HTML

* Hyper Text Mark-up Language
* **NOT** a programming language
* for creating webpages / documents
* building blocks of the web
* does not need a server
* file must end with .html extension
* runs in a web browser
* *index.html* is the root or home page of a website

Tag Syntax

* Element names surrounded by angle brackets
* normally comes in pairs
* end tag is usually the same but with a forward slash
* self-closing tags ( <br> )

HTML Skeleton

<!DOCTYPE html>

<html>

<head>

<title>Page Title</title>

</head>

<body>

<h1>This is a heading</h1>

<p>This is a paragraph</p>

</body>

<!-- comment -->

</html>

Inline Elements

* do not start on a new line
* take only necessary width
* <span>, <img>, <a>

Block elements

* start on a new line
* take full width available
* <div>, <h1> to <h6>, <p>, <form>

Attributes

* all tags can have attribute
* provide information about an element
* placed within the start tag
* key/value pairs (id = "some id")

Semantic tags - a semantic element clearly describes its meaning to both the browser and the developer

<header></header>

<footer></footer>

<aside></aside>

<main></main>

<article></article>

<nav></nav>

<section></section>

<details></details>

CSS

* cascading style sheets
* **NOT** a programming language
* styling language
* used for website layout and design

Methods for adding CSS

* inline CSS - directly in the html element
* internal CSS - using <style> tags within a single document
* external CSS - linking an external .css file

**a { background-color: yellow; }**

selector

declaration

property

property/value separator

value

declaration separator

Colors in CSS

* color names
* HTML5 Color names
* Hexadecimal
* RGB

body{

color: red;

background: coral;

}

h1{

color: #00ff00;

}

p {

color: rgb(0,0,255);

}

Margin is the space outside the content

Padding is the space inside the content

p{

margin-top: 5px;

margin-bottom: 5px;

margin-right: 10px;

margin-left: 10px;

}

p{

margin: 5px 10px 5px 10px;

}

p{

margin: 5px 10px;

}

Positioning in CSS

* Static
* Relative
* Absolute
* Fixed
* Initial
* Inherit

Javascript

* High level, interpreted programming language
* Conforms to the ECMAScript specification
* Multi-paradigm
* Runs on the client/browser as well as on the server (Node.js)
* it is the programming language of the browser
* build very interactive user interfaces with frameworks like react
* used in building very fast server side and full stack applications
* used in mobile development (React Native, NativeScript, Ionic)
* Used in desktop application development (Electron JS)

Variables - var, let and const

Data Types - Strings, Numbers, Boolean, null, undefined

console.log(`My name is ${name} and I am ${age}.`)

Arrays - variables that hold multiple values

.push() - adds value at the end of the array

.unshift() - adds value at the beginning of the array

.pop() - removes the last value of the array

Loops and Array Methods

//for

for(let i = 0; i < 10; i++){

console.log(`For loop item ${i}`);

}

//while

let i = 0

while(i < 10){

console.log(`While loop item ${i}`);

i++;

}

//for...of

for(let toDo of toDos){

console.log(toDo.text);

}

//for each

todos.forEach(function(todo){

console.log(todo.text)

});

//map

const todoText = todos.map(function(todo){

return todo.text;

});

console.log(todoText);

//filter

const todoCompleted = todos.filter(function(todo){

return todo.isCompleted === true;

});

console.log(todoCompleted);

Conditional Statements

//if

const x = 13;

if(x == 10){

console.log('x is 10');

} else if(x > 10){

console.log('x is greater than 10');

} else{

console.log('x is less than 10');

}

|| - OR statement

&& - AND statement

// (?) ternary - question mark, value if true followed by value if false

const x = 11;

const color = x > 10 ? 'red' : 'blue';

console.log(color);

//switch case

const color = 'green';

switch(color){

case 'red':

console.log('color is red');

break;

case 'blue':

console.log('color is blue');

break;

default:

console.log('color is neither red nor blue');

break;

}

Functions

// normal function

function addNum(num1, num2){

return num1 + num2;

}

console.log(addNum(5,3));

// arrow function ( => ) fat arrow

const addNum = (x,y) => { return x + y };

console.log(addNum(6,5));

Object Oriented Programming

// Constructor Function and Methods

function Person(fName, lName, dob){

this.fName = fName;

this.lName = lName;

this.dob = new Date(dob);

this.getFullName = function(){

return `${this.fName} ${this.lName}`;

}

}

// Prototype

Person.prototype.getBirthYear = function(){

return this.dob.getFullYear();

}

// Instantiate object

const person1 = new Person('John', 'Doe', '04/03/1980');

const person2 = new Person('Mary', 'Smith', '03/25/1990');

const person3 = new Person('Stan', 'Lee', '12/22/1922');

console.log(person1.getFullName());

console.log(person2.getBirthYear());

console.log(person3);

Class - Doing the same thing as function, just a different way of writing

class Person{

constructor(fName, lName, dob){

this.fName = fName;

this.lName = lName;

this.dob = new Date(dob);

}

getBirthYear(){

return this.dob.getFullYear();

}

getFullName(){

return `${this.fName} ${this.lName}`;

}

}

// Instantiate object

const person1 = new Person('John', 'Doe', '04/03/1980');

const person2 = new Person('Mary', 'Smith', '03/25/1990');

const person3 = new Person('Stan', 'Lee', '12/22/1922');

console.log(person1.getFullName());

console.log(person2.getBirthYear());

console.log(person3);

DOM - Document Object Model

Single element selection

console.log(document.getElementById('my-form')); //class name

console.log(document.querySelector('h1')); //tag name

Multiple element selection

console.log(document.querySelectorAll('.item'));

Manipulation

// example 3 list items

const ul = document.querySelector('.items');

ul.firstElementChild.textContent = 'hello';

ul.children[1].innerText = 'world';

ul.lastElementChild.innerHTML = '<h1>Heading</h1>';

What is GIT?

* Version Control System (VCS) for tracking changes in computer files
* distributed version control
* coordinates work between multiple developers
* who made what changes and when
* revert back at any time
* local and remote repository

Concepts of GIT

* keeps track of code history
* takes "snapshots" of your files
* you decide when to take a snapshot by making a "commit"
* you can visit any snapshots at any time
* you can stage files before committing

Basic Commands

$ git init // initialize local git repository

$ git add <file> // add file(s) to index

$ git status // check status of working tree

$ git commit // commit changes in index

$ git push // push to remote repository

$ git pull // pull latest from remote repository

$ git clone // clone repository into a new directory

Installing Git

http://git-scm.com/download

Branch - The default branch name in Git is master.

Merge - The command lets you take the independent lines of development created by git branch and integrate them into a single branch.