Features of Javascript:

JavaScript is one of the most popular programming languages which includes numerous features when it comes to web development.

Some of the features of Javascript are:-

## Light-Weight Scripting Language:

JavaScript is a lightweight scripting language because it is made for data handling in the browser or the client side.Because JavaScript is meant for client-side execution for web applications, hence the lightweight nature of JavaScript is a great feature.

1. Dynamic Typing

* JavaScript supports dynamic typing which means types of the variable are defined based on the stored value.
* To understand this, in languages like Java, we explicitly mention that a particular variable will store a certain type of data, whereas in JavaScript we do not have to provide the data type while declaring a variable.
* In JavaScript, we just have to use var or let keyword before the variable name to declare a variable without worrying about its type.

1. Object-Oriented Programming support

Starting from ES6, the concept of class and OOP is better defined.

Also, in JavaScript, two important principles with OOP in JavaScript are:

* + Object Creation Patterns **(Encapsulation)**
  + Code Reuse patterns **(Inheritance)**

1. Functional Style

* This implies that JavaScript uses a functional approach, even objects can be created using constructor functions and each constructor function represents a unique object type.
* Also, functions in JavaScript can be used as objects and can be passed to other functions too.
* Many important JavaScript concepts and features like callbacks, closures, etc. are implementations of functions only.

1. Platform Independent

* This implies that JavaScript is platform-independent or we can say it is portable.
* This simply means that you can simply write the script once and run it anywhere and anytime.
* In general, you can write your JavaScript applications and run them on any platform or any browser without affecting the output of the Script.

1. Prototype-based Language

* JavaScript is a prototype-based scripting Language.
* This means JavaScript uses prototypes instead of classes or inheritance.
* In languages like Java, we create a class and then we create objects for those classes.
* But in JavaScript, we define an object prototype, and then more objects can be created using this object prototype.

7. Interpreted Language

* JavaScript is an interpreted language which means the script written inside JavaScript is processed line by line.
* The JS code is interpreted by JavaScript interpreter which is a built-in component of the Web browser.
* But these days many JavaScript engines in browsers like the *V8 engine* in Chrome use just-in-time compilation for JavaScript code.

## 8. Single threaded

* JavaScript doesn't support multi-threading, by default it is single-threaded, which means it can execute only a single task at a time.
* But JavaScript provides some features using which you can implement parallel execution. They are:
  + Async processing
  + Web workers

9. Async Processing

* JavaScript supports **Promise** which enables asynchronous requests wherein a request is initiated and JavaScript doesn't have to wait for the response if a request takes time and may block the request processing.
* Also starting from ES8, Async functions are also supported in JavaScript, these functions don't execute one by one, rather they are processed parallelly which has a positive effect on the processing time, reducing it to a great extent.

10. Web Workers

* Using Web workers you can run processes in background threads so that you can have parallel execution.
* If you have any task that performs some heavy-duty work, then you can use a Web worker to run it in the background.

11. Client-side Validations

* This is a feature that has been available in JavaScript since the beginning and is still widely used because every website has a form in which users enter values, and to make sure that users enter the correct value, we must put proper validations in place, both on the client side and on the server-side.
* JavaScript is used for implementing client-side validations.

12. More control in Browser

JavaScript being a client-side language provides many features that help developers to divide processing between browser and server hence reducing the load on servers by having basic processing, validations, temporary data saving using cookies, etc. on the browser itself.

Along with all these features, JavaScript provides the following useful features too:

* Detecting browser type, name OS version, etc information of the client for analysis.
* Extensive in-built library with many useful functions for validation, data type conversion, string operations, etc.
* It is an object-centered language with Window Object being the most important object in JavaScript and it also supports Polymorphism.
* Support for commonly used complex data types like arrays, Maps, lists, etc. with support of in-built functions to operate on them.

**Main Advantages of Javascript**

**Simple:-** Javascript is simple to comprehend & pick up. Both users and developers will find the structure to be straight forward. Aditionally it is very doable to implement, saving web developers atone of money when creating dynamic content.

**Speed:-** JavaScript is a "interpreted" language, it cuts down on the time needed for compilation in other programming languages like [**Java**](https://www.tutorialspoint.com/java/index.htm). Another client-side script is JavaScript, which accelerates programme execution by eliminating the wait time for server connections.

**Saves time and bandwidth:-**  Regardless of where you host JavaScript, it always gets executed on the client environment to save lots of bandwidth and make the execution process fast.

**Easily send HTTP requests:-** In JavaScript, XMLHttpRequest is an important object that was designed by Microsoft. The object calls made by XMLHttpRequest as an asynchronous HTTP request to the server to transfer the data to both sides without reloading the page

**Compatible for all browsers:** The biggest advantage of JavaScript having the ability to support all modern browsers and produce an equivalent result.

**Vastly used:-** JavaScript is employed everywhere on the web.

**Environment Support:-** JavaScript plays nicely with other languages and may be utilized in an enormous sort of application.

**Programming language:**There are many available courses within the field of JavaScript, because of which you’ll quickly and simply expand your knowledge of this programming language.

**Easy to use:**It is not difficult to start working in JavaScript. For this reason, many of us prefer to start our adventure in the IT sector by learning this language. It gives the power to make rich interfaces.

**Backend usage**: There are some ways to use JavaScript through Node.js servers. It is possible to develop a whole JavaScript app from front to back using only JavaScript.

**Uses of Javascript**

* 1. Web Applications:

As day-by-day there is a continuous improvement in the browsers, so JavaScript gained popularity for making robust web applications. We can understand it by taking the example of **Google Maps**. In Maps user just requires to click and drag the mouse; the details are visible just by a click. There is a use of JavaScript behind these concepts.

* 1. Web development:

JavaScript is commonly used for creating web pages. It allows us to add dynamic behavior to the webpage and add special effects to the webpage. On websites, it is mainly used for validation purposes. JavaScript helps us to execute complex actions and also enables the interaction of websites with visitors. Using JavaScript, it is also possible to load the content in a document without reloading the webpage.

* 1. Mobile Applications:

Using JavaScript, we can also build an application for non-web contexts. The features and uses of JavaScript make it a powerful tool for creating mobile applications. The **React Native** is the widely used JavaScript framework for creating mobile applications. Using **React Native,** we can build mobile applications for different operating systems. We do not require writing different codes for the iOS and Android operating systems. We only need to write it once and run it on different platforms.

* 1. Games

JavaScript is also used for creating games. It has various libraries and frameworks for creating a game. The game can either be a 2D or 3D. Some JavaScript game engines such as **PhysicsJS, Pixi.js** help us to create a web game. We can also use the **WebGL**(web graphics library), which is the JavaScript API to render 2D and 3D images on browsers.

* 1. Presentations

JavaScript also helps us to create presentations as a website. The libraries, such as RevealJs, and BespokeJs, can be used to create a web-based slide deck. They are easier to use, so we can easily make something amazing in a short time.

* The Reveal.js is used to create interactive and beautiful slide decks with the help of HTML. These presentations work great with mobile devices and tablets. It also supports all of the CSS color formats.
* The BespokeJS includes animated bullet lists, responsive scaling, and a wide variety of features.

6.Server Applications:

A large number of web applications have a server-side to them. JavaScript is used to generate content and handle [HTTP](https://www.javatpoint.com/http) requests. JavaScript can also run on servers through Node.js. The [Node.js](https://www.javatpoint.com/nodejs-tutorial) provides an environment containing the necessary tools required for JavaScript to run on servers.

* 1. Web servers:

A web server can be created by using Node.js. Node.js is event-driven and not waits for the response of the previous call. The servers created using Node.js are fast and don't use buffering and transfer chunks of data. The HTTP module can be used to create the server by using the createServer() method. This method executes when someone tries to access the port 8080. As a response, the [HTTP](https://www.javatpoint.com/computer-network-http) server should display HTML and should be included in the [HTTP](https://www.javatpoint.com/http-tutorial) header.

**Qn:** Difference between var, let & const in Javascript?

**Var:**

* The **[var](https://www.geeksforgeeks.org/javascript-var/" \t "_blank)**is the oldest keyword to declare a variable in [JavaScript](https://www.geeksforgeeks.org/introduction-to-javascript/).
* The **var** statement is used to declare a variable in JavaScript. A variable declared with the **var keyword** is defined throughout the program.
* It can be declared globally and can be accessed globally.

**Let**

* The **let**statement is used to declare a local variable in TypeScript.
* It is similar to the var keyword, but it has some restriction in scoping in comparison of the var keyword.
* It can be declared globally but cannot be accessed globally.
* The let keyword was introduced in [ES6 (2015)](https://www.w3schools.com/js/js_es6.asp)
* Variables declared with let have Block Scope
* Variables declared with let must be Declared before use
* Variables declared with let cannot be Redeclared in the same scope

**Const**

* The const keyword was introduced in [ES6 (2015)](https://www.w3schools.com/js/js_es6.asp)
* Variables defined with const cannot be Redeclared
* Variables defined with const cannot be Reassigned
* Variables defined with const have Block Scope

**Popular Js Libraries**

* **ReactJs:**

React (also known as React.js or ReactJS) is a [free and open-source](https://en.wikipedia.org/wiki/Free_and_open-source_software) [front-end](https://en.wikipedia.org/wiki/Frontend_and_backend) [JavaScript library](https://en.wikipedia.org/wiki/JavaScript_library) for building [user interfaces](https://en.wikipedia.org/wiki/User_interface) based on [components](https://en.wikipedia.org/wiki/Component-based_software_engineering). It is maintained by [Meta](https://en.wikipedia.org/wiki/Meta_Platforms) (formerly Facebook) and a community of individual developers and companies.

React can be used to develop [single-page](https://en.wikipedia.org/wiki/Single-page_application), mobile, or [server-rendered](https://en.wikipedia.org/wiki/Server-side_rendering) applications with frameworks like [Next.js](https://en.wikipedia.org/wiki/Next.js). Because React is only concerned with the user interface and rendering components to the [DOM](https://en.wikipedia.org/wiki/Document_Object_Model), React applications often rely on [libraries](https://en.wikipedia.org/wiki/JavaScript_libraries) for routing and other client-side functionality.

* **Angular:**

Angular is a TypeScript-based free and open-source single-page web application framework run on Node.js. Angular is one of the most powerful and open-source JavaScript frameworks. Google operates this framework and is designed to use it to develop a Single Page Application (SPA). This development framework is known primarily because it gives developers the best conditions to combine JavaScript with HTML and [CSS.](https://www.geeksforgeeks.org/css-tutorial/)

* **Vue.js**

Vue is an open-source JavaScript framework for creating a creative UI. It is often regarded as one of the best JavaScript frameworks. The integration with Vue in projects using other JavaScript libraries is simplified because it is designed to be adaptable. Vue JS is one of the ideal JavaScript frameworks that allows you to develop cross-platform apps and websites and supports the development of SPAs or single page apps.

* **Node.js**

Node.js is a cross-platform, open-source JavaScript runtime environment that can run on Windows, Linux, Unix, macOS, and more. Node.js runs on the V8 JavaScript engine, and executes JavaScript code outside a web browser. Node.js lets developers use JavaScript to write command line tools and for server-side scripting.

* **Meteor**

Meteor, or MeteorJS, is a partly proprietary, mostly [free and open-source](https://en.wikipedia.org/wiki/Free_and_open-source_software) [isomorphic](https://en.wikipedia.org/wiki/Isomorphic_JavaScript) [JavaScript](https://en.wikipedia.org/wiki/JavaScript) [web framework](https://en.wikipedia.org/wiki/Web_framework)written using [Node.js](https://en.wikipedia.org/wiki/Node.js). Meteor allows for rapid prototyping and produces cross-platform ([Android](https://en.wikipedia.org/wiki/Android_OS), [iOS](https://en.wikipedia.org/wiki/IOS" \o "IOS), [Web](https://en.wikipedia.org/wiki/World_Wide_Web)) code. The server-side [MongoDB](https://en.wikipedia.org/wiki/MongoDB" \o "MongoDB) program is the only proprietary component of Meteor and is part of the Meteor download bundle. It is possible to use Meteor without using the server-side MongoDB.

* **Auerlia**

Aurelia is a **front-end, open-source framework** developed by Rob Eisenberg. Aurelia 1.0 was released for the first time in 2016. Aurelia consists of function-oriented modules, such as plugins, routing, testing, dependency injection, and many more. You can use the Aurelia JS framework to develop web, mobile, and desktop applications.

* **Polymer**

The polymer is an open-source JavaScript library maintained by Google to build web applications using web components. It also supports both one-way and two-way data binding, thus creating a more extensive area of application.

* **Backbone.js**

It’s one of the most popular JavaScript frameworks. It can be used to create a single-page application. The development of this framework involves the idea that all server-side functions must flow through an API, which would help to achieve complex functionality by writing less code.

* **Ember.js**

Ember.js is an open-source JavaScript framework that was originally released by **Yehuda Katz in 2011**. It was initially known as SproutCore 2.0 before it became known as Ember.js. Work on the Ember framework started in 2011 and version 1.0 was released two years later. Instead of one company supporting the project, Ember has many supporters who are usually users of the framework, such as LinkedIn and Yahoo.