

Front End Software Development

Introduction to JavaScript (weeks 1 - 6)

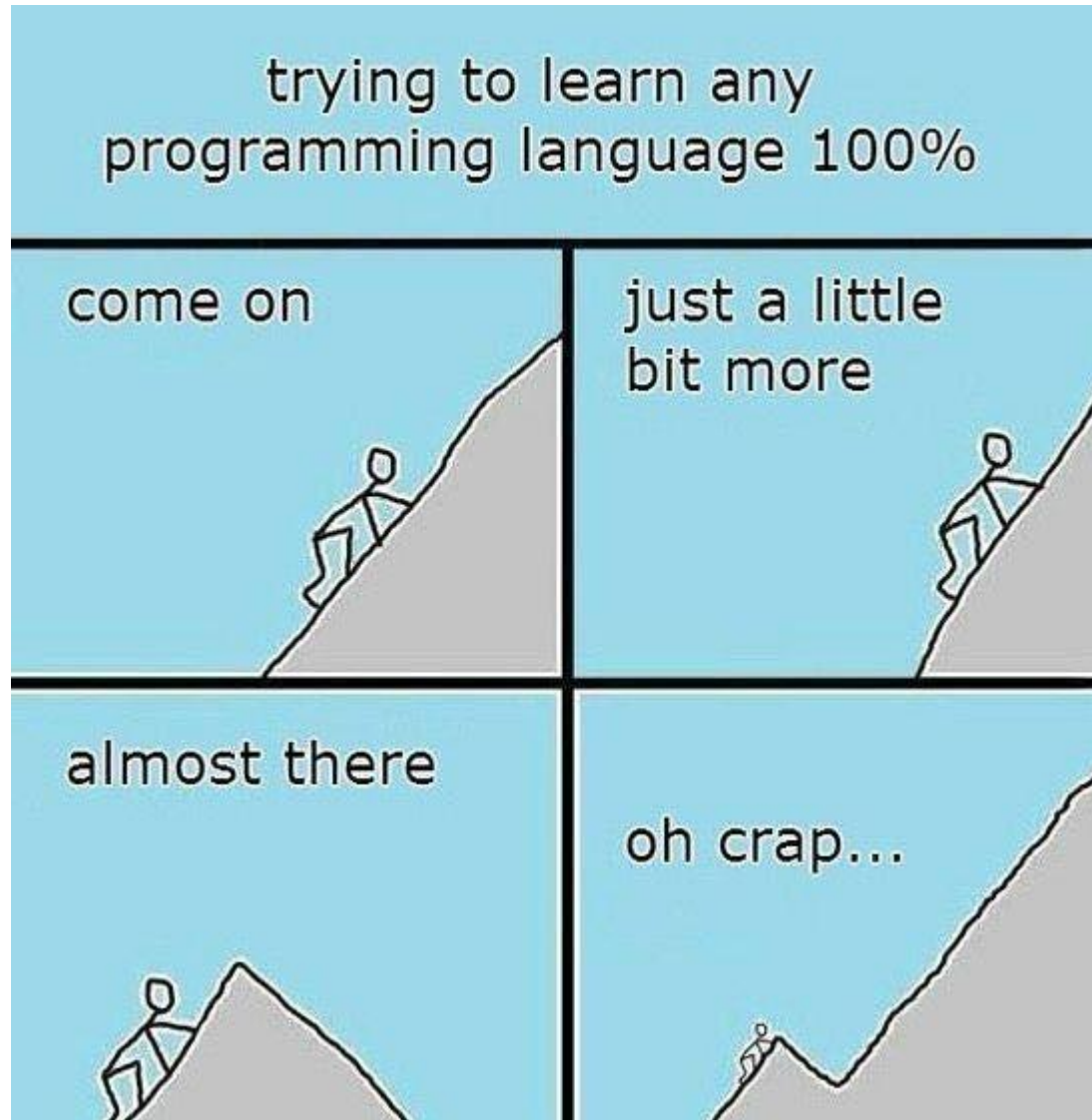
Week 01



Agenda

- Questions
- What is Programming?
- Command Line / Shell
 - Common commands / OS Comparison
- Source Control / GIT
- Variables & Data Types
- Operations

Questions



What is?

(Programming)



- Types of Programming Languages
 - Compiled (machine / CPU level code)
 - Assembly, C, C++, Pascal, Rust, Go
 - Scripted / Interpreted
 - **JavaScript**, PHP, Ruby, Python, Java, C#
- Classification
 - Procedural
 - Object-Oriented
 - Functional



Command Line / Shell



Windows

- Command Prompt
- PowerShell



MacOS, Linux, BSD, UNIX, etc.

- BASH (**B**ourne **A**gain **S**hell)
- Zsh (**Z** shell)

Home / Users Folder (*Directory*)

C:\Users\yourusername

/Users/yourusername
/home/yourusername

Take note of the difference in the path
separators between Operating Systems: \ vs. /

Command Line / Shell

(Common commands / OS comparison)

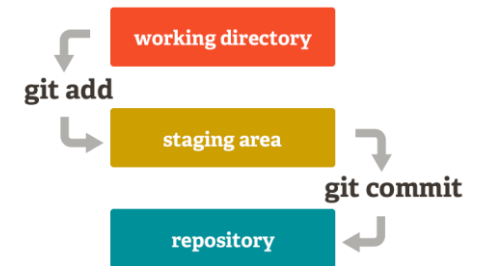


Windows	MacOS / Linux	Description
dir dir foldername	ls ls foldername	Displays list of folders and files.
cd foldername cd ..	cd foldername cd ..	Changes the current folder. Note: .. is the directory above
cd	pwd	Displays the name of the correct directory.
mkdir foldername	mkdir foldername	Creates a folder
rmdir foldername	rm foldername	Deletes a folder
copy from to	cp from to	Copies a file.
del filename	rm filename	Deletes a file.
move from to	mv from to	Moves and/or renames a file.
cls	clear	Clears the screen
type filename	cat filename	Displays the contents of the file.
help command	man command	Displays help / usage for a command.

Source Control / GIT



- Distributed vs Centralized
 - **GIT**, SVN, CVS, Mercurial, Subversion, TFS, Bazaar, ...
- Typical Workflow
 - Modify Files > Add > Commit [> **Push**]
push makes commits available for other developers.
 - Local vs remote repository
commit != push
- Tips
 - Focused Commits
 - Commit Early / Commit Often (*First In Wins*)
 - Branching Strategies – Bug Fix, New Features





DEMO

Source Control / GIT

Variables & Data Types

- Cannot start with number or punctuation except _
- No spaces (use _)
- Use camelCase

var
let
const

variableName = *value* ;
variableName = *value* ;
variableName = *value* ;

Type	Examples
boolean	true, false
number	-10, 1.12, 63, 99
string	"Tom", 'Alvin'
symbol	Symbol("key")
object	{ age: 5 }
array	[1, 2, 4]
null	
undefined	

var

- Scope – Hoisted to top of function block
Optional (**warning: global variable*)

let

- Scope – Hoisted to top of code { } block
(most programming languages)

const

- read only, once set cannot be changed

```
const salesTax = 0.055;  
var isRaining = false;  
var name = 'Sam';  
let state = "AZ";  
let age = 45;  
let index = 0;  
temperature = 212;
```



DEMO

Declaring Variables / Naming Conventions

Operations

- Arithmetic Operators

Operator	Description	Examples
/	Division	$4/2 = 2$, $5/2 = 2.5$
%	Modulus (Remainder)	$4/3 = 1$, $15/7 = 1$
*	Multiplication	$5*2 = 10$, $1*0=0$
+	Addition	$3+9=12$, $7+2=9$
-	Subtraction	$4-4=0$, $9.3-1.2=8.1$

A hand-drawn long division problem: 7 divided into 15. The quotient 2 is written above the 5. 7 times 2 is 14, which is subtracted from 15 to leave a remainder of 1. The numbers are color-coded: 7 is green, 15 is blue, 14 is red, and 1 is pink. An orange arrow points from the remainder 1 to the code block below.

- Equality Operators

Operator	Description
==, ===	Is equal, (same type)
!=, !==	Is not equal, (same type)
<, <=	Less than, less than or equal
>, >=	Greater than, greater than or equal

```
let a = 5;
let b = 10;
let c = a + 5; // 15
let b = c * a; // 65
let a = 1+2*2; // 5
let a = (5+5)/2; // 5
(b == b) // true
(b == c) // false
(a != c) // true
(b > c) // true
(a > c) // false
"b" == 'b' // true
"c" == "C" // false
```



DEMO

Operations: Mathematics & Equality