**Javascript**

Introduction to Javascript

Javascript is a scripting or programming language that allows you to implement features on webpages making your webpage interactive.eg.

* Update content and display messages to the user as part of a web page
* Animate images or create images that change when you move the mouse over them
* Validate the contents of a form and make calculations.

In web development there are different types of scripting: *Server side* and *Clientside*

**Server-side Javascript**

Server-side scripting refers to scripts that run on the web server, which then sends results to the web browser. If you have ever submitted a form on a website, which includes a search engine, you have experienced the results of a server-side script

**Clientside Javascript**

Clientside scripting refers to scripts that run within your web browser and is commonly used to validate data before sending it to the web server.

**Including Javascript in your HTML**

**Console**

The console is a panel that displays important messages, like errors. Web browsers have this built in option for programmers to see code or to print values to the console so we can see the work that we are doing.

Another action or method built into the console object is the **.log()** method.

console.log(5);

This example will log 5 to the console. The semicolon indicates the end of the line or statement.

To enter the console:

1. Hit **F12** on your computer while you are in the web browser, or right-click and select  **inspect**, you will see a screen appear.
2. If you click on the console tab you can enter your code directly.

**Comments**

There are two types of code comments in Javascript:

*Single line comment* will comment out a single line and is indicated by two forward slashes //.

// This is a single line comment.

Multi-line comment will comment out multiple lines and is indicated with /\* to begin the comment, and \*/ to end the comment.

/\*

This is a multi-line comment.

None of this is going to run!

\*/

Alert()

The **alert()** command will create a popup box that provides a message.

alert(“Hi”);

Prompt()

The **prompt()** method displays a dialog box that prompts the user for input.

prompt(“How old are you?”);

**Variables**

**Variables** is a container used to store values. Think of it as a box or memory card that stores information (usernames, account number, personalized greeting) that lives in a computer’s memory. **Variables** contain values that are represented by the content, and the name is represented with the label.

**var winners = 2; //stores a number/ numeric value.**

**var name = “Duke”; //stores a string of characters**.

There are only a few things you can do with variables:

1. Create a variable with a descriptive name.
2. Store or update information stored in a variable.
3. Reference or “get” information stored in a variable.

**Create a variable / Declaring Variables**

Creating a variable in javascript is called “declaring” a variable. You declare a variable by using the keywords **var, let and const.**

The **var**  keyword was used to declare variables since Javascript was created. It is confusing and error-prone.

The **let** keyword is the new and recommended way of declaring variables in Javascript.

The **const** keyword is used to declare a constant variable which value cannot be changed.

**let** myName;  
// This is a **variable declaration**.It does not have a value yet.

\*The console output will show **undefined**.It is not an error! That just means the console has nothing to tell you.

There are a few general rules for naming variables:

* Variable names cannot start with numbers.
* Variable names are case sensitive, so **myName** and **myname** would be different variables. It is bad practice to create two variables that have the same name using different cases.
* We use *camelCase* to create variable names that has multiple words eg. ageOfBuyer
* Variable names can begin with a letter, underscore( \_ ), or a dollar sign $.
* Variable names cannot be the same as *keywords*. For a comprehensive list of keywords check out [MDN’s keyword documentation](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Lexical_grammar#Keywords).

**Assigning a value**

To assign a value to a variable we use the equal sign( **=** ).

**let** age = 12;

This is called a Javascript statement.

**Reassigning a value**

To reassign a value to a variable we only use the variable name.

age = 15;

c

However, a **const** variable cannot be reassigned because it is constant. If you try to reassign a **const** variable, you’ll get a **TypeError**.

Constant variables *must* be assigned a value when declared. If you try to declare a **const** variable without a value, you’ll get a **SyntaxError**.

**Data Types**

**Data types** are the classifications we give to different kinds of data that we use for programing. In Javascript, there are eight fundamental data types:

* **Number**

Any number, including numbers with decimals: 4, 1516, 23.42.

* **String**

Any group of characters(letters, numbers, spaces, symbols, etc.) surrounded by single quotes: **‘ … ’** or double quotes **“…”.**

Think of a string data type as text.

* **Boolean**

This data type only has two possible values – either true or false(without quotes).

Think of booleans as on and off switches or answers to a yes and no question.

* **Null**

This data type represents the intentional absence of a value, and is represented by the keyword null (without quotes).

* **Undefined**

This data type is indicated by the keyword undefined (without quotes).It also represents the absence of a value but it has a different use than null.

undefined means that the given value does not exist.

* **BigInt**

Any numeric value that is too large to be represented by the number data type.

900719925740991n.

* **Symbol**

Symbols are unique identifiers, useful in more complex coding.

* **Object**

Collections of related data

*\*The first 7 are consider primitive data types. The most basic data types in the language. Objects are more complex.*

**Strings**

The string data type is used to store a text value.It is surrounded by single quotes or double quotes.

**let** greeting = “Hello”;

Due to the fact that quotation marks are used to indicate strings, you need to be careful when using apostrophes and quotes in strings. If you attempt to use them Javascript will end the string and throw an error.

**let** quote = “He said,”I learnt how to code!””;

>>Uncaught SyntaxError:Unexpected identifier ‘I’

l

There are three ways to fix the error.

1. By using the opposite string syntax.

**let** quote = ‘He said,”I learnt how to code!”’;

>>He said,”I learnt how to code!”

1. Use escape characters: This is done by using a backslash( \ ) before the character.

**let** quote = “He said,\”I learnt how to code!\””;

>>He said,”I learnt how to code!”

1. Use Template Literals: special template strings in which you can use variables directly

**let** language = “Javascript”;

**let** message = `Let’s learn ${language}.`;

>>Lets learn Javascript.

**Concatenating strings/Joining strings**

We use the **+** operator to join two strings.

**let** greeting = “Hello”;

**let** myName = “Nick”;

console.log(greeting + myName)

>>”HelloNick”

>>Lets learn Javascript.

\**Adding a extra space inside the quotes will create a space in the final string.*

**let** greeting = “Hello ”;

>>”Hello Nick”

Another method used to join strings is **.concat()**