Episode-08 → Deep Dive into V8 JS Engine

V8 JS Engine → Written in C++, developed and maintained by Google. It is a Javascript engine. It passes and executes the code.

#Several Stages when code is given to V8 Engine are :-

① Parsing Stage

Step (i) Lexical Analysis → The code is broken down into multiple tokens
(Tokenization) → Also known as tokenisation.

code → Tokens

Step (ii) Syntax Analysis → The tokens are then converted into Abstract Syntax Tree (AST)
(Parsing)

Tokens → AST { Abstract syntax Tree }

Note: visit astexplorer.net

Why Syntax error? → When V8 engine cannot generate AST, it throws syntax error

---

2 Types of Languages →

Interpreted	Compiled
→ executed line by line	→ executes whole code at once first compilation, then HL code → ML code
→ Fast Initial Execution	→ Initially Heavy but executed fast
→ Interpreter	→ Compiler
→ Ex: Python	→ Ex: C++

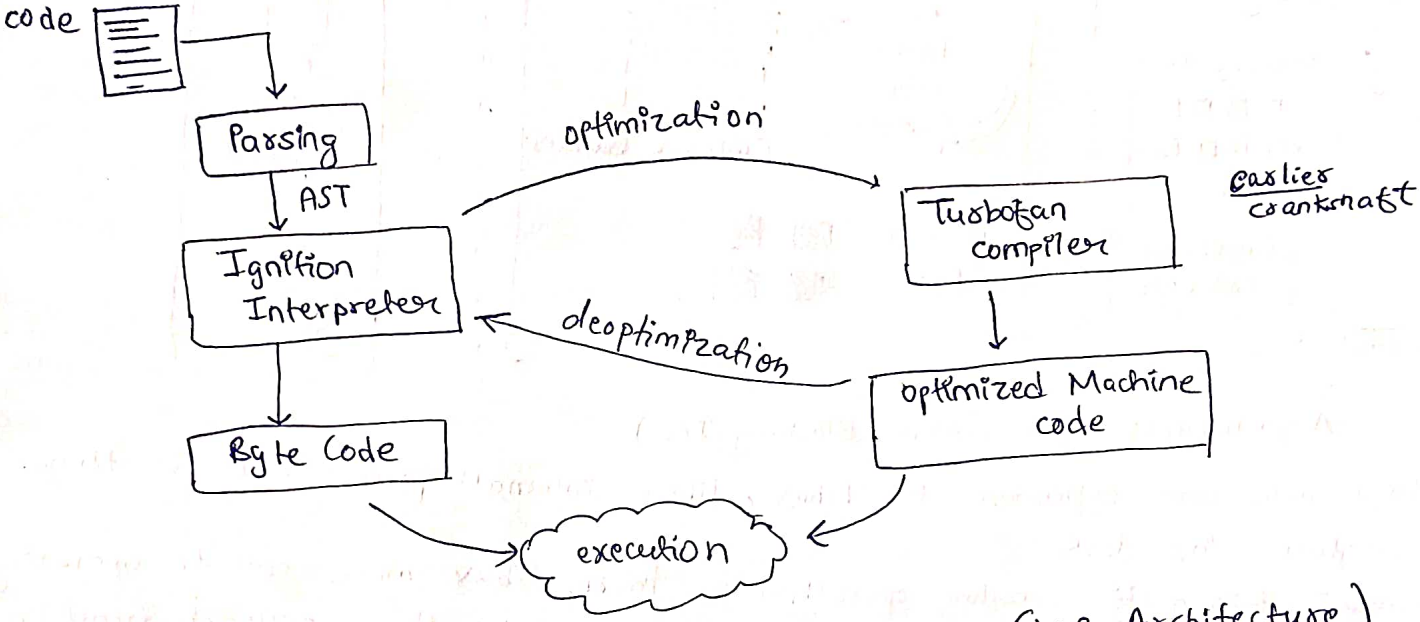
Which Type is JavaScript then? → Both, It has both interpreter And Compiler Hence, Javascript is called Just in Time (JIT) compilation Type language.

---

② Interpreter Stage

- Google call its interpreter as ignition interpreter and its compiler as Turbofan compiler.
- The AST is given to ignition interpreter which converts the AST to Byte Code that is finall given for execution.
- Any code that is used again and again (called as HOT CODE) is given to Turbo fan compiler for optimization so that the next time the same code runs, it is executed faster.
- Turbofan converts the HOT code into optimized machine code and gives it for execution.

Problem → Turbofan makes assumptions. Ex: in case of sum(10,5) function which is used again, the Turbofan will assume ~~Turbofan~~ compiler sum will always accept integers. Hence when string value will come on sum function, then it will give code back to ignition interpreter. This is called deoptimisation. So Turbofan deoptimises the code. Then ignition interpreter will call the sum function with strings and executes the code. when code is deoptimised by Turbofan.



Just in Time Compilation Diagram (V8 Architecture)