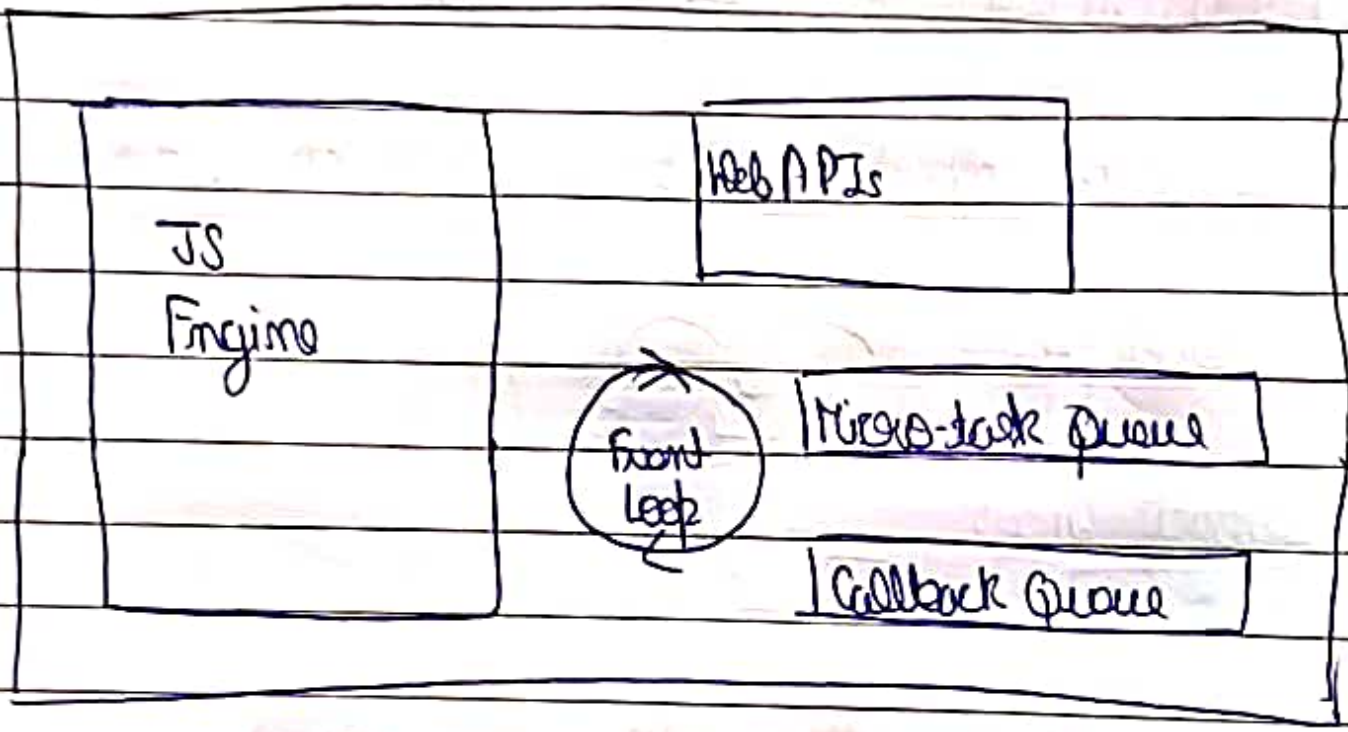
. github : divyaranjan10

DATE

CHAP 16: JS Engine exposed, Google's V8 Architecture

Javascript Runtime Environment



* JRE is not possible without JS engine, therefore JS engine is the heart of JRE *

* Browser can easily execute JS code because it has JRE *

* Whenever you run a JS code, you need a container which has everything required to run it, it is called JRE.

DATE

- Popular JS engine:

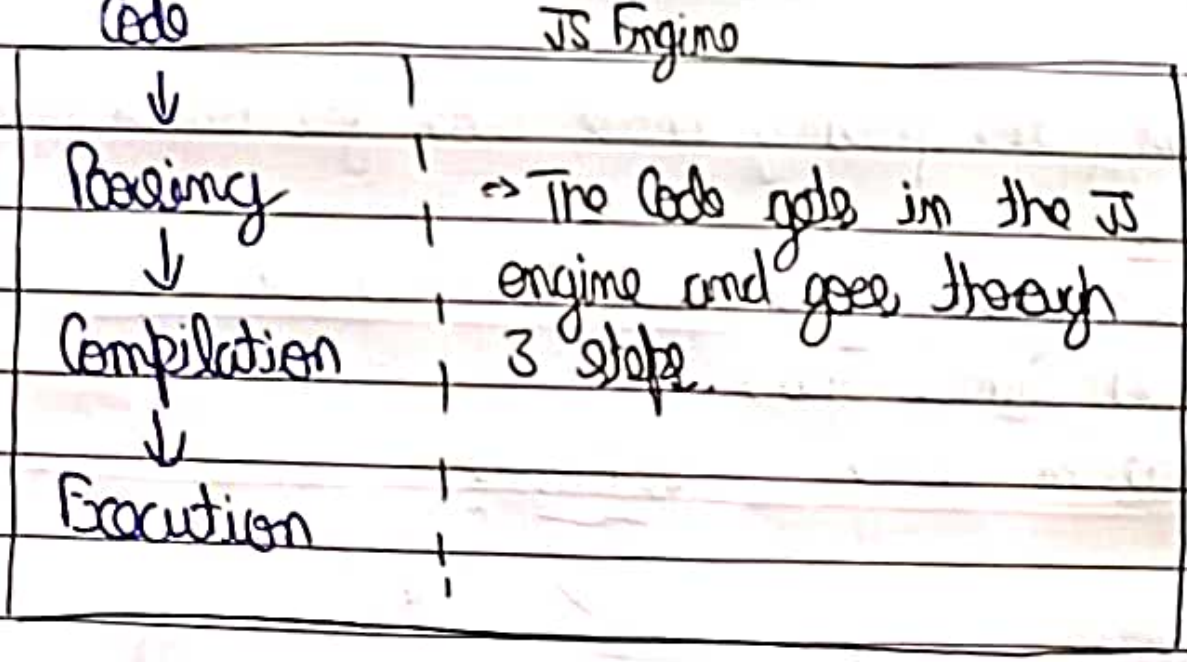
=> V8 : Chrome

=> Chakra : Microsoft Edge

=> Spider Monkey : Firefox

" JS engine is not a machine, but a program"

- Architecture of JS Engine:



DATE

- **Parsing Phase:**

Code

↓

Parsing

↓ AST

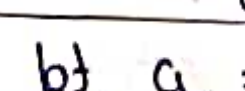
Compilation

↓

Execution

JS Engine

Tokens



Syntax Parser

=> During the parsing phase, code gets broken into tokens.

=> Syntax parser takes code and convert into AST (Abstract Syntax Tree)

- **Compilation Phase:**

Parsing

↓ AST

Compilation

↓

Execution

JS Engine

JIT Compiler

Interpreter + Compiler

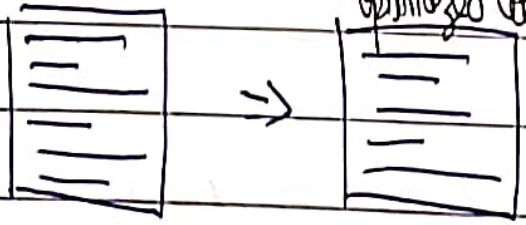
- It uses both interpreter and compiler to run the code.

DATE_____

• Interpreter vs Compiler :

⇒ Interpreter : In this, the code is executed line by line.

⇒ Compiler : In this, the whole code is compiled before executing and a new code is formed that is optimized form of code. [Optimized - Runs very fast and has lot of performance improvement].

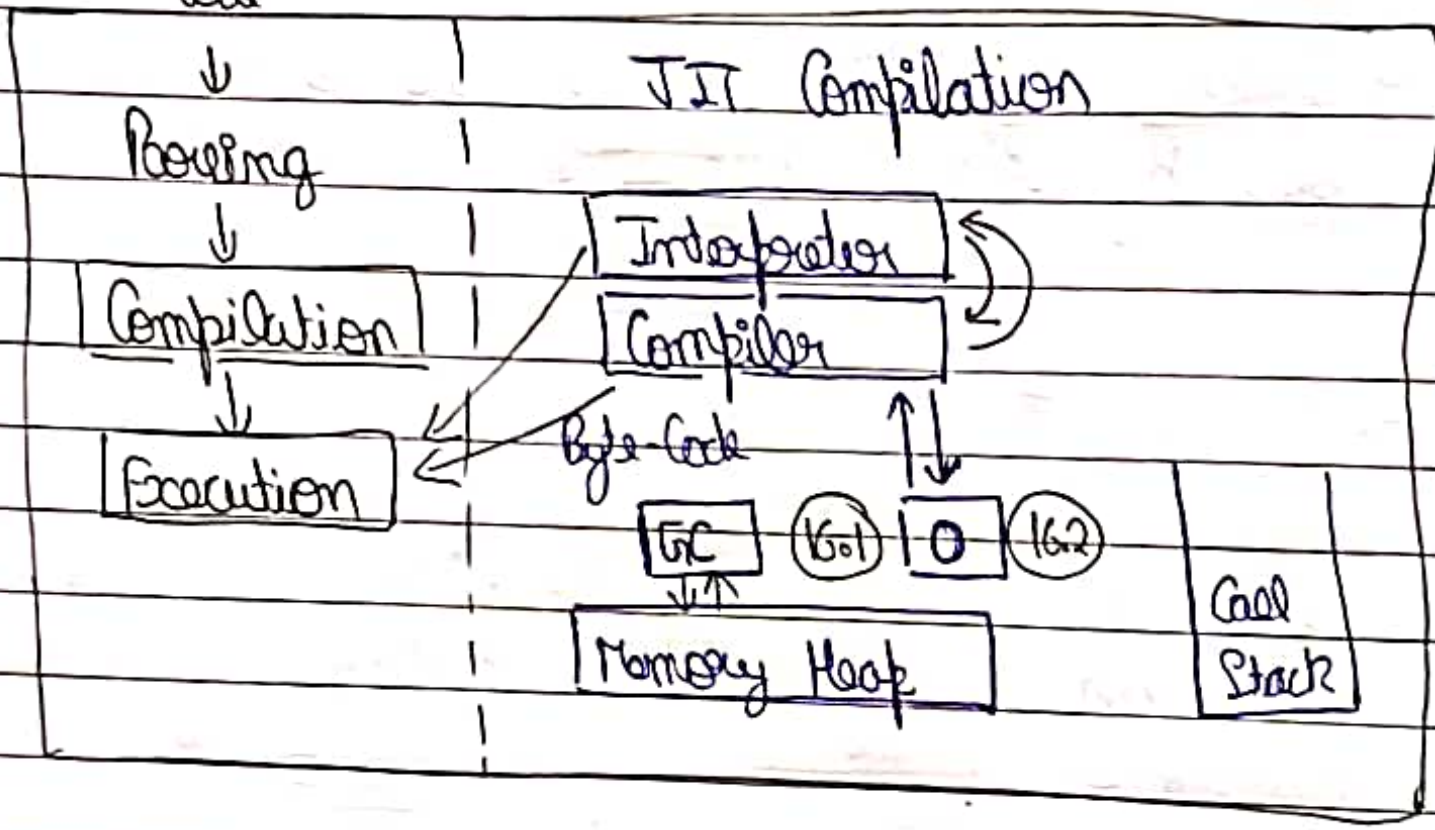


* Interpreter is fast vs. Compiler is efficient

* JS can behave both as an interpreter as well as an compiler language depending on the JS engine.

DATE

- Compilation and Execution Phase go hand-in-hand



- So, compilation and execution go hand-in-hand.
- After parsing we get AST, which then interpreted and converted into byte-code and it moves to the execution phase.
- Also, while it is being interpreted, the interpreter and compiler work together and the compiler optimizes the code on the run time and the compiler produces the byte code and send it for execution.

DATE

- Execution phase can not be completed without Memory Heap and Call Stack.
- Memory Heap : this is the place where all the memory is stored, it is continously in sync with call stack and Garbage collector.

- Garbage Collector : (16.1)
⇒ It collects all the garbage like unused variables etc.
⇒ It uses Mark & Sweep Algorithm.

(16.2) There are many forms of optimizations which a compiler does for the code like :
(a) Inlining (b) Copy Elision (c) Inline Caching

- Fastest JS Engine : V8 Engine, it has Interpreter named as 'Ignition'.
⇒ Compiler named as 'Turbo Fan'.

DATE

V8 JS Engine