Introduction to Programming JavaScript

Part 2

Arrays

Arrays

Codetrain

Arrays store multiple values in a single variable.

```
let students = new Array(“Sam”, “Joe”, “Kofi”)
```

```
let students = [“Sam”, “Joe”, “Kofi”]
```

Accessing Arrays

Codetrain

By default, every item in an array is assigned a unique identifier called the **index**

```
["Sam", "Joe", "Kofi"]
   0       1       2
```

```
let firstStudent = [0]
//firstStudent will be equal to "Sam"
```

Conditionals and Loops

If Else Statement

Codetrain

```
if (condition is true) {
 // execute any code here if condition is true
} else {
 // execute any code here if condition is false
}
```

Else If Statement

Codetrain

You can use the **else if** statement to specify a **new condition** if the first condition is false

```
if (condition is true) {
 // executed any code here if condition is true
} else if (another condition is true){
 /* executed any code here if above condition is
false and this condition is true */
}else {
 /* executed if any of the conditions above is
false */
}
```

The Switch Statement

Codetrain

```
switch (expression) {
 case n1:
    statements
    break;
 case n2:
   statements
   break;
 default:
   statements
}
```

For Loop

Codetrain

Loops can execute a block of code a number of times. They are handy in cases in which you want to run the same code repeatedly, adding a different value each time.

```
for (statement 1; statement 2; statement 3) {
  //code block to be executed
}
```

Statement 1 is executed before the loop (the code block) starts.
Statement 2 defines the condition for running the loop (the code block).
Statement 3 is executed each time after the loop (the code block) has been executed.

While Loops

Codetrain

The while loop repeats through a block of code, as long as a specified condition is true.

```
while (condition) {
 //code block
}
```

The condition can be any conditional statement that returns **true or false**.

Functions

JavaScript Function

Codetrain

A function is a **block of code** designed to perform a particular task.

The main advantage of using functions is **Code reuse.** Define the code once, and use it many times.

Defining a Function

Codetrain

To define a JavaScript function, use the **function** keyword, followed by a name, followed by a set of **parentheses ()**.
The code to be executed by the function is placed inside **curly brackets {}**

```
function name() {
 //code to be executed
}
```

Function names can contain letters, digits, underscores, and dollar signs (same rules as variables).

Calling a Function

Codetrain

To execute the function, you need to **call it**.
To call a function, start with the **name of the function**, then follow it with **parentheses**.

```
function myFunction() {
 console.log("Calling a Function!");
}

myFunction();
//logs "Calling a Function!"
```

Function Parameters

Codetrain

Function parameters are the names listed in the function's definition

```
function myFunction(name) {
 console.log("Hi, " + name);
}

myFunction(“Angela”);
//logs "Hi, Angela"
```

Function Return

Codetrain

A function can have an optional return statement. It is used to return a value from the function.

```
function addTen(num) {
 let ans = num + 10;
 return ans;
}

let answer = addTen(30);
//answer will be equal to 40
```

Pre built Functions

Codetrain

```
//Alert Box
 alert("Do you want to quit?");

//Prompt Box
 let age = prompt("Please enter you age");
 document.write(user);

//confirm box
 let result = confirm("Are you sure?");
//result can be true or false
```

Arrow Functions

Codetrain

```
const addNumbers = (num1, num2)=>{
 let total = num1 + num2
 console.log(total)
}

addNumbers(3, 10)
//this logs 13
```

Now let's practise some JavaScript!

EXERCISE 1: INSTRUCTIONS

1. Create a new file in your text editor and save it as *conditionals.html* (make sure it's saved in the js-practice folder you created)
2. Let's start with the script tag
3. Create an **Array** containing names of your colleagues
4. Write a **for loop** to print out each name in the array you created in step 3
5. Write an if else, else if statement

EXERCISE 2: INSTRUCTIONS

1. Create a new file in your text editor and save it as *conditionals.js* (make sure it's saved in the js-practice folder you created)
2. Link your JS file to the HTML file you created in EXERCISE 1
3. In *conditionals.js* file, create a function that adds 3 numbers and returns the answer.
4. Console.log the answer returned

Thanks!

Codetrain
Mahama Block,
Accra Digital Centre - Accra Central

info@codetraingh.com
www.codetraingh.com