HTML

<html>

all code goes in here

</html>

<head> contains the Metadata which is the info about the page that isn’t displayed. This is the information about the code itself

Anything between the <html> will be interpreted as html code

<!DOCTYPE html> - this must be your first line of code when using html

There are 6 different heading <h1> </h1> to <h6> </h6> - the h1 being the lasrges

<div> </div> is the division or a section in an html document.

Attributes are made up of 2 parts – The name and value of attribute

One commonly used attribute is the id. We can use the id attribute to specify different content (such as <div>s) <div id="introduction">

* *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.
* The <em> tag will generally render as *italic* emphasis.
* The <strong> will generally render as **bold** emphasis.
* The spacing between code in an HTML file doesn't affect the positioning of elements in the browser. If you are interested in modifying the spacing in the browser, you can use HTML's *line break* element: <br>.
* <ul> - is an unordered list –
  + <ul> <li> </li> </ul> (the li add bullet points
* <ol> are like unordered except that they are numbered
  + <ol> <li></li> </ol>

Image = <img src="URL" /> This will add an image

<img src="#" alt="A field of yellow sunflowers" /> - the alt= feature is used for alternate text, it brings meaning to the images on our site

<video></video> is used to add video, must have opening/closing tag

<video src="myVideo.mp4" width="320" height="240" controls> Video not supported </video>

After the src attribute, the width and height attributes are used to set the size of the video displayed in the browser. The controls attribute instructs the browser to include basic video controls: pause, play and skip. The text “Video is not supported” will only be displayed if browser can’t load video.

To Add a link

<a

href=<https://www.wikipedia.org/>> This is a link to Wiki </a>

To add link that will pop up in another browser

<a

href=<https://www.wikipedia.org/>> This is a link to Wiki </a>

To add a link for contact information (the ./ tells the browser to look for the file in the current folder

<a href= “./index.html:> Contact <a/>

To add an image that acts like a link – this will open up a new browser

<a href="https://en.wikipedia.org/wiki/Opuntia" target="\_blank"><img src="#" alt="A red prickly pear fruit"/></a>

<ul>

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

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

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

</ul>

<table></table> creates a table

each <tr></tr> creates a row

<td></td> adds table data

This would show one row with two columns of data

<table>

<tr>

<td>73</td>

<td>81</td>

<tr>

</table>

<table>

<tr>

<th></th>

<th scope="col">Saturday</th>

<th scope="col">Sunday</th>

</tr>

<tr>

<th scope="row">Temperature</th>

<td>73</td>

<td>81</td>

</tr>

</table>

The scope has two values

row – this value makes it clear that the heading is for a row

col – this value makes if clear the the heading is for a column

The colspan denotes the number of columns data can span acrss(out of town Monday-Wednesday)

<tr>

<td colspan=”2”>Out of Town</>td>

<td>Back in Town</td>

<tr>

rowspan extends to multiple rows

<tr>

<th>Morning</th>

<td rowspan =”2”>Work</td>

<td rowspan=”3>Relax</td>

</tr>

The<thead></head> is used to section off the tables heading

The <tbody></tbody> sections off the table body element

The <tfoot></tfoot> is used to create a for totals for example

<tfoot>

<td>Total</td>

<td>28</td>

</tfoot>

Forms are responsible for collecting information and sending it somewhere else

<form action=”/example.html” method=”POST”>

</form>

In the above example the action attribute determines where the information is sent and the method attribute is assigned a HTTP verb that is included in the HTTP request

To add text:

<form action=”/example.html” method=”POST”>

<input type=”text” name=”first-text-field”>

</form>

say you type in ‘important details’ in the text bar

When the form is submitted, the text: “first-text-field=”important details” is sent to /example.html

The label HTML tag is used to associate a text label with a form

<label for=”meal”> What do you want to eat?</label>

<input type=”text” name=”food” id=”meal”>

The number makes it so that you can only input numbers(side note: if you did type=”password” this would cover what your typing by using astericks or dots

<label for=”years”> Years of experience: </label>

<input id=”years name=”years” type=”number” step=”1”</form>

This is how you input a range – and the smoothness on how you move around within that range

<label for=”volume”>Volume Control</label>

<input id=”volume” name=”volume” type=”range” min=”0” max=”100” step=”1”>

Checkbox input

<input id=”cheese” name=”topping” type=”checkbox” value=”cheese”>

#we are putting input first so that text is on the right

<label for=”cheese> Cheese</label>

radio – when we present multiple options but only want a user to pick one

<input type=”radio” id=”two” name=”answer” value=”2”>

<label for=”two”> 2 </label

Drop down list – if you have a whole list of options

Select creates a drop-down list

<label for=”lunch> What’s for lunch?</label>

<select id=”lunch” name=”lunch>

<option

value=”pizza”>Pizza</option>

<option

value=”curry”>Curry</option>

<form>

<label for=”city”>Ideal city to visit?</label>

<input type=”text” name=”city” list=”cities” id=”city”>

<select

While <select> and <datalist> share some similarities, there are some major differences. In the associated <input>element, users can type in the input field to search for a particular option.

<form>

<label for=”city”>Ideal city to visit?</label>

<input type=”text” name=”city” list=”cities” id=”city”>

<datalist=”cities”>

<option value=”New York City”> </option>

<option value =”Tokyo”></option

<textarea> is used to create a bigger text box for blogs and posts

<label for=”blog”> New Blog Post:</label>

<textarea id=”blog” name=”blog” rows=”5 cols=”30”></textarea>

this creates a New Blog Post: title with 5 rows by 30 columns

To submit a form

<form>

<input type=”submit” value=”send”>

</form>

The required pattern states: that we the payment must be all numbers that are 14-16 characters long.

<input id=”payment” name=”payment” type=”text” required pattern=”[0-9]{14, 16}>

JAVASCRIPT

Adding quotes = the \ tells Javascript to ignore the character’s special meaning and just use the literal value of the character. Ex. “the man whispered \”please speak to me.\””

Javascript implicit coerces integer into string or vice versa “hello” + 1 => “hello1” “1” == true => true because it changes 1 to true

Strict equality- === “1” === 1 is false but “1” == 1 is true

var price = 15.00

var money = 20.00

if(money>= price) {

console.log("buy the hammer")}

else {

console.log("dont buy the hammer")

}

//the curly braces is the code that is executed given the condition

var weather = "sunny";

if (weather ==="snow") {

console.log("bring a coat")

} else if (weather ==="rain"){

console.log("bring a rain jacket")

} else {

console.log("wear what you have on")

}

Falsy values: “”, null, undefined, 0, NaN

Truthy values Examples: true, 42, “pizza”, “0”, “null”, “undefined”, {},[]

To use the ternary operator, first provide a conditional statement on the left-side of the ?. Then, between the ? and : write the code that would run if the condition is true and on the right-hand side of the : write the code that would run if the condition is false

var isGoing = true;

var color = isGoing ? "green": "red";

console.log(color)

A switch statement is another way to chain multiple else if statements that are based on the same value without using conditional statements. Instead, you just switch which piece of code is executed based on a value.

The else if statement (option ===[value]) has been replaced with a case clause (case:[value]) and those clauses have been wrapped inside the switch statement

Var option = 3

Switch (option) {

Case 1:

Console.log(“you selected option 1.”);

Break;

Case 2:

Var tier = “none”;

Var output = “youll receive”;

Switch(tier){

…

default:

output += “one copy”

}

default is whats automatically added if nothing else matches the case

WHILE LOOPS

var start = 0; // when to start

while (start < 10) { // when to stop

console.log(start);

start = start + 2; // how to get to the next item

}

FOR LOOPS

for (var i = 0; i < 6; i += 1){

console.log(“Printing out i = “ + i);

You can even set a function as a variable

Var catSays = function(max){

put fun setTimeout(function() {}, 5000)ction here

}

you can just put function since you already named it, otherwise itd be redundant

HOW TO DEFINE AN OBJECT

Create a variable and assign it to {}

Something an object can do is a method

// function expression catSays

var catSays = function(max) {

var catMessage = "";

for (var i = 0; i < max; i++) {

catMessage += "meow ";

}

return catMessage;

};

// function declaration helloCat accepting a callback

function helloCat() {

return "Hello " + catSays(3);

}

// pass in catSays as a callback function

console.log(helloCat(catSays));

SPLICE

Arr = [1,2, 3, 4]

Arr.splice(1, 2, “Hello”, “there”)

This starts at index[1] and deletes two elements, and add “hello” and “there”

* [1, “Hello”, “There”, 4]

ENNUMERABLE forEach

var donuts = ["jelly donut", "chocolate donut", "glazed donut"];

function printDonuts(donut){

donut += " hole";

donut = donut.toUpperCase();

console.log(donut);

}

donuts.forEach(printDonuts)

but could express as an inline function like:

donuts.forEach(function(donut){

donut += " hole"

donut = donut.toUpperCase();

console.log(donut)

})

forEach(element, index, array)

MAP alters the original array and returns a new one

var donuts = ["jelly donut", "chocolate donut", "glazed donut"];

var newer = donuts.map(function(donut, i, arr) {

donut += " hole";

donut = donut.toUpperCase()

return donut

});

console.log(newer)