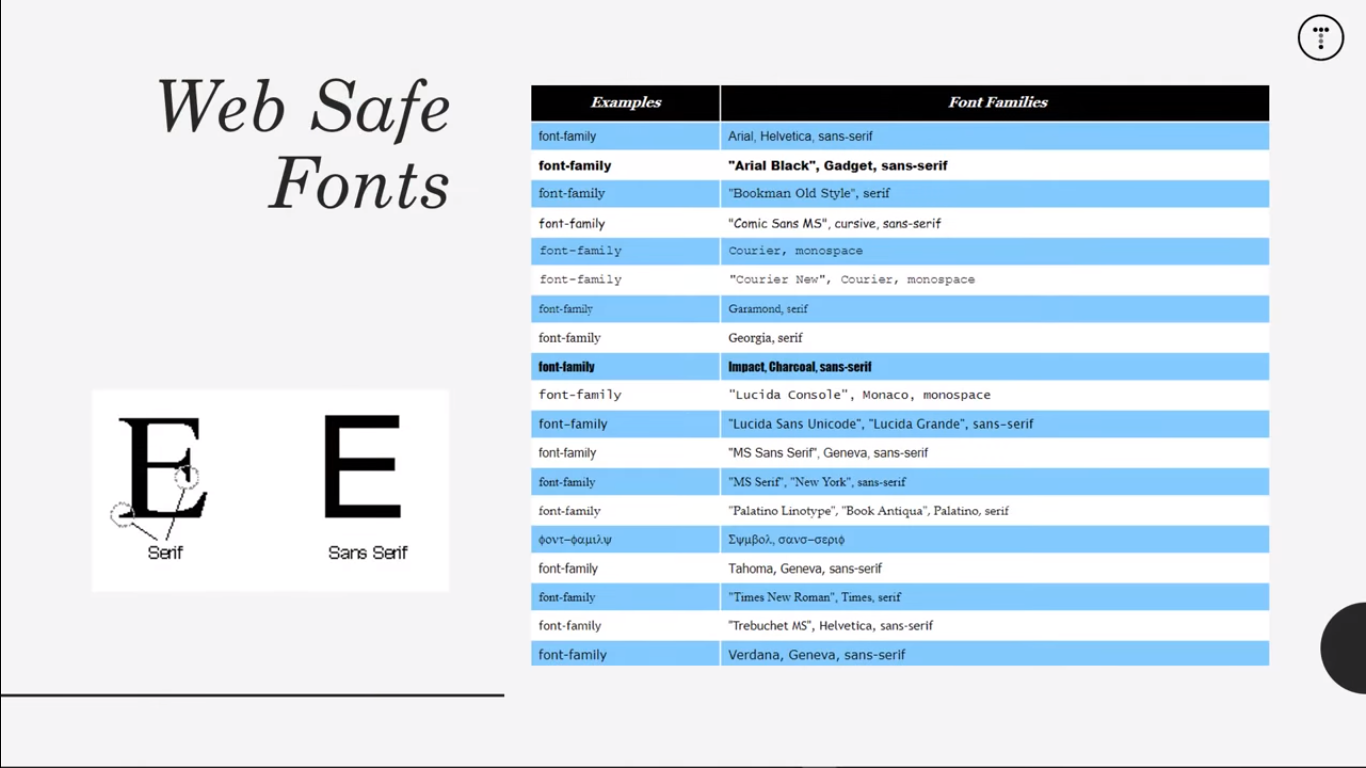
</head>

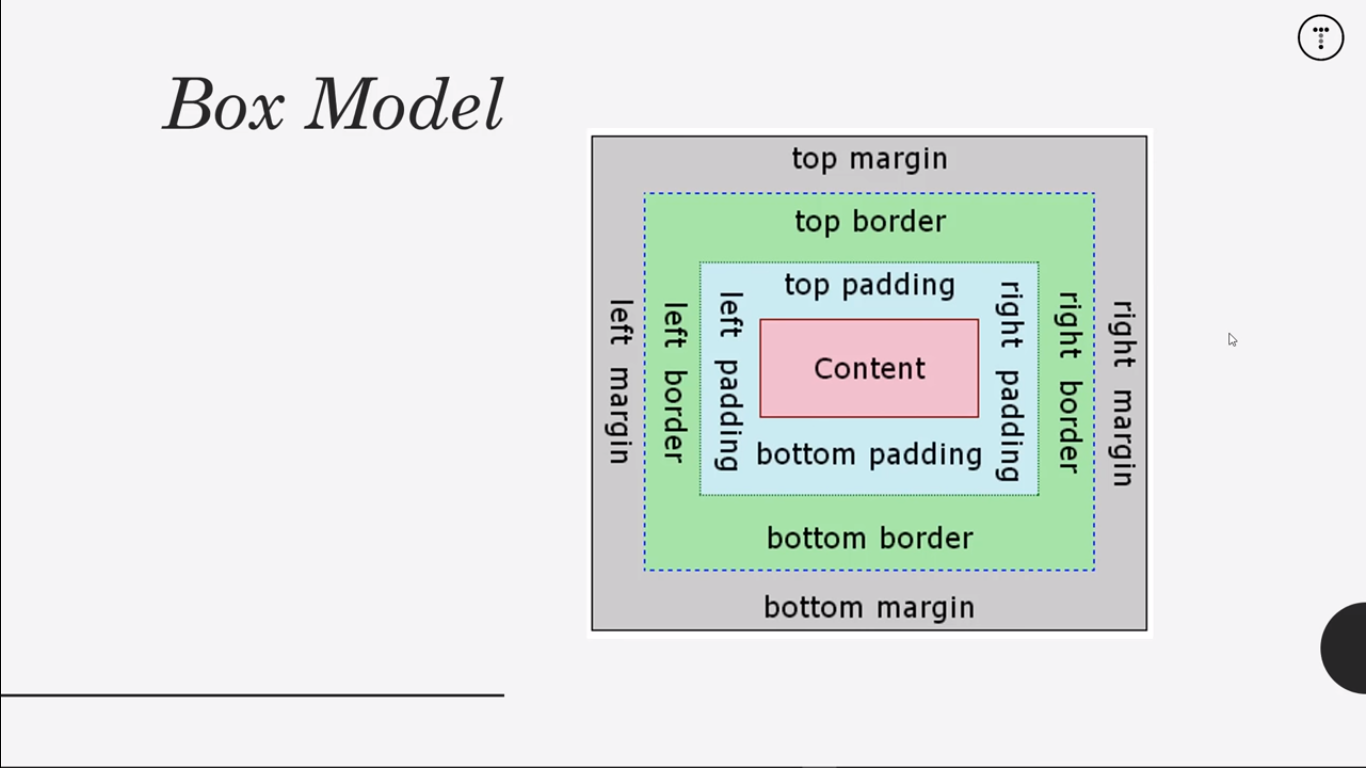






Progress 4 :

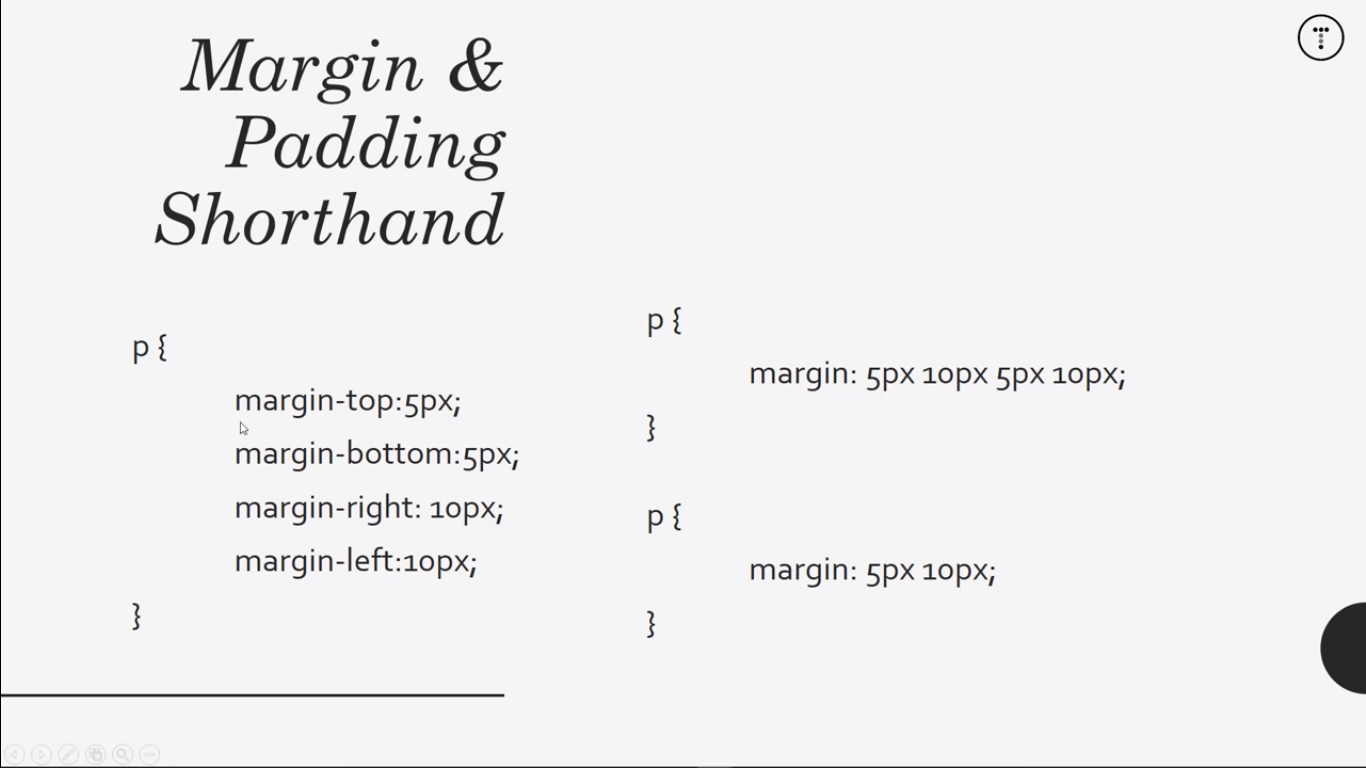


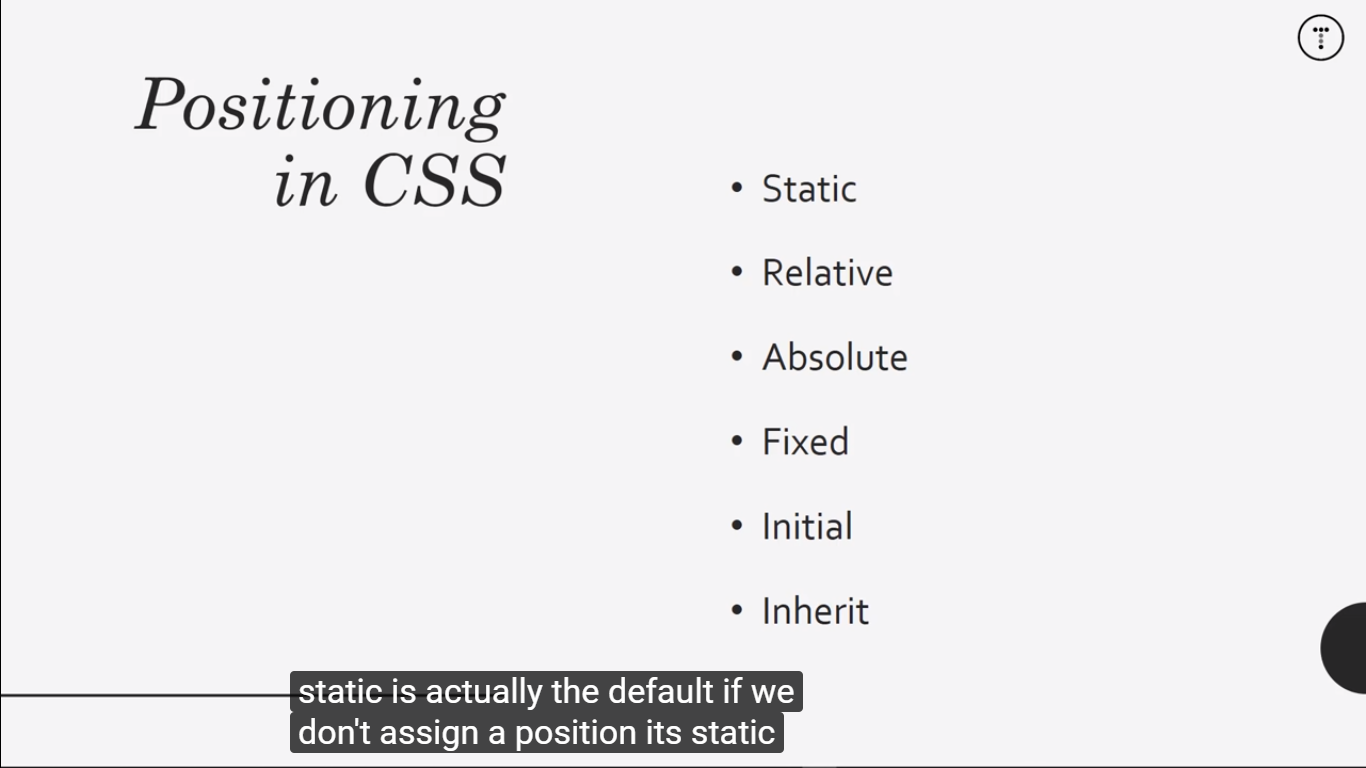


Content : paragraph, a , etc (content)

Padding : space inside (give space text and space)

Margin : space outside





## **Part 3: Introduction to JavaScript**

### Declare JavaScript Variables

* to create or *declare* a variable by putting the keyword **var**
  + var Ourname;
  + var a;
  + a=7;
* Understanding unutilized variablesCalendar

  Description automatically generated
* Case sensitivity

**Graphical user interface, text, application, chat or text message

Description automatically generated**

* Extraction in JavaScript

*//Addition*

var add = 10+10;

*//substraction*

var sub = 20-10;

*//multiplication*

var mul = 10\*8;

*//division*

var div = 66/33;

* Increment a number in JavaScript

*increment* or add one to a variable with the ++ operator.

i++;

is the equivalent of

i = i + 1;

* Decrement a number in JavaScript

*decrement* or decrease a variable by one with the -- operator.

i--;

is the equivalent of

i = i - 1;

* Finding a reminder in JavaScript

The *remainder* operator % gives the remainder of the division of two numbers. Example:

*//remainder*

var remainder;

remainder = 11%3;

* Compound assignment:
  + Addition
  + Subtraction
  + Multiplication
  + Division

*//Compound assigment addition*

var comAdd = 1

comAdd += 5;

console.log(comAdd) *// result will be = 6*

*//Compound assignment with Substraction*

var comSub = 11;

comSub -= 8;

console.log(comSub);

*//Compound assignment with Multiplication*

var comMul = 5;

comMul \*=5;

console.log(comMul);

*//Compound assignment with division*

var comDiv = 20;

comDiv /= 10;

console.log(comDiv);

* Declare String variables

*//Declare String variables*

var myFirstName = "Matt";

var myLastName = "Matt";

* Escaping Literal Quotes in Strings
  + Using backslash \

*//Escaping quotes*

var EscQuotes = "I am a \"double quoted\" string inside \"double quotes\".";

* + Output:

"I am a "double quoted" string inside "double quotes"."

* Quoting a string with single quotes

*//quoting string with single quotes*

var quoStr ='<a href="http://www.example.com" target="\_blank">Link</a>';

console.log(quoStr);

* + Output:

<a href="http://www.example.com" target="\_blank">Link</a>

Graphical user interface, table

Description automatically generated with medium confidence

* Example code:

var escSeq = "FirstLine\n\t\\SecondLine\nThirdLine";

console.log(escSeq);

* Concatenating Strings with operator. Example code:

*//String plus string*

var strPlus = "This is the start. "+ "This is the end.";

* Constructing Strings with variables. Example code:

*//Constructing with variables*

var myName = "Matt";

var myStr = "Hello " +myName+ ", how are you?";

* Appending variables to Strings:

*//Appending variables to Strings*

var someAdjective = "Hey";

var myAdd = "Learning to code is ";

myAdd += someAdjective;

* Find a Length of a String
  + Use the .length property to count the number of characters
* Use Bracket notation to find the first character in a String

*//Use a Bracket notation to find the Character in a String*

var firstLetterOfLastName = "";

var lastName = "Lovelace";

*// to get the L letter set 0 for the first letter inside the string*

firstLetterOfLastName = lastName[0]

* String Immutability

*//String Immutability*

*// Setup*

var StrImm = "Jello World";

*//Changing to hello world*

StrImm = "Hello World";

StrImm[0]; *// Apply H*

# 2021 Progress

## **1.Portofolio project (2021/04/19) (Finished)**

* Navigation bar
* Live server of visual studio
* Using bootstrap for responsive design
* Suggestion:
  + Understand of using bootstrap & responsive design device