

As a client/server model, the browser is the client run on a computer that contacts the Web server and requests information. The Web server sends the information back to the Web browser which displays the results on the computer or other Internet-enabled device that supports a browser.

## JavaScript Engine

The most famous one is the V8 Engine developed by Google which is used in Chrome and Node.js, but there are many more JavaScript engines.

How does the JS engine work?

### 1. Parser

The HTML parser encounters a script tag with a source. The source code inside this script gets loaded as a UTF-16-byte stream to a byte stream decoder. This byte stream decoder then decodes the bytes into token which are sent to the parser.

**2. AST :** The parser creates nodes based on the tokens it receives. With these nodes, it creates an Abstract Syntax Tree (AST).

### 3. Interpreter

Next up is the interpreter, it walks through the AST and generates byte code. It reads the code line by line. When the byte code has been generated, the AST is deleted, clearing up memory space.

The problem with interpreters is that running the same code multiple times can get slow, that is why we have the compiler which does not repeat loops and is more optimized.

**4. Profiler:** The Profiler monitors and watches code to optimize it.

5. Compiler : The compiler works ahead of time and creates a translation of the code that has been written and compiles down to a lower-level language that machines can read.

**6. Optimized code :** Why do you need to know these things? So, we can optimize the code to be run quicker.

## Call stack and Memory Heap

The call stack keeps track of where we are in the code. It uses first in and last out and stacks for example functions. The memory heap is a place to store and write information, where we allocate, use, and remove the memory. The call stack calls a function from the memory heap and after executing removes it from the stack. When the maximum call stack has been reached, e.g., with an infinite loop, we call it a stack overflow.

JS automatically frees up memory that no longer is used. It marks and sweeps it from the memory. For example, when a value gets reassigned, and the original value is not used anymore. We call this garbage collection.