1). **What is JavaScript**?

2). **What is JavaScript used for**?

3). **Setting Up Your Development Environment**

4). **Meet the console tab of DevTools**

5). **JavaScript syntax**

6). **Adding JavaScript to Your Web Pages**

7). **Variables in JavaScript**.

8). **Data Types in JavaScript**.

1). **What is JavaScript**?

JavaScript is the most popular and widely used client-side scripting language. Client-side scripting refers to scripts that run within your web browser. JavaScript is designed to add interactivity and dynamic effects to the web pages by manipulating the content returned from a web server.

JavaScript was originally developed as LiveScript by Netscape in the mid 1990s. It was later renamed to JavaScript in 1995, and became an ECMA standard in 1997. Now JavaScript is the standard client-side scripting language for web-based applications, and it is supported by virtually all web browsers available today, such as Google Chrome, Mozilla Firefox, Apple Safari, etc.

JavaScript is an object-oriented language, and it also has some similarities in syntax to Java programming language. But, JavaScript is not related to Java in any way.

JavaScript is officially maintained by ECMA (European Computer Manufacturers Association) as ECMAScript. ECMAScript 6 (or ES6) is the latest major version of the *ECMAScript standard*.

2). **What Can Do with JavaScript**?

   a). **Web** & **Server Apps**

The most common answer to the question "What is JavaScript used for?" is something like *'creating interactive web elements'.* But this isn't all. The creation of new libraries and frameworks is allowing web developers to build a lot of back-end programs with JavaScript. These include things like web apps and server apps. JavaScript is becoming just as essential for back-end web developers as it is for front-end developers.

   b). **Serverless Websites**

Whilst still a relatively new concept, Serverless Websites could be a new direction for website hosting. Making use of services like AWS Lambda, Serverless Websites can be a relatively efficient way to host a website.

   c). **Progressive Web Apps**

Thanks to companies like Google and Mozilla. PWAS are an amazing new technology. For those of you that have not heard of PWAs, they're a technology that allow users to install a website to their phone or laptop just like any other native applications providing a range of benefits.

d). **Animate Websites**

Even simple animations can add a lot to a website; they keep you engaged and make you want to just keep on going to see what more it has to give. Making websites look cool and engaging is a great way to make use of animations, but they can have other more practical benefits as well.

3). <u>**Setting Up Your Development Environment**</u>

To Write basic JavaScript code you need to have a Text editor and modern web browsers.

– **Text editors**.

A text editor is a computer program that lets a user enter, change, store, and usually print text (characters and numbers, each encoded by the computer and its input and output devices, arranged to have meaning to users or to other programs).

Examples of text editor you will need are: VS Code, Atom, Sublime etc…

– **Browser**.

A browser is a computer program with a graphical user interface for displaying HTML files, used to navigate the World Wide Web.
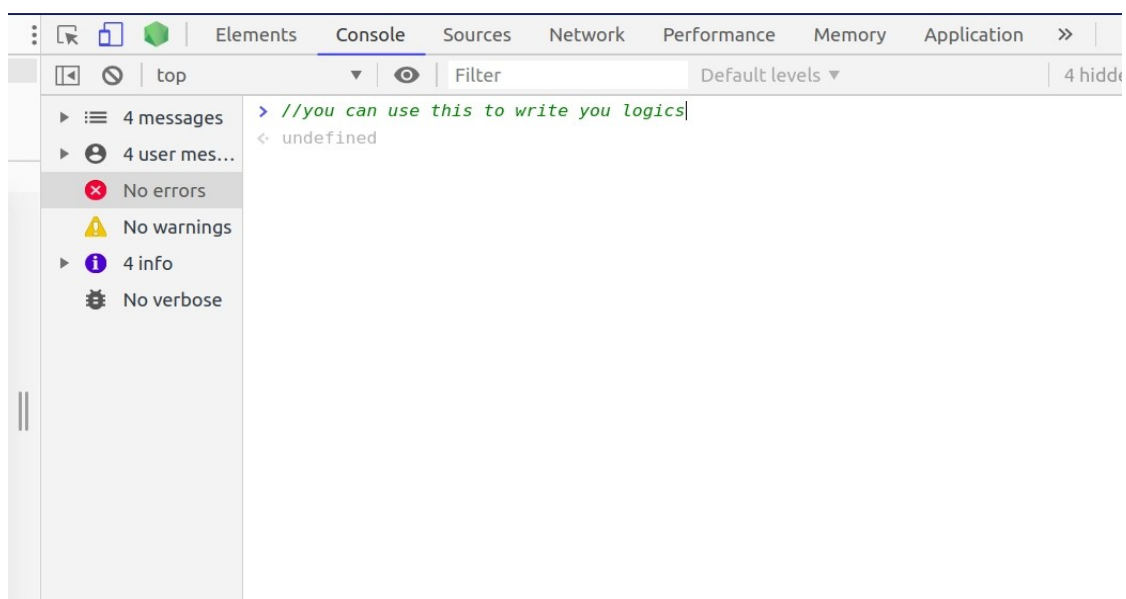
4). **Meet the console tab of DevTools**.

Web development tools allow you to test and debug the JavaScript code. Web development tools are often called devtools.

Modern web browsers such as Google Chrome, Firefox, Edge, Safari, and Opera provide the devtools as built–in features. Generally, devtools allow you to work with a variety of web technologies such as HTML, CSS, DOM, and JavaScript.

In this section, you will learn how to open the Console tab of the devtools to view messages output by JavaScript.

Press F12 on windows or Cmd+Opt+J if you are on Mac to view the Console tab of the dev tool which should appear as show below:

Note That –

You open the Console tab of the devtools in Chrome, Firefox and Edge using F12 and have similar user interfaces.

If you are using Safari browser on Mac, you need to enable the Developer Menu first. (If you are having trouble enabling Developers Menu we can guide you through).

5). **JavaScript Syntax**.

The syntax of JavaScript is the set of rules that define a correctly structured JavaScript program.

A JavaScript consists of JavaScript statements that are placed within the <script> </script> HTML tags in a web page, or within the external JavaScript file having .js extension.

It is worthy noting that JavaScript is case-sensitive. This means that variables, language keywords, function names, and other identifiers must always be typed with a consistent capitalization of letters.

For example, the variable myVar must be typed as myVar not MyVar or myvar. Similarly, the method name getElementById() must be typed with the exact case not as getElementByID().

A *comment* is simply a line of text that is completely ignored by the JavaScript interpreter. Comments are usually added with the purpose of providing extra information pertaining to source code.

It will not only help you understand your code when you look after a period of time but also others who are working with you on the same project. Avoid too much comment in your code.

JavaScript support single-line as well as multi-line comments. Single-line comments begin with a double forward slash (//), followed by the comment text. Whereas, a multi-line comment begins with a slash and an asterisk (/*) and ends with an asterisk and slash (*/).

6). **Adding JavaScript to Your Web Pages**.

There are typically three ways to add JavaScript to a web page:

• Embedding the JavaScript code between a pair of <script> and </script> tag.

• Creating an external JavaScript file with the .js extension and then load it within the page through the src attribute of the <script> tag.

```
<script src ="static/js/hello.js"> </script>
```

• Placing the JavaScript code directly inside an HTML tag using the special tag attributes such as onclick, onmouseover, onkeypress, onload, etc.

```
<button onclick ="myFunction()">Click Me</button>
```

   a). **Embedding the JavaScript Code**

You can embed the JavaScript code directly within your web pages by placing it between the <script> and </script> tags. The <script> tag indicates to the browser that the contained statements are to be interpreted as executable script and not HTML.

Example:

```
<!DOCTYPE html>

<html lang="en">

<head>

 <meta charset="UTF-8">

 <title>#100DaysOfCode</title>

 </head>

<body>

 <script>

var greet = "Hello World!";

 document.write(greet); // Prints: Hello World!

</script>

 </body>

</html>
```

b). **Calling an External JavaScript File**.

You can also place your JavaScript code into a separate file with a .js extension, and then call that file in your document through the src attribute of the <script> tag, like this:

```
<script src ="static/js/hello.js"> </script>
```

This is useful if you want the same scripts available to multiple documents. It saves you from repeating the same task over and over again, and makes your website much easier to maintain.

Well, let's create a JavaScript file named "hello.js" and place the following code in it:

```
// A function to display a message

function sayHello() {

alert("Hello World!");

} // Call function on click of the button

document.getElementById("myBtn").onclick = sayHello;
```

Now, you can call this external JavaScript file within a web page using the <script> tag, like this:

```
<!DOCTYPE html>

<html lang="en">

 <head>

<meta charset="UTF-8">

 <title> JavaScript from External File</title>

</head>

 <body>

<button type="button" id="myBtn">Click Me</button>

<script src="js/hello.js"></script>

</body>

</html>
```

c). **Placing the JavaScript Code Inline**

You can also place JavaScript code inline by inserting it directly inside the HTML tag using the special tag attributes such as onclick, onmouseover, onkeypress, onload, etc.

However, you should avoid placing large amount of JavaScript code inline as it clutters up your HTML with JavaScript and makes your JavaScript code difficult to maintain.

Here's an example:

```
<!DOCTYPE html>

 <html lang="en">

 <head>

<meta charset="UTF-8">

 <title>Inlining JavaScript</title>

</head>

<body>

<button onclick="alert('Hello World!')">Click Me</button>

</body>

 </html>
```

7). **Variables in JavaScript**.

Variable is a named storage of data. We can use variables to store goodies, visitors, and other data. For example in a chat application we need to store users, messages, and much more, in such a case we will need to use variables. In layman's language Variables are used to store this information.

JavaScript uses reserved keywords var and let to declare a variable. A variable must have a unique name. You can assign a value to a variable using equal to (=) operator when you declare it or before using it.

Const is used to declare a constant and values of a constants can't change, that why we excluded.

The following are different ways of declaring variables:

let <variable-name>; // declared a variable without assigning a value

var <variable-name>; // declared a variable without assigning a value

let <variable-name> = <value>;

var <variable-name> = <value>;

Also you can declare multiple variables a single line separated by comma.

Examples:

let name = 'anonymous', age = 23, country;

JavaScript allows variable declaration without var keyword but you must assign a value when you declare a variable without var keyword. The scope of the variables declared without var keyword become global irrespective of where it is declared. Global variables can be accessed from anywhere in the web page. We will cover more about scope of a variable later.

age = 23;

Note that:

Variables in JavaScript are loosely-typed variables. It can store value of any data type through out it's life time.

JavaScript allows multiple white spaces and line breaks when you declare a variable with let/var keyword.

Var and let keywords.

Var is rather a keyword which defines a variable globally regardless of block scope let gives you the privilege to declare variables that are limited in scope to the block, statement of expression unlike var.

Let allows you to declare variables that are limited in scope to the block, statement, or expression on which it is used. This is unlike the var keyword, which defines a variable globally, or locally to an entire function regardless of block scope.

Variables declared by let have as their scope the block in which they are defined, as well as in any contained sub-blocks. In this way, let works very much like var.

8). **Data Types in JavaScript**.

Data types basically specify what kind of data can be stored and manipulated within a program.

There are six basic data types in JavaScript which can be divided into three main categories: primitive (or primary), composite (or reference), and special data types. String, Number, and Boolean are primitive data types. Object, Array, and Function (*which are all types of objects*) are composite data types. Whereas Undefined and Null are special data types.

Primitive data types can hold only one value at a time, whereas *composite data* types can hold collections of values and more complex entities.

a). The String Data Type.

The string data type is used to represent textual data (i.e. sequences of characters). Strings are created using single or double quotes surrounding one or more characters.

Examples:

var a = 'Hi there!'; // using single quotes

var b = "Hi there!"; // using double quotes

You can include quotes inside the string as long as they don't match the enclosing quotes.

var a = "Let's have a cup of coffee."; // single quote inside double quotes

b). The Number Data Type.

The number data type is used to represent positive or negative numbers with or without decimal place, or numbers written using exponential notation e.g. $1.5e{-4}$ (equivalent to $1.5 \times 10{-4}$).

c). The Boolean Data Type.

The Boolean data type can hold only two values: true or false. It is typically used to store values like yes (true) or no (false), on (true) or off (false), etc.

d). The Undefined Data Type.

The undefined data type can only have one value–the special value undefined. If a variable has been declared, but has not been assigned a value, has the value undefined.

e). **The Null Data Type**.

This is another special data type that can have only one value–the null value. A null value means that there is no value. It is not equivalent to an empty string (") or 0, it is simply nothing.

A variable can be explicitly emptied of its current contents by assigning it the null value.

f). **The Object Data Type**.

The object is a complex data type that allows you to store collections of data.

An object contains properties, defined as a key–value pair. A property key (name) is always a string, but the value can be any data type, like strings, numbers, booleans, or complex data types like arrays, function and other objects.

g).**The Array Data Type**.

An array is a type of object used for storing multiple values in single variable. Each value (also called an element) in an array has a numeric position, known as its index, and it may contain data of any data type–numbers, strings, booleans, functions, objects, and even other arrays. The array index starts from 0, so that the first array element is arr[0] not arr[1].

h).The Function Data Type.

The function is callable object that executes a block of code. Since functions are objects, so it is possible to assign them to variables,

```
var greeting = function(){
    return "Hello World!";
}
```

The typeof Operator

The **typeof** operator can be used to find out what type of data a variable or operand contains. It can be used with or without parentheses (typeof(x) or typeof x).

The **typeof** operator is particularly useful in the situations when you need to process the values of different types differently, but you need to be very careful, because it may produce unexpected result in some cases,

Conclusion.

JavaScript is one of the most popular programming languages in the world, especially among web app developers. Although it was designed to help front-end web developers create interactive elements, the use cases of JavaScript have rapidly expanded to include things like back-end web

development, game creation, and even mobile app development. If any of these things interest you, you should consider learning JavaScript.

Discovering what is JavaScript used for will open your eyes to the benefits of learning how to code. Learning JavaScript will make you more employable, will give you the chance to become an entrepreneur or a freelancer, and will give you a whole lot more job security.

Most importantly, make sure that you have fun while you're learning how to code with JavaScript!

Good Luck and best wishes.

Lux Tech Academy
Call : *0794598834*
Email : *luxtinc@gmail.com*
Website : *https://luxtechacademy.com*