Javascript

Web Tech Overview

* HTML is the car frame
* CSS is the body and the paint, tells you the details
* Javascript is the functionality of the car or the animation/ interactivity

All Javascript names

* Mocha
* LiveScript
* Jscript
* ECMAScript

Javascript is the world’s most misunderstood programming language

* Often derided as being a toy
* Deceptively simple to start
* Wickedly powerful once mastered
* “Script” does not mean lesser language
* Script means interpreted language means compiled

History of Javascript

* Creates in 1995 by Brendan Eich at Netscape
* Javascript was first released to the public in version to of netscape
* First called livescript renamed because java was popular
* Java and javascript is completely different
* We write the two codes very differently
* A few months after the release of javascript, microsoft released the equivalent with internet explorer 3 called JScript
* Because microsoft had the same product netscape sent the language to ECMA International, a European Standards Organization, Fist standardization was born
* ECMA script is the official name of Javascript

Getting Started

* Displays the requested info in the console. Useful for debugging code.
* A large part of programming is experimentation
* Javascript does not know the difference between a single quote and an apostrophe
* To be more explicit add a backslash

Data Types

* Define: data types are bits either 1 or 0, bits mean basically either on or off, data types are for different bits of information, also develop expectations on how data can be used
* Typeof operator can explore javascripts data types
* A string is any collection of numbers
* When you put a quotations around one of the numbers you are trying to add javascript tries to make them both compatible
* Javascript sees anything in quotation marks as a string
* We have to provide context for javascript according to the rules
* Strings can be closed in single or double quotation marks
* Javascript also has the following primitive types
  + Boolean= True or False
  + Undefined= no value or type
  + Null= Empty Value

Javascript uses number like how we read them

* String are collections of characters
* A character is anything you can type on your keyboard
* We have to escape quotation marks when they are inside of a string
* When we add strings together this is called concatenation
* Template literal behave like strings instead they aren’t wrapped in a single or double quote, they are wrapped in backticks
* Inserting strings into other strings is called interpolation
* Backsign and curly braces are used are not a common thing to be put out
* Javascript is an interpreted vs a compiled language
* Think of string interpolation as a sort of order of operations for javascript
* Parameters are placeholders that we put between the parentheses when declaring a function
* Arguments are only available within the function
* Most parameters are separated from a coma
* What matters with functions is not what the argument is called, but its location in the arguments list.
* Our functions are only logging the information in their local scope
* When we return inside a function, we’re giving that value back to the world outside the function
* Byt wrapping a return in quotes it makes the console know we are returning a string, instead of just logging values
* If we don’t return anything from a function, the function returns the default value,undefined,

**JS Functions**

* Console.log is a great way to test out javascript
* Functions are ways of giving javascript instructions that is can run over and over again.
* A function is almost like a container for javascript
* Basic syntax of a function is parentheses or curly braces
* We start with the function keyword, then the name, followed by parentheses. After we have a pair of curly braces
* We can use an argument to pass information to a function
* We can pass arguments to a function between its parentheses

Git- Version management software

* Removes the need to copy files to and from the class share and your “H” drive
* Like using your camera to take a snapshot of you files at a specific point in time that you can magically go back to if terrible things happen
* GIT is a checkpoint for your files
* Exists so you can
  + Modify
  + Change
  + Break
  + improve...

Your code

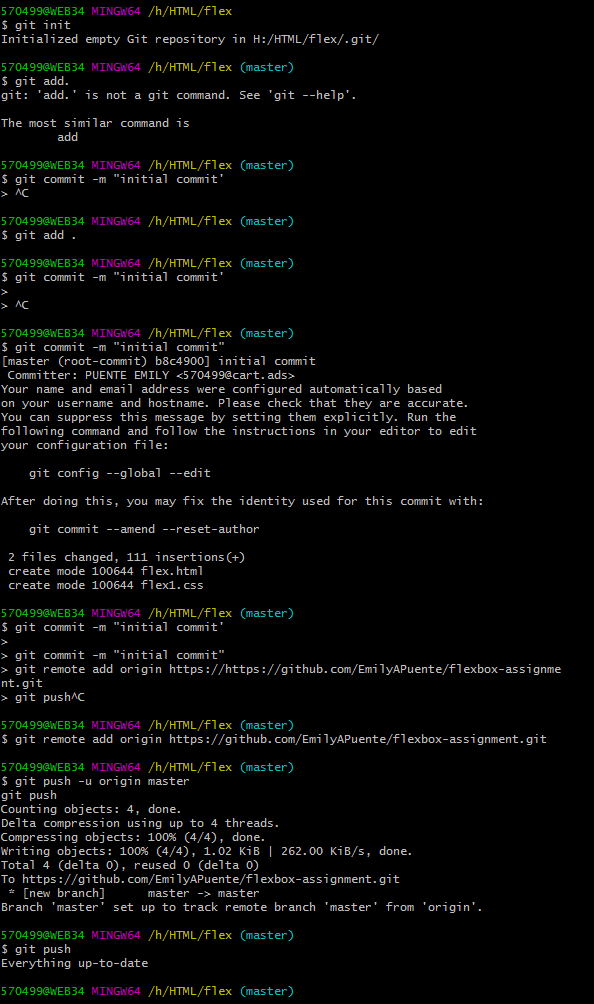
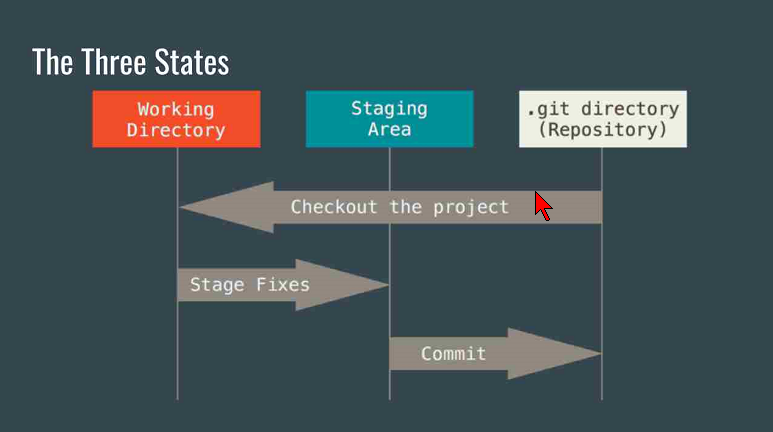
* Secure in the knowledge that you can not ruin your work too badly because you created save points along the way
* Collaboration tool that allows people to work on all the parts of a project at the same time
* Protects yourself and others from yourself and others

THE LOCAL WORKFLOW

* Getting started:
  + Open file explorer
  + Create folder named practice in “h”drive
  + Type cmd in address bar
  + Command prompt should open H:\practice>
  + Tell git to watch git init
* Git init create a repository in the folder you ran the command on
* Often shortened to repo, this is a hidden location where file checkpoints will be stored
* **DO NOT DELETE THE FOLDER**

The three main states

* Modified
  + Filed that are new or have changes not yet saves by git
* Staged
  + The current version of a file, tagged to be included in the next commit
* Committed
  + Files that are stored by git



Add an h2 that contains what you consider the big idea

Write a complete p that summarizes todays lesson

On a scale of 1 to 4 how would you rate your understanding

Add commit log your changes

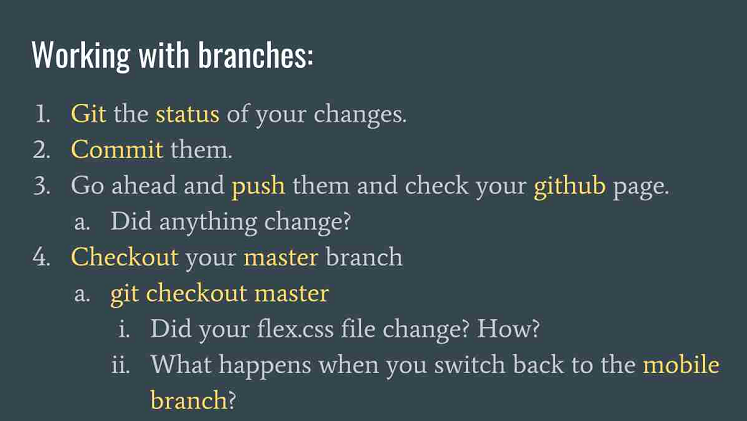
Turn in your HTML file and screenshot

Working With Git

* Remote depository
  + Copy of project that is stored “in the cloud”
  + Where we backup our work and share it with others
  + Accessible anywhere there is an internet connection
* Git push tells git to upload all your changes to the server
* It DOES NOT need to be done after every commit
* Branches
  + Smaller bits from a tree trunk
  + Represent different versions of our code
  + Allow us to work on code fixes and features without breaking what we already have working
  + Fixes and new features should always start on a branch
* The master branch is the “trunk” of your code tree
* Master branch should only contain clean code ready for deployment
* Git branch <name> tells git to maintain a new copy of our code with the given name
* Git branch on its own will list the branches available and display an asterisk next to the one we are currently working on
* Git checkout <branch> tells git to switch our working folder to the branch name specified

Working with Branches

* Git branch mobile
* Switch to our new branch
* Git checkout mobile
* To make sure your on the right branch type git branch



Merge Conflict

* When a file has changed in both branches you are trying to combine
* Git can’t automatically determine what you want to keep
* Basically git is asking for help

Summary:

Today I started to understand more. And it’s not as hard as I thought it was going to be. On a scale of 1-4 I think I would be a 3 because I know what to do but I am going to have to look at my notes a couple of times. It will take practice for me to get it down. I think the branching was what made me understand the best.