Intro to Front End - Intro to Javascript Part 2

Set up the Live Stream

Front End Fundamentals

How's Javascript going?

Intro to Javascript Part 2

Today we'll do the same as Tuesday's class with a bit of lecture and a bit of coding.

Today we'll talk about:
Arrays and Objects in Javascript
Getting and setting values from both types
Writing a for loop

Review

Last class we talked about:

- Console Logs: a helper function that prints out to the console

```
console.log("Happy Thursday")
```

- Data Types: the values we use in javascript are all of a type and these are:
  - Numbers: we can add, subtract, multiply and divide numbers in JS

```
  console.log(1 + 1)
  console.log(10 - 5)
```

- Strings: often represent names, email addresses in code

```
  console.log("123"); // not a number!
```

- Booleans: values that are true or false

```
  console.log(true);
  console.log(false);
```

Variables: store values for reuse

```
var name = "Lighthouse Labs"

console.log(name)
```

If statements: allow our code to do different things based on conditions

```
var year = 2030;

if (year >= 2019) {
  console.log("HOW DID YOU GET TO THE FUTURE");
}
```

Functions: help us store instructions and actions for reuse;

```
function isOdd(num) {
  return num % 2 !== 0;
}

console.log(isOdd(10));
console.log(isOdd(11));
```

Objects

Objects are a way of organizing related information in key-value pairs.

We can save these sets inside variables.

Objects are best when you want to organize based on data labels. Kind of like a newspaper. When you read a newspaper you don't necessarily read it back to front - you flip to different sections based on a sections name.

Wherever that section in the newspaper is located, you can flip to it immediately and have the right context.

This is different from, say a book, where the order of the chapters and sections matter.

If you need a real world example, check out this analogy: JavaScript Arrays and Objects are like Book and Newspapers

---

Object Syntax

To define an object, we can assign it to a variable.

We wrap the key value pairs in curly braces, define a key, and then give it a value after a colon. To separate key value pairs, use a comma.

Live Code

```
var book = {
  title: "Harry Potter and the Prisoner of Azkaban",
  author: "J.K Rowling",
  releaseDate: "July 8, 1999"
};

console.log(book);
```

Accessing Values

To access the values stored in an object you can use:

- Bracket Notation: call the name of the object then the key name you are accessing. They key name is wrapped in square brackets.

- Dot Notation: call the name of the object, then a dot and the key name you are accessing; These behave the same way but sometimes you'll need to use bracket notation (which we will learn in later classes!)

Live Code

- How do we access the title using bracket notation?
- How do we access the author using dot notation?
- How do we access releaseDate using bracket notation?

Add this:

```
var book = {
  ratings: {
    amazon: "4.6 / 5",
    goodreads: "4.54 / 5"
  }
};
```

- How to access the amazon rating using bracket notation
- How to access the goodreads rating using dot notation

_Get them to go to Compass and do Objects

Arrays

Arrays let your store a list or collection of items.

In JavaScript, an array can be of any length and contain any data type.

Both arrays and objects and be used to store related pieces of data.

Array Syntax

To define an array, we can assign it to a variable.

We list items in an array and separate each item with a comma.

The items can be of any data type - see 4, 5 above

You should use arrays when the order is the most important factor for organizing the info.

Live Code

```
var readingList = [
  "1984",
  "The Great Gatsby",
  "Lord of the Flies",
  "Charlotte’s Web",
  "The Hobbit"
];

var mixedArray = ["string", true, 23, 37, false, "false"];
```

Accessing Values

To access values in an array we use the same bracket notation we use for objects,
But instead of key names we put the values “index” or the position inside the array.

ARRAYS START AT ZERO - so the first position is accessed by using zero.

Live Code

- How would we get The Hobbit?

```
readingList[4];
```

- How would we get Lord of the Flies?

```
readingList[2];
```

- How would we get 37?

Note: The type of value doesn’t matter at all here, just the index

```
mixedArray[3];
```

_Get them to go to Compass and do Arrays

Loops

Loops are a really useful feature of JavaScript.

They let us repeat the same action or piece of code to each item in a set.

Live Code

Can you think of a way to print out each item in the readingList array?

Well we can console.log each one but this hurts my lazy developer brain.

```
console.log(readingList[0]);
console.log(readingList[1]);
console.log(readingList[2]);
console.log(readingList[3]);
console.log(readingList[4]);
```

Loops

For loops are used to loop over arrays:

- for: The keyword;
- (): The loop's expressions will be wrapped in round braces;
- var i = 0: The first expression initializes the loop with a new variable named i for "index". We set it to zero so the loop starts at the first item in the and counts up;
- i < array.length:
  The second expression is the loop's condition. It tells the loop to keep running as long as the " index " is less than the length of the array being looped over;
- .length: Like "console.log" this is another javascript helper function. It will return the length of an array.
- i++: A final expression that is evaluated at the end of each loop. It usually updates the counter variable " i ".
  " i++" increments the counter by +1

Live Code

- .length

```
readingList.length()
mixedArray.length()
```

- For Loop

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

- With an If Statement

```
var numbers = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10];

for (var i = 0; i < numbers.length; i++) {
    if (numbers[i] % 2 === 0) {
        console.log(“even!”)
    } else {
        console.log(numbers[i])
    }
};
```

While Loops

While loops will keep performing their action until the condition provided evaluates to false.

Live Code

```
var count = 1;

while (count < 10) {
  console.log(count);
  count = count + 2;
}
```

** count += 2:

- increments the count variable by 2 each time the loop executes; the equivalent of count = count + 2

_Get them to go to Compass and do Loops

After exercise, put in pairs for pair programming!

Fizz Buzz

```
for (var i = 1; i < 101; i++) {
  if (i % 15 == 0) console.log("FizzBuzz");
  else if (i % 3 == 0) console.log("Fizz");
  else if (i % 5 == 0) console.log("Buzz");
  else console.log(i);
}
```

Yellow Pager

```
function yellowPager(word) {
  let str = word.toLowerCase().split("");
  let answer = "";

  for (var i = 0; i < str.length; i++) {
    if (str[i] == "a" || str[i] == "b" || str[i] == "c") {
      answer += "2";
    } else if (str[i] == "d" || str[i] == "e" || str[i] == "f") {
      answer += "3";
    } else if (str[i] == "g" || str[i] == "h" || str[i] == "i") {
      answer += "4";
    }
  }

  console.log(answer);
}

yellowPager("Lighthouse");
```

Word Count

```javascript
var foods = [
  "pizza",
  "celery",
  "bananas",
  "pizza",
  "lettuce",
  "cookies",
  "pizza"
];

function wordCount(wordArray, searchWord) {
  let count = 0;

  for (var i = 0; i < wordArray.length; i++) {
    if (wordArray[i] === searchWord) {
      count += 1;
    }
  }

  return count;
}

console.log(wordCount(foods, "celery"));
```