

Episode-3-Let's Write Code

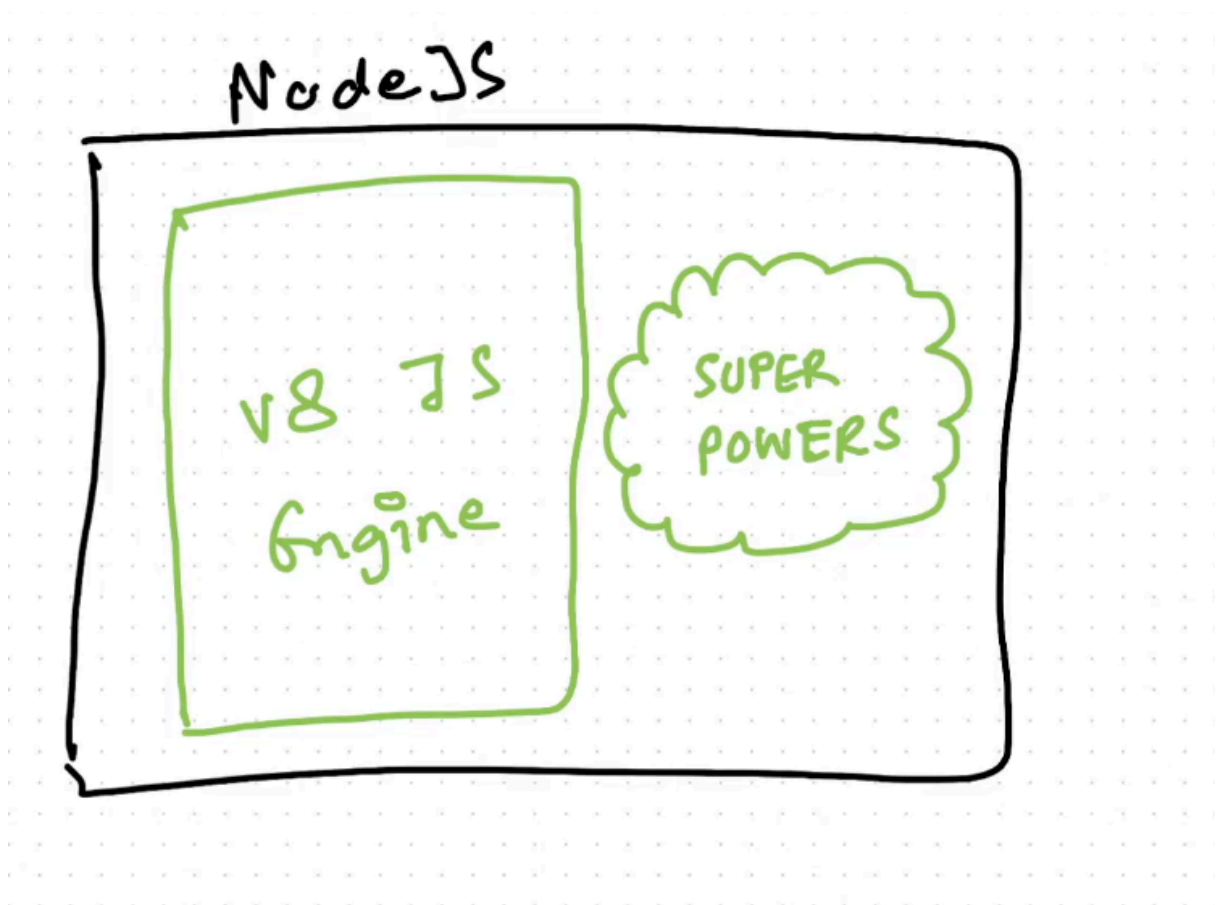
- install → <https://nodejs.org/en/download>
- check node -v
- check npm -v
- and the easiest way to write node code is Node REPL
- REPL
 - READ the code,
 - EVALUATE the code,
 - PRINT the output,
 - run a LOOP
- so when in terminal you hit command ⇒ node
- you go on REPL mode which is made of c++
- over here any NodeJS and JS Code Work Perfectly Fine

```
PS G:\Interview-Prep\NamasteNode\Season-1> node -v
v22.18.0
PS G:\Interview-Prep\NamasteNode\Season-1> npm -v
10.9.3
PS G:\Interview-Prep\NamasteNode\Season-1> node
Welcome to Node.js v22.18.0.
Type ".help" for more information.
> 1+1
2
> var a = "dipesh joshi"
undefined
> a
'dipesh joshi'
> var i = 10;
undefined
> var j = 20;
undefined
> i+j
30
>
```

REPL MODE

- nodejs is JS Runtime Environment . Now You Know Why. behind the scene its working on V8

- Whatever JS code we put behind the scene nodejs gives to v8 and its gives our output



- same happen when we code browser console .but superpower don't have
- but its not have for production so we make file and do code there

App.js

```
var name = "Namste NodeJS";  
var a = 10;  
var b = 20;  
console.log(name);  
console.log(a + b);
```

- How to Run this code on Node or you can say JS Runtime Environment
- open terminal → node filename.js

OUTPUT

Namste Nodejs

30

till T.S. → 23 MIN

Global Object In NodeJS

- in Console We have Global Object which is called ⇒ "window"
 - if we write "this " then its also gives use window{ }
 - and this is not given by V8 and .it gives By Browser
-
- in node js we have also global object and it's called global

```
console.log(global)
```

OUTPUT

(c) Microsoft Corporation. All rights reserved.

G:\Interview-Prep\NamasteNode\Season-1>node "g:\Interview-Prep\NamasteNode\Season-1\app.js"
Namste NodeJS

30

```
<ref *1> Object [global] {  
  global: [Circular *1],  
  clearImmediate: [Function: clearImmediate],  
  setImmediate: [Function: setImmediate] {  
    [Symbol(nodejs.util.promisify.custom)]: [Getter]  
  },  
  clearInterval: [Function: clearInterval],  
  clearTimeout: [Function: clearTimeout],  
  setInterval: [Function: setInterval],  
  setTimeout: [Function: setTimeout] {  
    [Symbol(nodejs.util.promisify.custom)]: [Getter]  
  },  
  queueMicrotask: [Function: queueMicrotask],  
  structuredClone: [Function: structuredClone],  
  atob: [Function: atob],  
  btoa: [Function: btoa],  
  performance: [Getter/Setter],  
  fetch: [Function: fetch],  
  navigator: [Getter],  
  crypto: [Getter]  
}
```

G:\Interview-Prep\NamasteNode\Season-1>

- setTimeout, setInterval, clearTimeout all function ,method comes from Global Object of NodeJS
- global is part of superpower and not comes from v8 .
- Global gives lots of function, method and properties
- v8 engine only understand core JS . it does not understand Global
- it only understand when nodejs gives access of global inside v8 engine

browser's Global Object

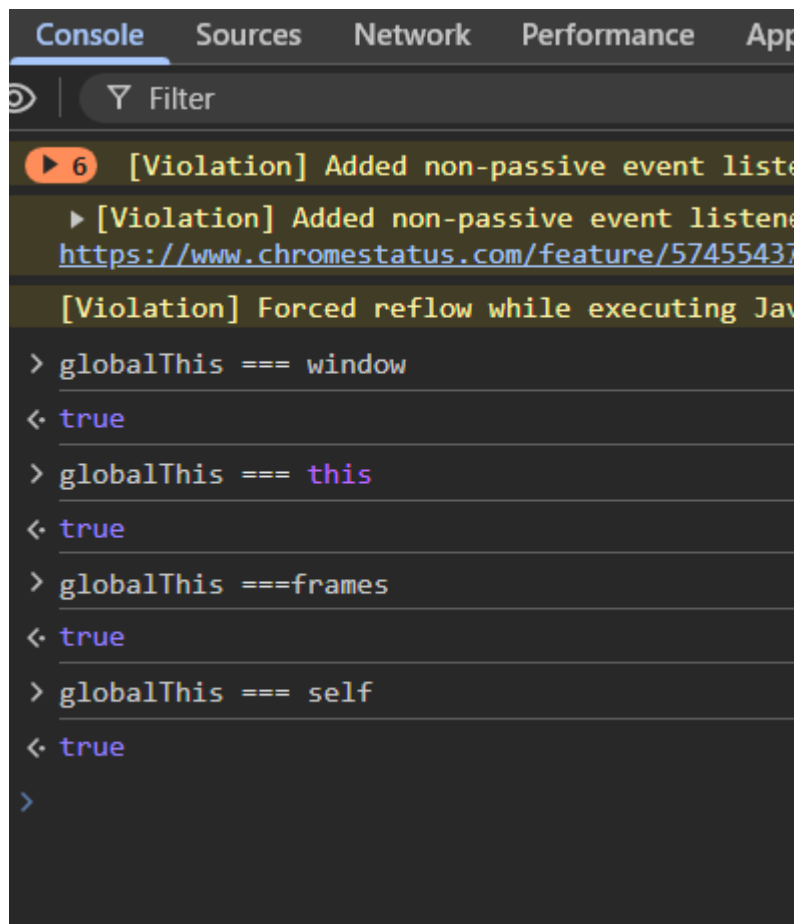
⇒ Window ⇒ "this", "self", "frames" Keyword also Represent Global

NodeJS Global Object

⇒ Global ⇒ "this" Keyword Don't Represent Global

globalThis Refer as Global Object in NodeJS and Browser. Any JS Runtime Environment

```
console.log(global);           // Gives Property Of Global Many function lik
                               // setTimeout,setInterval
console.log(this)              // Empty Object {}
console.log(globalThis);       // Gives Property Of Global Many function li
                               // ke setTimeout,setInterval
console.log(globalThis === global); // true
```



The screenshot shows a web browser's developer console with the 'Console' tab selected. At the top, there are tabs for 'Console', 'Sources', 'Network', 'Performance', and 'App'. Below the tabs is a search bar with a magnifying glass icon and the text 'Filter'. The console displays several messages:

- A yellow warning message with a play icon and the number '6': `[Violation] Added non-passive event listener to a passive listener at https://www.chromestatus.com/feature/57455437`
- A yellow warning message: `[Violation] Forced reflow while executing JavaScript`
- A series of console logs showing the result of `globalThis ===` comparisons:
 - `> globalThis === window`
`< true`
 - `> globalThis === this`
`< true`
 - `> globalThis === frames`
`< true`
 - `> globalThis === self`
`< true`
 - `>`