How does JavaScript work on a web browser?

JavaScript is a client-side scripting language and one of the most efficient, commonly used scripting languages. The term .**client-side scripting language means** that it runs at the client-side( or on the client machine) inside the web-browsers, but one important thing to remember is that client's web-browser also needs to support the JavaScript or it must be JavaScript enabled. Nowadays, most of the modern [web browsers](https://www.javatpoint.com/browsers) support JavaScript and have their JavaScript engines. For example, Google Chrome has its own [JavaScript](https://www.javatpoint.com/javascript-tutorial) engine called V8.

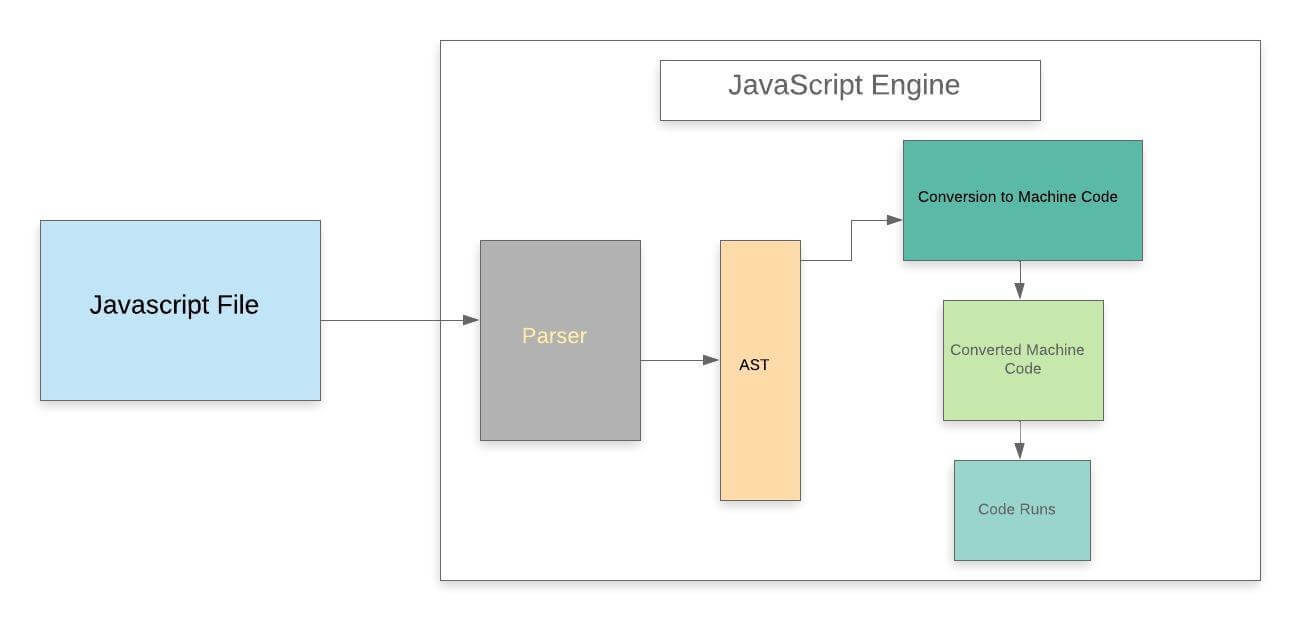
**Some other web-browsers with their JavaScript engines**

|  |  |  |
| --- | --- | --- |
|  | **Web Browser** | **JavaScript engines** |
| 1. | Edge | Chakra |
| 2. | Safari | JavaScript Core |
| 3. | Firefox | Spidermonkey |

It totally depends on the web-developers how they use JavaScript and for what, because it can be used for several things in web development. One of the most common uses of JavaScript is to validate data given by the user in the form fields.

In this case, we have used a [chrome browser](https://www.javatpoint.com/google-chrome) to run our program that has the "V8" JavaScript engine, which is also used for creating the [Node.js](https://www.javatpoint.com/nodejs-tutorial). As we already know, JavaScript is an interpreted language that means it gets executed in line by line manner (or which means the JavaScript engine converts the Js code line by line and runs in the same manner instead of converting the whole program once).

We can understand how a typical JavaScript engine works with help of a diagram:



Whenever we run a JavaScript program inside a web browser, JavaScript code is received by the browser's engine and the engine runs the source code to obtain the output.

In a standard JavaScript engine, the source code goes through several steps and gets executed as you can see in the above given diagram.

Let us understand each of these steps in more detail.

**Step 1: Parser**

This is the first stage of the engine, every time we run a JavaScript program, our code is first received by the "parser" inside the JS engine. The parser's job is to check the JavaScript code for syntactic errors in line by line manner because JavaScript is an interpretive scripting language, so whenever an error is detected by the parser, it throws a kind of error and stops execution of the code.

In short, we can say that it parses JavaScript code.

**Step 2: AST**

Once the parser checks all JavaScript codes and gets satisfied that there are no mistakes/errors in the code, it creates the data structure called AST (it stands for Abstract Syntax Tree).

We can easily understand what is AST with help of following example.

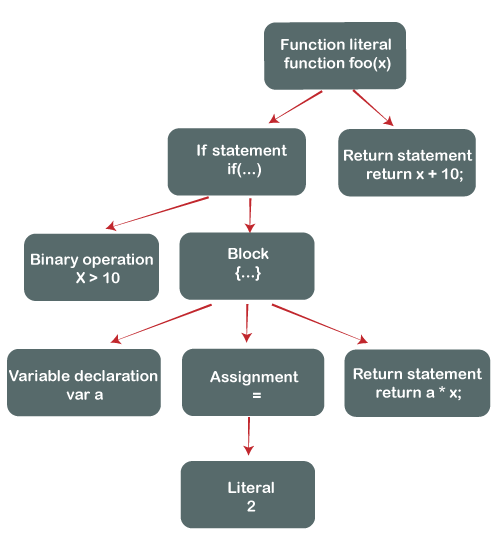
Example: Let's suppose we have a JavaScript program as given below:

**Program**

1. function fun(x) {
2. if (x **>** 15) {
3. var a = 4;
4. return a \* x;
5. }
7. return x + 10;
8. }

Once the parser checks the above JavaScript code, it will create a data structure called AST as we have already discussed above. The created **AST** (Abstract Syntax Tree) looks like the given image

Once the parser checks the above JavaScript code, it will create a data structure called AST as we have already discussed above. The created **AST** (Abstract Syntax Tree) looks like the given image.



Note: It is not the exact abstract syntax tree, but it is the pictorial representation of the Abstract Syntax Tree.

**Step 3: Conversion to Machine code**

Once the Abstract Syntax Tree is created by the parser, the JavaScript engine converts the JavaScript code into the machine code (or in the language that machine can understand).

**Step 4: Machine code**

When the program written in the JavaScript gets converted in the machine language (or in byte code), the converted code is sent to the system for execution, and finally, that byte code run by the system/engine just like we observe in our first example.