DATA TYPES & WORKING WITH ARRAYS

SOME QUESTIONS FROM LAST TIME

- How does Javascript interact with other languages?
- How does server-side code interact with client-side code?
- What's the benefit of using the terminal instead of a GUI?
- How do I find a local git repo if I forgot where I cloned it?
- Can you host a site through GitHub?

LEARNING OBJECTIVES

- Have a strong grasp on Javascript variables and how to use them
- Understand what data types are, what types exist in Javascript, and the differences between them
- Know what arrays are and how to declare them
- Be able to manipulate arrays and access the data in them
- Know the basics of iterating through arrays

DATA TYPES

DATA TYPES IN JAVASCRIPT

- Strings - a collection of characters
- Numbers - any real number, integer or decimal
- Booleans - a binary value (true or false)
- Arrays - a list of data
- Objects - a map of data
- Null - the value signifying no value
- Undefined - the absence of a value

JAVASCRPIT VARIABLES

A variable in Javascript is a name that you can assign values to. Variables can hold any valid Javascript value, and they can change value at any time. You can use a variable in any place that you could use a value.

JAVASCRIPT KEYWORDS

A keyword in Javascript is a reserved word in the language. It was special functionality in code, and therefore cannot be used as the name of a variable.

break	export	super
case	extends	switch
class	finally	this
catch	for	throw
const	function	try
continue	if	typeof
debugger	import	var
default	in	void
delete	instanceof	while
do	new	with
else	return	yield

INTRO TO ARRAYS

WHAT IS AN ARRAY?

An array is a list of data. Arrays are ordered and have a definite size. Arrays can contain any valid Javascript values, including other arrays.

WORKING WITH ARRAYS

- Create an array with square brackets

```
var array = [1, 2, 'three']
```

- Access elements using square brackets and an index

```
console.log( array[0] )
```

- Get the size of an array using the length property

```
console.log( array.length )
```

10 MINUTE BREAK

ARRAY METHODS

- join(separator) - create a string containing all of the items in the array separated by the separator
- push(item) - adds a new item to the end of the array
- pop() - removes the last item of the array
- shift() - removes the first item of the array
- unshift(item) - adds a new item to the start of the array
- reverse() - reverses the order of the elements in the array

LOOPS

WHAT IS A LOOP?

A loop repeatedly executes a section of code until a condition becomes false, at which point the code stops executing.

SOME EXAMPLES (IN ENGLISH)

- Run this code once for every item in this array
- Run this code as long as my variable is equal to true
- Run this code as long as my variable is less than 10, and I will increment my variable every time the code runs

THE FORMAT

```
for (var i = 0; i < array.length; i = i + 1) {
    console.log(array[i]);
}
```

ANOTHER WAY

```
array.forEach(function(element) {
    console.log(element);
});
```

INDEPENDENT PRACTICE

HOMEWORK

BUILD A PAGE THAT:

- Will generate the idea for a random startup based on a pre-determined list of popular startups and a target group of consumers (ie “It’s like Tinder for astronauts!”)
- Will allow the user to add any randomly generated startup idea to a list of favorite ideas
- Displays the list of favorites in some form (can be a button that shows them somehow, or the list can always be on the screen and constantly updating)

QUESTIONS?

EXIT TICKETS

MORE ARRAY METHODS

- map(fn) - transforms an array of elements into an array of elements based on the first array
- filter(fn) - removes elements from an array based on a function that determines whether or not an element should stay
- some(fn) - determines if there is an element in the array that satisfies some condition
- every(fn) - determines if *every* element in an array satisfies some condition