



COMMAND LINE, DATA TYPES, & GIT

UNIX COMMANDS

<code>ls</code>	List the contents of the directory	<code>pwd</code>	Print working directory
<code>cd</code>	Change directories	<code>say</code>	Make your computer talk
<code>mkdir</code>	Make directory	<code>open</code>	Open a particular file in its default application
<code>rmdir</code>	Remove <i>empty</i> directory	<code>mv</code>	Move file or rename
<code>rm</code>	Remove files or directories	<code>cat</code>	Show the contents of a file
<code>touch</code>	Create an empty file	<code>code .</code>	Open contents of directory in Visual Studio Code
<code>echo</code>	Return a string	<code>subl .</code>	Open contents of directory in Sublime Text
<code>clear</code>	Clear the terminal	<code>atom .</code>	Open contents of directory in Atom

DIRECTORY NAVIGATION PATHS

<code>.</code>	The current folder
<code>..</code>	One folder above your current working directory
<code>~</code>	The home folder

DIRECTORY NAVIGATION COMMANDS

<code>cd myFolder</code>	go into myFolder
<code>cd ..</code>	go up one folder/directory
<code>cd ~</code>	go to home folder/directory
<code>cd</code>	shorthand for <code>cd ~</code>

GIT COMMANDS

<code>git init</code>	Create a new repository
<code>git status</code>	See the status of files in your repo
<code>git log</code>	Look at a list of commits
<code>git add filename</code>	Add file to the stage for committing
<code>git add .</code>	Shortcut command to add all changed files within the current directory and its descendants to the stage
<code>git commit -m "description of changes"</code>	Create a snapshot of your project. Requires a message (like "added comments throughout")
<code>git push origin master</code>	Push updates to GitHub, where <code>origin</code> is a common alias for the remote url and <code>master</code> is the default local branch
<code>git pull</code>	Pull updates from GitHub
<code>git remote -v</code>	View the remote Github push and pull URLs for the current directory

VARIABLES

Declaring a variable: `let age;`
Assigning a variable: `age = 29;`
Both in one step: `let age = 29;`

EQUAL SIGNS

`=` *assigns value on right to object on left*
`===` *evaluates whether values on left and right are the same*

DATA TYPES

String	Literal characters, enclosed in quotes	<code>"about"</code>
Number	Numbers treated as numeric values (not in quotes)	<code>15</code>
Array	Collection of data	<code>["Larry", "Curly", 15, 42]</code>

ARITHMETIC OPERATORS

<code>+</code>	<i>add (also concatenates strings)</i>
<code>-</code>	<i>subtract</i>
<code>*</code>	<i>multiply</i>
<code>/</code>	<i>divide</i>
<code>%</code>	<i>modulus (remainder)</i>

SPECIAL NUMBER OPERATORS

<code>Math.pow(m, n)</code>	<i>Returns m to the power of n</i>
<code>Math.sqrt(n)</code>	<i>Returns the square root of n</i>
<code>Math.random()</code>	<i>Returns a random number between 0 (inclusive) and 1 (exclusive)</i>
<code>Math.floor(n)</code>	<i>Returns largest integer less than or equal to n</i>
<code>Math.ceil(n)</code>	<i>Returns smallest integer greater than or equal to n</i>

COMPOUND OPERATORS

<code>+=</code>	<i>adds a number to a variable and assigns the new value to the same variable</i>
<code>-=</code>	<i>subtracts a number from a variable and assigns the new value to the same variable</i>
<code>++</code>	<i>adds 1 to a value</i>
<code>--</code>	<i>subtracts 1 from a value</i>
