JavaScript

JavaScript

- Alongside HTML and CSS, JavaScript is one of the core technologies of the World Wide Web
- It's a programming language for web design
- Can be used to make web pages more interactive
- Allows you to change the HTML and CSS dynamically
- Most browsers support JavaScript but users can disable it for security

Syntax

- Every line of code **MUST** end in a semicolon;
- The only exceptions are lines that end with a curly brace: { or }

Data Types

- Numbers
  - Integer (whole numbers) - e.g. 5, 7, 192, 86
  - Float (numbers with a decimal point) - e.g. 3.14, 5.7, 1.0, 99.99
- Strings
  - This is text, e.g. "hello", "How are you?", "ABC123"
  - The text is **always** inside quotation marks
- Booleans
  - These are logical values, either True or False

Numerical Operators

Operator	Description	Example
+	add	2 + 3
-	subtract	3 - 1
*	multiply	9 * 2
/	divide	8 / 4

Strings

- A **String** is any text and must be written inside quotation marks
- These are all Strings:
  - "Hello World"
  - "abc"
  - "How are you?"
  - "This is a String"
  - "abc123"
  - "123456"
  - "5" <-- This is not a number

String Concatenation

- Strings can be joined together to form a new String by using the '+' operator
  - "ABC" + "DEF" = "ABCDEF"
  - "Hello" + "World" = "HelloWorld"
  - "Hello " + "World" = "Hello World"
  - "This " + "is " + "a " + "sentence." = "This is a sentence."
  - "A" + "3" = "A3"
  - "A" + 3 = "A3"
  - "123" + "456" = "123456"
  - "2" + "2" = "22"
  - "2" + 2 = "22"

Logical Operators

Operator	Description	Example
==	...is equal to...	5 == 5
!=	...is not equal to...	2 != 3
>	...is greater than...	7 > 3
<	...is less than...	9 < 12
>=	...is greater than or equal to...	7 >= 3
<=	...is less than or equal to...	9 <= 9

Boolean

- A **Boolean expression** evaluates as either **TRUE** or **FALSE**
- These expressions are **true**:
  - 5 > 2
  - 3 <= 5
  - 7 != 8
  - 6 == 6
  - "hello" == "hello"

Boolean

- A **Boolean expression** evaluates as either **TRUE** or **FALSE**
- These expressions are **false**:
  - 4 > 9
  - 7 >= 8
  - 9 != 9
  - "hello" == "helloo"
  - "two" == 2

Variables

- Use a variable to store a value, such as a number or a String or a Boolean expression
- A variable has a **name** which you use to refer to it, and it has an associated **value**
- Think of your age as a variable. The name of this variable is the word "age" and the value is your age as a number.
- It's called a **variable** because it can change, as opposed to a **constant** which remains the same

Declaring Variables

- Variables are declared (created) by using the "var" keyword.
- To declare a new variable called "age" you would write:
  - var age;
- A value can be assigned to a variable by using the "=" operator:
  - age = 21;
- It is possible to declare a new variable and assign a value to it at the same time:
  - var age = 21;

Using Variables

- var **age** = 21;
- **age** + 1;
- age now equals 22

- var **name** = "John";
- var **greeting** = "Hello " + **name**;

Using JavaScript in your HTML

- Inside the website directory, make a JavaScript folder called "js"
- Inside the "js" folder make a JavScript file called "scripts.js"
- Link to the js file inside your html file by using the script tag

```
<script src="js/scripts.js"></script>
```

- The script tag goes at the end of body…it must be the final tag inside body
- Check that it works by adding the following code inside scripts.js:

```
alert("This works!");
```

Alert Function

- The **alert** function is some pre-written code which allows you to display some text in an alert box (popup window)
- You can **call** (use) the alert function by typing alert(x) where 'x' is any valid value/expression.
  - `alert("Hello World");`
  - `alert("Hello " + "World");`
  - `alert("My age is " + 18);`
  - `alert(999);`
  - `alert(100 > 50);`

Functions

- A function is a block of code that performs some task.
- Instead of typing the same code again and again, we can store the code in a function, give the function a name and then refer to that block of code by the function name.

```
function greeting() {
    var name = "John";
    alert("Hello " + name);
}
```

Calling Functions

- A function isn't normally executed (run) unless you "call" it
- You call a function by writing its name
- e.g. to use the "greeting" function from the last slide you would write:

```
greeting();
```

Function Parameters

- The greeting function doesn't have any parameters, which is why there was nothing inside the brackets
- Sometimes it is necessary to "pass" (give) some data to the function so that it can do something with it
- An example is the alert function. Remember how you need to put a String inside the alert function so that it knows what text to display.
- If you want the previous greeting function to work for any name you pass in, you would rewrite the function as follows:

```
function greeting(name) {
    alert("Hello " + name);
}
```

- You then call this function by writing:

```
greeting("John");
```

Functions with Multiple Parameters

- You can write a function that takes multiple parameters, just separate those parameters with a comma

```
function person(name, age) {
    alert(name + " is " + age);
}
```

- Call the function like this:

```
person("John", 20);
```

- This function will create an alert box that says "John is 20"

Conditional Statements

- If a boolean expression is true, do something, otherwise do something else

```
var age = 17;
if (age < 18) {
   alert("Child");
} else {
   alert("Adult");
}
```

- What will be the output of this code?

Conditional Statements

- Sometimes you want to check multiple conditions, in that case you use "else ifs":

```
var age = 18;
if (age < 18) {
   alert("Child");
} else if (age >= 60) {
   alert("Senior");
} else {
   alert("Adult");
}
```

- What will be the output of this code?

jQuery

- JavaScript can be verbose – need to write a lot of code to achieve simple tasks
- jQuery is a library which contains a lot of pre-written JavaScript code and makes it easier to do certain things
- The code for jQuery is stored in a CDN, you just need to link to the URL in your html
- Make sure that the link for the jQuery CDN appears BEFORE the link to any other JavaScript files
- Go to: https://cdnjs.com/libraries/jquery , copy the second link (jquery.js) and add it to a script tag as follows:

```
<script src="https://cdnjs.cloudflare.com/ajax/libs/jquery/3.3.1/jquery.min.js"></script>
```

jQuery Syntax

- jQuery allows you to select an element and perform some action on it
- ***$("selector").action()***
- The dollar sign says this is a jQuery function
- The selector can be any html element, ID or Class
- The action is any jQuery function that can be applied to the selected element

jQuery Document Ready

- Your JavaScript file should start with the following jQuery function

```
$(document).ready(function(){
  // jQuery code goes here...
});
```

- Any code inside the document ready function will only run once the html file has fully loaded
- Most of your code will go inside this function

jQuery Events

- There are some builtin events in JavaScript, so you can run some specific code when a specific event occurs
- An example is the "on click" event:

```
$("button").click(function()  {
   alert("You just clicked on a button");
});
```

jQuery Show/Hide

- There is a builtin function to hide elements:

```
$("h1").hide();
```

- There is a builtin function to show elements:

```
$("h1").show();
```