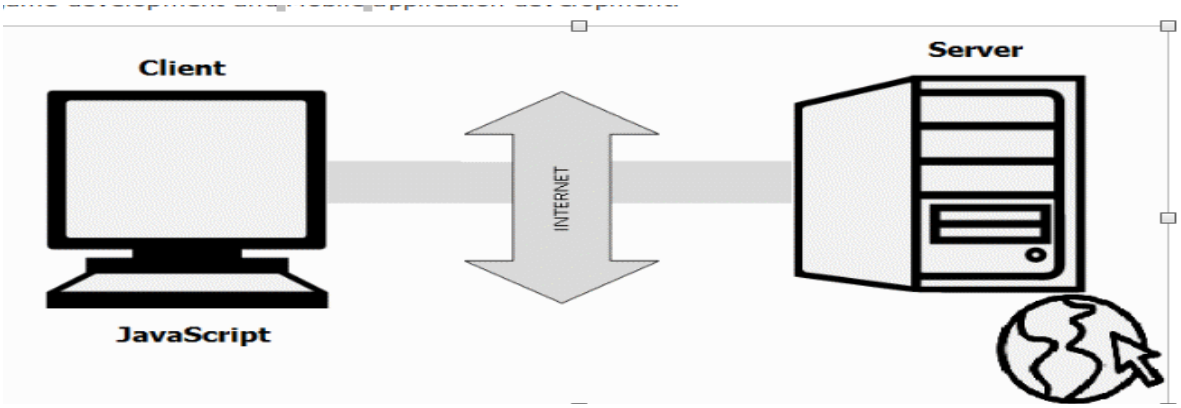
Introduction

What is JavaScript?

JavaScript is a very powerful **client-side scripting language**. JavaScript is used mainly for enhancing the interaction of a user with the webpage. In other words, you can make your webpage more lively and interactive, with the help of JavaScript. JavaScript is also being used widely in game development and Mobile application development.

JavaScript is a *lightweight, cross-platform*,*single-threaded,*and *interpreted compiled* programming language which is also known as the scripting language for webpages. It is well-known for the development of web pages, and many non-browser environments also use it. JavaScript is a weakly typed language(dynamically typed). JavaScript can be used for Client-side developments as well as Server-side developments. JavaScript is both an imperative and declarative type of language. JavaScript contains a standard library of objects, likeArray,Date, andMath, and a core set of language elements like operators,control structures, and statements.



•Client-side:It supplies objects to control a browser and its **Document Object Model (DOM).**Like if client-side extensions allow an application to place elements on an HTML form and respond to user events such as mouse clicks,form input, and page navigation. Useful libraries for the client side are**AngularJS**,**ReactJS**,**VueJS,**and so many others.

•Server-side:It supplies objects relevant to running JavaScript on a server. For if the server-side extensions allow an application to communicate with a database, and provide continuity of information from one invocation to another of the application, or perform file manipulations on a server. The useful framework which is the most famous these days is **node.js**.

•Imperative language –In this type of language we are mostly concerned about how it is to be done. It simply controls the flow of computation. The procedural programming approach, object, oriented approach comes under this as async await we are thinking about what is to be done further after the async call.

•Declarative programming –In this type of language we are concerned about how it is to be done, basically here logical computation requires. Her main goal is to describe the desired result without direct dictation on how to get it as the arrow function does.

JavaScript can be added to your HTML file in **two ways**:

•Internal JS:We can add JavaScript directly to our HTML file by writing the code inside the <script> tag. The <script> tag can either be placed inside the <head> or the <body> tag according to the requirement.

•**External JS**:We can write JavaScript code in another files having an extension.js and then link this file inside the <head> tag of the HTML file in which we want to add this code.

History of JavaScript:It was created in 1995 by Brendan Eich while he was an engineer at Netscape. It was originally going to be named LiveScript but was renamed. Unlike most programming languages, JavaScript language has no concept of input or output. It is designed to run as a scripting language in a host environment, and it is up to the host environment to provide mechanisms for communicating with the outside world. The most common host environment is the browser.

Features of JavaScript:According to a recent survey conducted by Stack Overflow, JavaScript is the most popular language on earth.
With advances in browser technology and JavaScript having moved into the server with Node.js and other frameworks, JavaScript is capable of so much more. Here are a few things that we can do with JavaScript:

•JavaScript was created in the first place for DOM manipulation. Earlier websites were mostly static, after JS was created dynamic Web sites were made.

•Functions in JS are objects. They may have properties and methods just like other objects. They can be passed as arguments in other functions.

•Can handle date and time.

•Performs Form Validation although the forms are created using HTML.

•No compiler is needed.

Applications of JavaScript:

•Web Development:Adding interactivity and behavior to static sites JavaScript was invented to do this in 1995. By using AngularJS that can be achieved so easily.

•Web Applications:With technology, browsers have improved to the extent that a language was required to create robust web applications. When we explore a map in Google Maps then we only need to click and drag the mouse. All detailed view is just a click away, and this is possible only because of JavaScript. It uses Application Programming Interfaces(APIs) that provide extra power to the code. The Electron and React are helpful in this department.

•Server Applications:With the help of Node.js, JavaScript made its way from client to server and Node.js is the most powerful on the server side.

•Games:Not only in websites, but JavaScript also helps in creating games for leisure. The combination of JavaScript and HTML 5 makes JavaScript popular in game development as well. It provides the EaseJS library which provides solutions for working with rich graphics.

•Smartwatches:JavaScript is being used in all possible devices and applications. It provides a library PebbleJS which is used in smartwatch applications. This framework works for applications that require the Internet for their functioning.

•Art:Artists and designers can create whatever they want using JavaScript to draw on HTML 5 canvas, and make the sound more effective also can be used **p5.js** library.

•Machine Learning:This JavaScript ml5.js library can be used in web development by using machine learning.

•Mobile Applications:JavaScript can also be used to build an application for non-web contexts. The features and uses of JavaScript make it a powerful tool for creating mobile applications. This is a Framework for building web and mobile apps using JavaScript. Using React Native, we can build mobile applications for different operating systems. We do not require to write code for different systems. Write once use it anywhere!

Limitations of JavaScript:

•Security risks:JavaScript can be used to fetch data using AJAX or by manipulating tags that load data such as <img>, <object>, <script>. These attacks are called cross-site script attacks. They inject JS that is not part of the site into the visitor’s browser thus fetching the details.

•Performance:JavaScript does not provide the same level of performance as offered by many traditional languages as a complex program written in JavaScript would be comparatively slow. But as JavaScript is used to perform simple tasks in a browser, so performance is not considered a big restriction in its use.

•Complexity:To master a scripting language, programmers must have a thorough knowledge of all the programming concepts, core language objects, and client and server-side objects otherwise it would be difficult for them to write advanced scripts using JavaScript.

•Weak error handling and type checking facilities:It is a weakly typed language as there is no need to specify the data type of the variable. So wrong type checking is not performed by compile.

Why JavaScript is known as a lightweight programming language?
JavaScript is considered lightweight due to the fact that it has low CPU usage, is easy to implement, and has a minimalist syntax. Minimalist syntax as in, has no data types. Everything is treated here as an object. It is very easy to learn because of its syntax similar to C++ and Java.

A lightweight language does not consume much of your CPU's resources. It doesn't put excess strain on your CPU or RAM. JavaScript runs in the browser even though it has complex paradigms and logic which means it uses fewer resources than other languages. For example, NodeJs, a variation of JavaScript not only performs faster computations but also uses fewer resources than its counterparts such as Dart or Java.

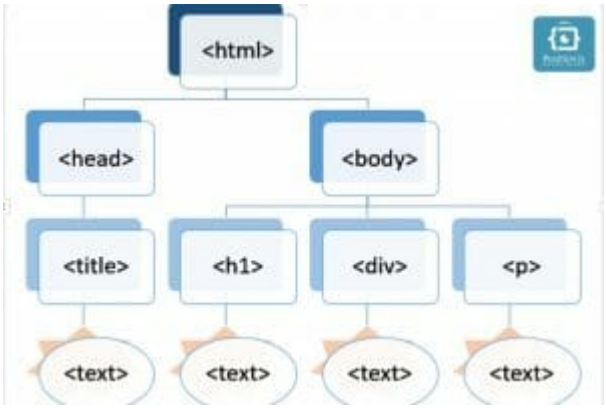
Additionally, when compared with other programming languages, it has fewer in-built libraries or frameworks, contributing as another reason for it being lightweight. However, this brings a drawback in that we need to incorporate external libraries and frameworks.

Is JavaScript compiled or interpreted or both?
JavaScript is both compiled and interpreted. In the earlier versions of JavaScript, it used only the interpreter that executed code line by line and shows the result immediately. But with time the performance became an issue as interpretation is quite slow. Therefore, in the newer versions of JS, probably after the V8, the JIT compiler was also incorporated to optimize the execution and display the result more quickly. This JIT compiler generates a bytecode that is relatively easier to code. This bytecode is a set of highly optimized instructions.
The V8 engine initially uses an interpreter, to interpret the code. On further executions, the V8 engine finds patterns such as frequently executed functions, and frequently used variables, and compiles them to improve performance.

JavaScript Document: Properties and Functions

by Position is Everything

A **JavaScript document can be any web page that is loaded in a browser** to provide access to the content on that page, often referred to as the DOM tree. This tree has a body, header, title, and many other elements. The functionality is **global** for a document and it is a process that is associated with the page's URL. In this guide, we will be discussing **JavaScript document methods** with examples. Let's start the process!



What Is Document in JavaScript?

Document in JavaScript is an **object API** in the host environment normally seen in web pages, interacting with the user agent we normally refer to as browser. JavaScript document objects describe the JavaScript document properties and the methods. For example in HTML documents, the text/HTML content type will be used or HMTLDocument interface. For other types such as XML and SVG and Core, they have their interface.

Pros and Cons of JavaScript Document

Let's quickly glance over the advantages and disadvantages of **JavaScript document functions**

– Pros:

- The time is reduced as JavaScript is a client-side script, hence; speedy processes.
- Easy to operate and understand as the structure is suitable for both users and developers.
- Popular platforms such as the leading search engines and e-commerce companies such as Amazon use JavaScript.

– Cons:

- The client-side accessibility makes JavaScript document functions less private. Hence, malware attacks may be common.
- Different browsers have different responses towards JavaScript, hence the flexibility factor is affected.

●If a single error occurs, it will stop the entire code in JavaScript on a page or document in JavaScript.

Constructor

To create a constructor object, we can use Document(). For all the objects, do remember that the **HTML DOM document holds the "owner" position**.

HTML DOM Document

As we already know the **document object works with the web page** and we can access the page to find the elements. We can also add or delete or change documents and content on the page. In the examples below, we will be seeing the **methods** to perform the aforementioned functions.– To Find

HTML Elements:

Method To Use	Description
document.getElementById(id)	By using the element id, we can find the element required.
document.getElementsByTagName(name)	If trying to find an element by the tag name, we can use this.
document.getElementsByClassName(name)	If we want to find by class name, we can use this method.

– To Change HTML Elements:

Property To Use	Description
element.innerHTML = new HTML content	We can change the innerHTML of the element.
element.attribute = new value	We can change the attribute value of the element.
element.style.property = new style	By this, we can change the style of the element.
element.setAttribute(attribute, value)	Using this, we can change the attribute value of the HTML element. Do note this and the element.attribute = value are similar.

JavaScript Statements

JavaScript statements are composed of:

Values, Operators, Expressions,Keywords, and Comments.

This statement tells the browser to write "Hello Dolly." inside an HTML element with id="demo"

```
document.getElementById("demo").innerHTML="Hello Dolly.";
```

Most JavaScript programs contain many JavaScript statements.

The statements are executed, one by one, in the same order as they are written.

JavaScript programs (and JavaScript statements) are often called JavaScript code.

Semicolons ;

Semicolons separate JavaScript statements.

Add a semicolon at the end of each executable statement:

Examples

```
let a, b, c;  // Declare 3 variables
a = 5;        // Assign the value 5 to a
b = 6;        // Assign the value 6 to b
c = a + b;    // Assign the sum of a and b to c
```

When separated by semicolons, multiple statements on one line are allowed:

```
a = 5; b = 6; c = a + b;
```

JavaScript Functions

A JavaScript function is a block of code designed to perform a particular task.A JavaScript function is executed when "something" invokes it (calls it).

Example

```
// Function to compute the product of p1 and p2
function myFunction(p1, p2) {
  return p1 * p2;
}
```

JavaScript Function Syntax

A JavaScript function is defined with the `function` keyword, followed by a **name**, followed by parentheses **()**.

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

The parentheses may include parameter names separated by commas:
(parameter1, parameter2, ...)

The code to be executed, by the function, is placed inside curly brackets: **{}**

```
function name(parameter1, parameter2, parameter3) {
  // code to be executed
}
```

Function **parameters** are listed inside the parentheses () in the function definition.

Function **arguments** are the **values** received by the function when it is invoked.

Inside the function, the arguments (the parameters) behave as local variables.

Function Invocation

The code inside the function will execute when "something" **invokes** (calls) the function:

- When an event occurs (when a user clicks a button)
- When it is invoked (called) from JavaScript code
- Automatically (self invoked)

Functions Used as Variable Values

Functions can be used the same way as you use variables, in all types of formulas, assignments, and calculations.

Example

Instead of using a variable to store the return value of a function:

```
let x = toCelsius(77);
let text = "The temperature is " + x + " Celsius";
```

You can use the function directly, as a variable value:

```
let text = "The temperature is " + toCelsius(77) + " Celsius";
```

Local Variables

Variables declared within a JavaScript function, become **LOCAL** to the function.

Local variables can only be accessed from within the function.

Example

```
// code here can NOT use carName

function myFunction() {
  let carName = "Volvo";
  // code here CAN use carName
}

// code here can NOT use carName
```

JavaScript Objects

> In real life, a car is an **object**.

A car has **properties** like weight and color, and **methods** like start and stop:

> All cars have the same **properties**, but the property **values** differ from car to car.

All cars have the same **methods**, but the methods are performed **at different times**.

Objects are variables too. But objects can contain many values.
This code assigns **many values** (Fiat, 500, white) to a **variable** named car:

```
const car = {type:"Fiat", model:"500", color:"white"};
```

Object Properties

The **name:values** pairs in JavaScript objects are called **properties**:

Property	Property Value
firstName	John
lastName	Doe
age	50
eyeColor	blue

Accessing Object Properties
You can access object properties in two ways:

```
objectName.propertyName
```

or

```
objectName["propertyName"]
```

Example1

```
person.lastName;
```

JavaScript objects are containers for **named values** called properties.

Object Methods

Objects can also have **methods**.

Methods are **actions** that can be performed on objects.

Methods are stored in properties as **function definitions**.

Property	Property Value
firstName	John
lastName	Doe
age	50
eyeColor	blue
fullName	function() {return this.firstName + " " + this.lastName;}

Introduction to AJAX

AJAX is a web development technique for creating interactive web applications. If you know JavaScript, HTML, CSS, and XML, then you need to spend just one hour to start with AJAX.

Why to Learn Ajax?

AJAX stands for **A**synchronous **Ja**vaScript and **X**ML. AJAX is a new technique for creating better, faster, and more interactive web applications with the help of XML, HTML, CSS, and Java Script.

- Ajax uses XHTML for content, CSS for presentation, along with Document Object Model and JavaScript for dynamic content display.
- Conventional web applications transmit information to and from the sever using synchronous requests. It means you fill out a form, hit submit, and get directed to a new page with new information from the server.
- With AJAX, when you hit submit, JavaScript will make a request to the server, interpret the results, and update the current screen. In the purest sense, the user would never know that anything was even transmitted to the server.
- XML is commonly used as the format for receiving server data, although any format, including plain text, can be used.
- AJAX is a web browser technology independent of web server software.
- A user can continue to use the application while the client program requests information from the server in the background.
- Intuitive and natural user interaction. Clicking is not required, mouse movement is a sufficient event trigger.
- Data-driven as opposed to page-driven.

Rich Internet Application Technology

AJAX is the most viable Rich Internet Application (RIA) technology so far. It is getting tremendous industry momentum and several tool kit and frameworks are emerging. But at the same time, AJAX has browser incompatibility and it is supported by JavaScript, which is hard to maintain and debug.

AJAX is Based on Open Standards

AJAX is based on the following open standards −

- Browser-based presentation using HTML and Cascading Style Sheets (CSS).
- Data is stored in XML format and fetched from the server.
- Behind-the-scenes data fetches using XMLHttpRequest objects in the browser.
- JavaScript to make everything happen.

AJAX - Technologies

AJAX cannot work independently. It is used in combination with other technologies to create interactive webpages.

JavaScript

- Loosely typed scripting language.
- JavaScript function is called when an event occurs in a page.
- Glue for the whole AJAX operation.

DOM

- API for accessing and manipulating structured documents.
- Represents the structure of XML and HTML documents.

CSS

- Allows for a clear separation of the presentation style from the content and may be changed programmatically by JavaScript

XMLHttpRequest

- JavaScript object that performs asynchronous interaction with the server.