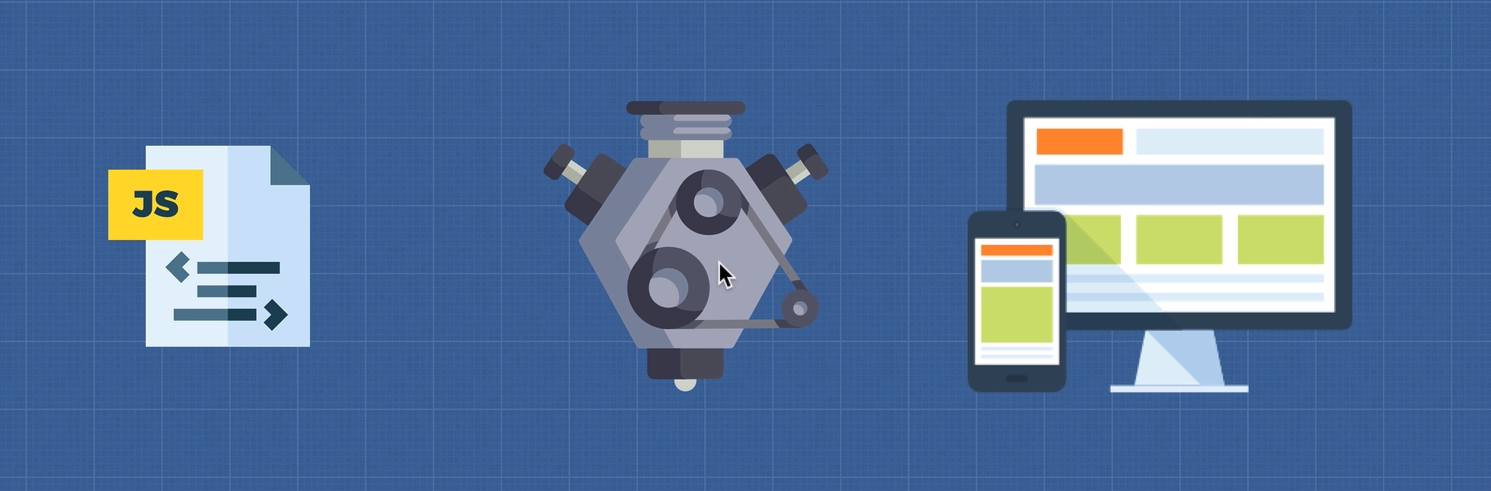
JavaScript: Advanced Concepts

# JavaScript Foundation

## JavaScript Engine

1. JS is a single threaded language.
2. It is an interpreted language.

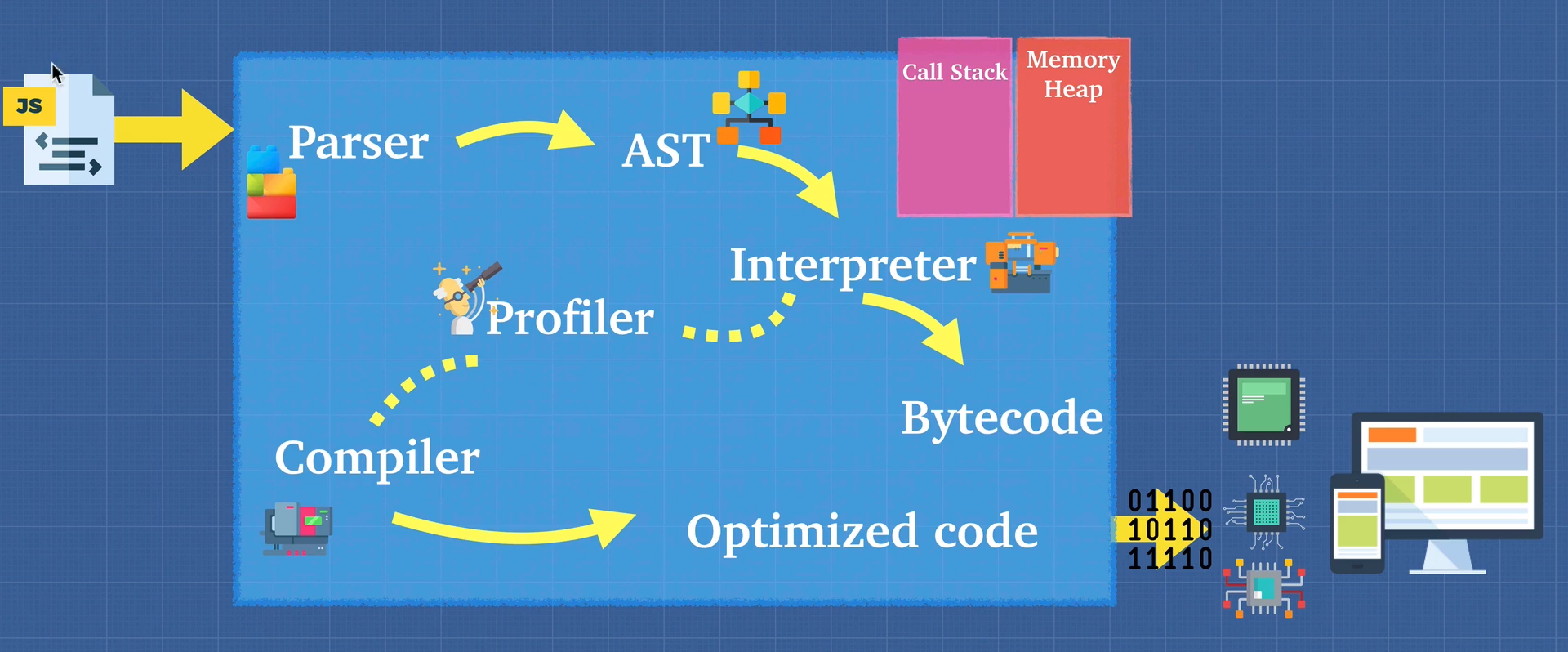
A JS Engine helps the computer understand a JS file by translating it into machine code.



There are various JS engines available, example, V8, Chakra, Chakran, etc.

Google’s V8 enables JS to run fast. It was released in 2008.

### How JS Engine works?

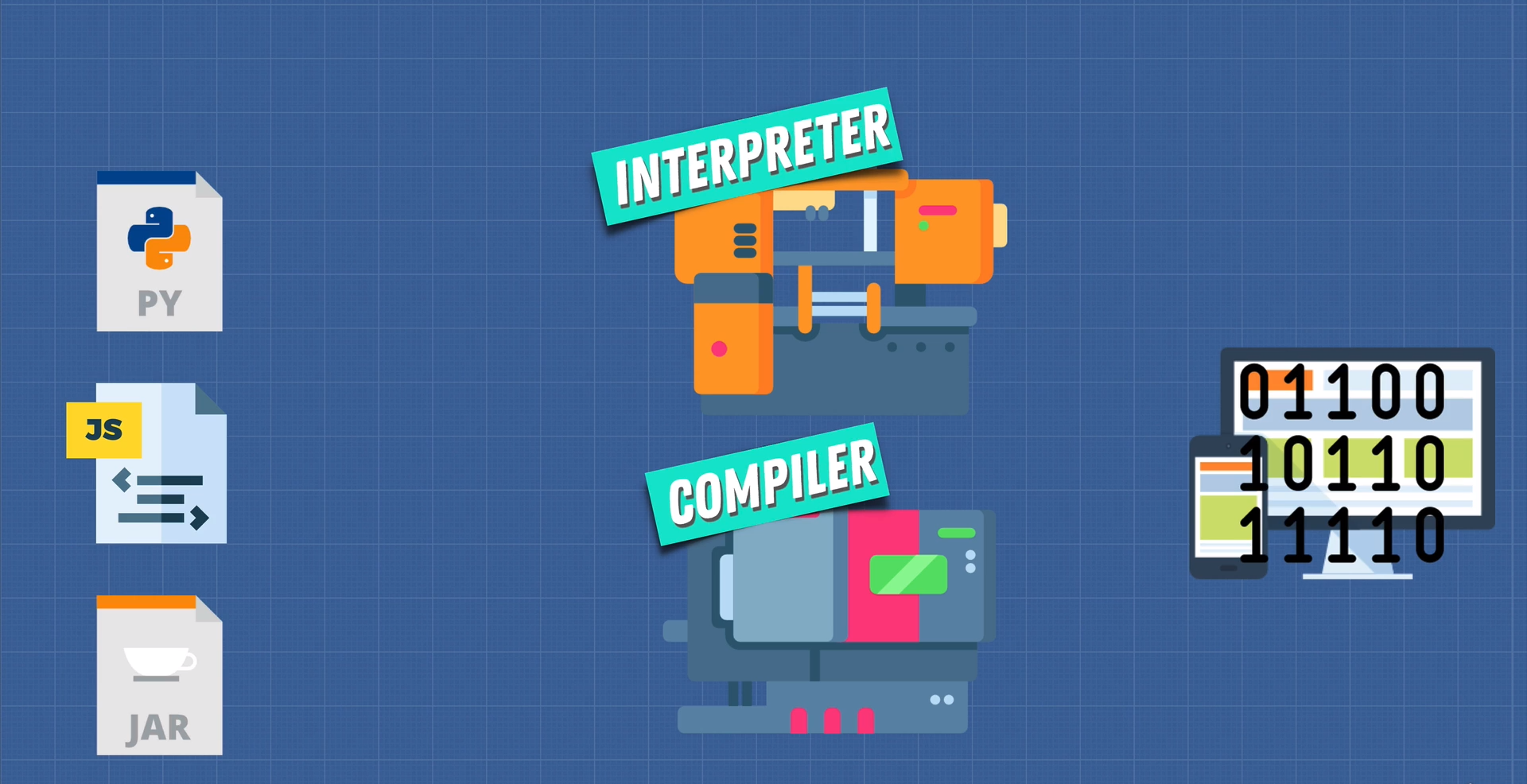


1. The first step is to do a **lexical analysis** of the JS code. This is done by a **parser** which tokenize the code.
2. Then an **AST (Abstract Syntax Tree)** is formed from the tokens.
3. Then an interpreter spits out byte code. This process is called **ignition**.
4. At the same time, a **Profiler** checks for places where the code can be optimized. Wherever, it finds such a code it passes it to the compiler to optimize that code. Example, a for loop.
5. Now the compiler replaces the optimized code in the byte code. The output, byte code + optimized code = machine code that is dumped out.

***JavaScript can thus be an interpreted language or compiled language or both depending on the engine.***

**NOTE: A JS Engine should conform to ECMAScript standard.** [List of ECMAScript engines - Wikipedia](https://en.wikipedia.org/wiki/List_of_ECMAScript_engines)

## Interpreter & Compiler



### Interpreter

* Reads file line by line

### Compiler

* Works ahead of time.
* Compiles down at beginning into machine code.
* However, at the end, the machine code is interpreted by the CPU

### Pros & Cons

* Interpreter starts up quickly and is suited for JS as the browsers need to work fast.
* Startup is slow in a compiler but it runs faster at runtime because the entire code is in machine code, i.e., compiler does optimization.

Taking the pros of both interpreter and compiler, a new compiler was developed called the **JIT Compiler**. V8 uses JIT Compiler.

## Babel & TypeScript

Have you heard of Babel or TypeScript? They are heavily used in the Javascript ecosystem and you should now have a good idea of what they are:  
  
[Babel](https://babeljs.io/) is a JavaScript compiler that takes your modern JS code and returns  browser compatible JS (older JS code).  
[Typescript](https://www.typescriptlang.org/) is a superset of JavaScript that compiles down to Javascript.  
  
Both of these do exactly what compilers do: Take one language and convert into a different one!

## WebAssembly

It is a standard for compiling the JS code for browsers. This can remove the need for a JS Engine as there will be a standard just like .exe files.