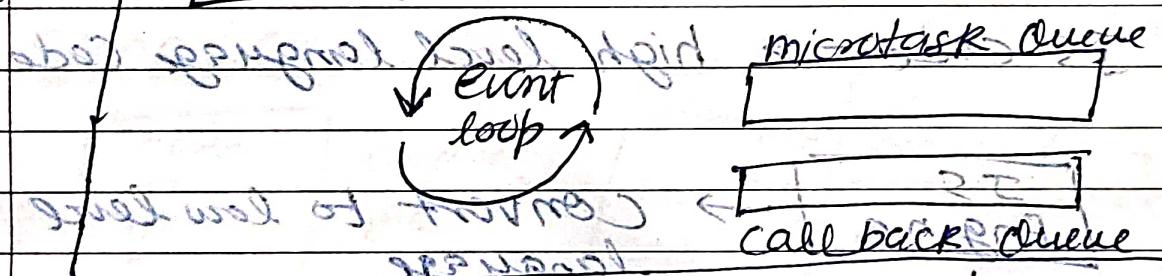
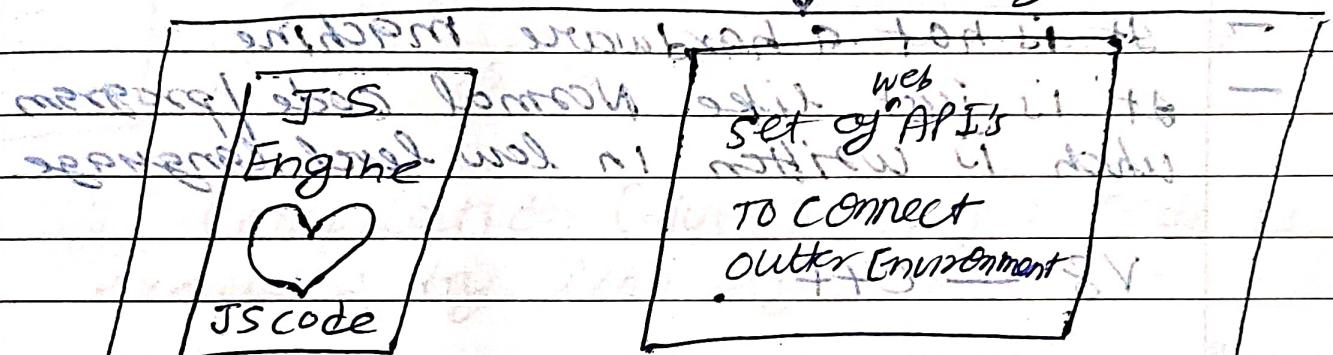


18 Video

JS Engine EXPOSED Google's V8 Architecture

- JS can be executed in a browser, server, smart-watch, robot and smart bulb.
- It is possible because of JS Runtime Environment
- JS runtime environment has all the things required to run the JS code.
- JS runtime environment is a big container.



JS Runtime Environment

- Every Browser is having JS Runtime Environment.
- Node.js - It is an open source JS Runtime Environment.
- It can Run JS Code outside the browser.

- set of APIs may differ in different RTE [Node.js & Browsers may differ in API's]
- setTimeout APIs of Node.js and Browsers may differ in their implementation.

* List of JS Engine

chakra - IE 11 & Edge

spidermonkey - Firefox

v8 - Chrome, Node.js, Deno

→ one can also generate a JS engine.

→ must follow ECMA Script standard

→ V8 like governing body for JS language

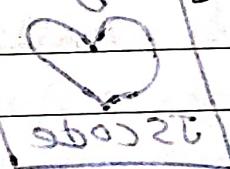
→ SpiderMonkey - Inventing smart 2T

→ very fast JS Engine

→ written in C/C++

- It is not a hardware machine
- It is just like Normal code/program which is written in low level language

V8 C/C++



JS code → high level language code

JS Engine → Convert to low level language.

Device → Any machine.

Code

passing

AST

compilation

Execution

Fig: JS Engine - IE -

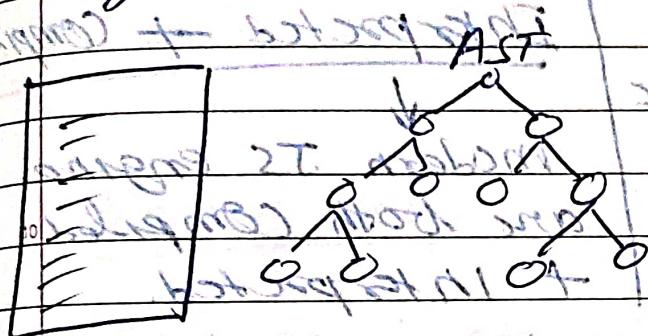
IE -

Parsing (parser) is an algorithm. 2T *

It breaks human readable JS code (high level) into smaller unit called Tokens in

It generates a Abstract Syntax Tree (AST)

Syntax Parser



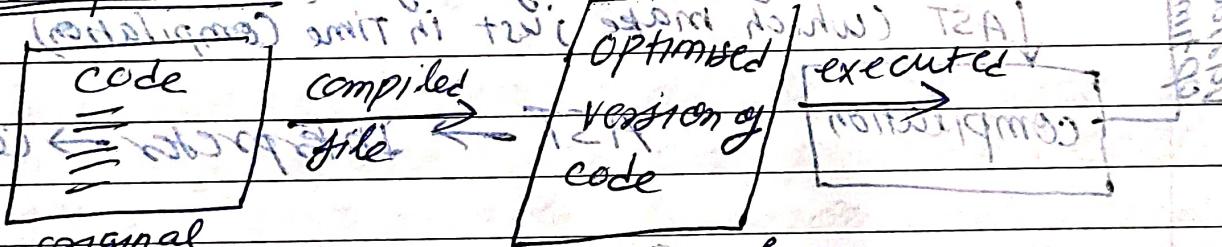
by JS AST - 2T

and then it's passed to

compilation

JIT Compilation (just-in-time) ! Code is executed by line by line

Compiler



original code is modified & executes very fast

Interpreter vs Compiler

→ Code executes very

→ more efficiency.

fast. It does not

have to wait for compilation of code

Immediately starts to go to execution. Wait till no errors till no

high speed. TOA is more when it's wrong if

it finds also other errors along with. (intelligence)

No waiting for total errors. ex: 1st or 2nd error

* JS language can be compiler language (and) as well as interpreted language. It is completely depend on JS Engine.

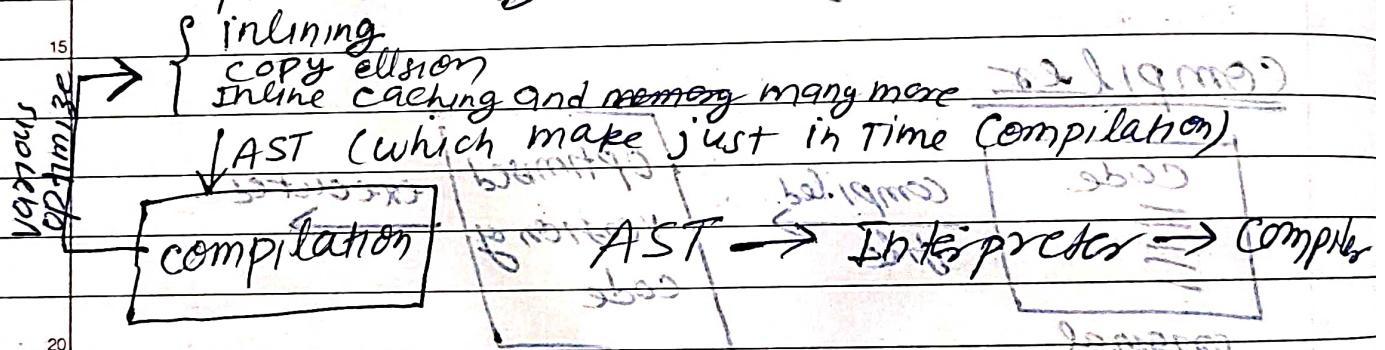
Initially

JS - Interpreted as browser don't have to wait.

Now Days

Interpreted → Compiled
Modern JS engine are both compiled + interpreted

JIT: Compilation → JS Engine can use compiler along with interpreter.



→ convert high level code to low level code.

but taking more time

↓ (byte code)

→ memory room

Execution

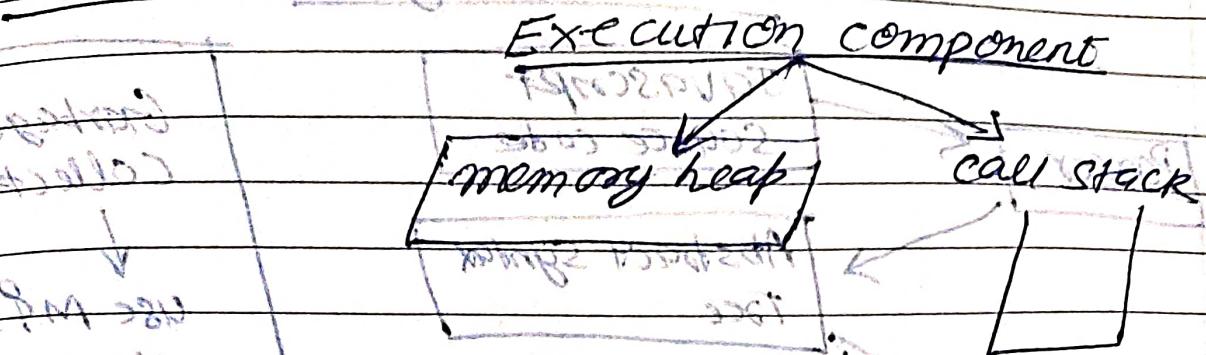
time

→ Interpreter takes help of compiler to optimize the process.

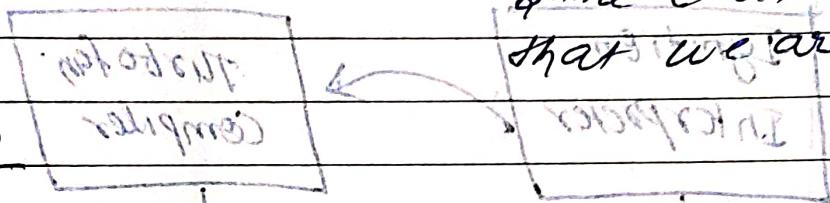
→ Compiler - It's just optimizes the process as much as it can on the run time. (Just In-time compilation)

→ In series js engine there is AOT (ahead of time compilation). here compiler takes the code which is supposed to be executed later and optimize as much as it can and generate a byte code which is passed for execution.

Execution



Memory Heap



- All memory is stored here
- At constantly in sync with call stack, garbage all the variables collector & lot of other thing & function →
- memory is assigned from here only

* Garbage Collection tries to free up memory space whenever possible. collects all the undesirable function and variable and sweeps it.

uses mark & sweep Algorithm] H.W
 ↳ widely used algo. across the garbage collectors

Compiler optimizes by various function operations

- ① Inlining
 - ② copy division] H.W
 - ③ inline caching
- & many more

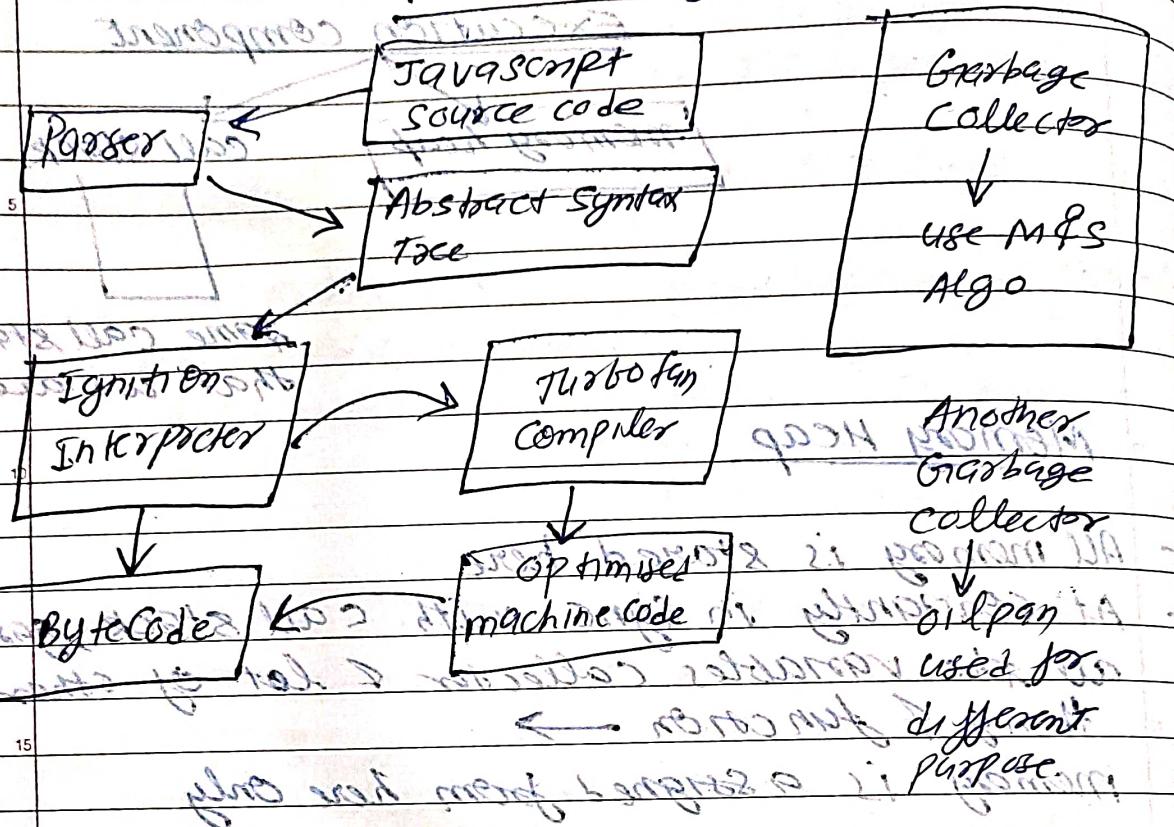
V8 - fastest JS Engine right now

↪ Ignition Interpreter

→ compiler for optimizing compiler for optimization.

19 Video

#

V8 JS Enginethe way of writing JS language

~~Procedural
programming
language~~

~~functional
programming
language~~

~~object oriented
programming
language~~

Conca

Start
End

with [parallel ①
asynchronous ②
event driven ③
task based ④]

writing code in this - 8V

in this video we discuss various ways of writing