Namaste Node JS

Episode-1

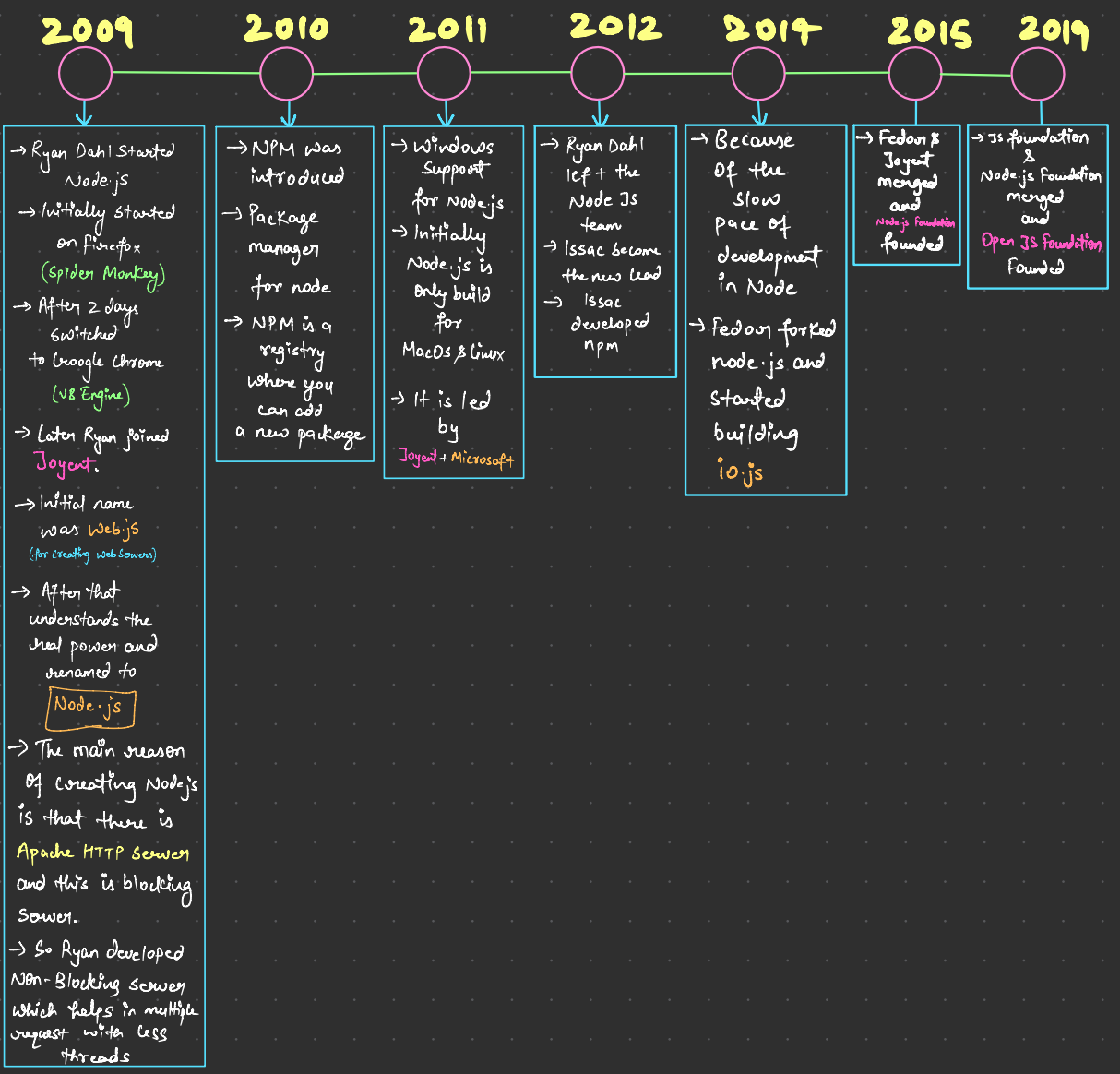
What is Node.js?

- Node.js is a javascript runtime build on chrome V8 Javascript engine.
- Node is a cross-platform, open source maintained by (Open Js foundation) Javascript runtime environment that can run on "Windows", "Linux" and more.
- Node Js helps us to run Javascript outside of the browser.
- Node Js has an event-driven architecture capable of asynchronous IO which is also known as Non-Blocking IO.
- The first version of Node Js was released in 2009.
- Node Js was developed by Ryan Dhal.

History of Node Js

- Wherever there is Js, there must be a Javascript Engine.

Node.JS Timeline

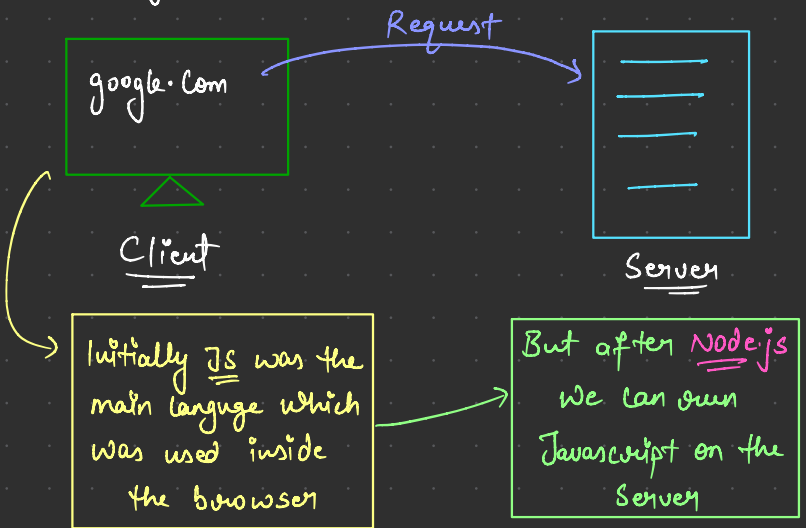


Episode-2

Node JS - Javascript on Server

What is Server?

→ Server is nothing but a remote computer / CPU



→ With Javascript come on Server, it gives us the opportunity to develop the Full Stack application.

→ Node.js is C++ code.

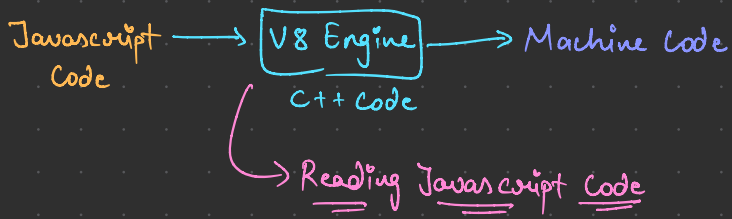
→ Even V8 Engine is written in C++ program.

What is V8?

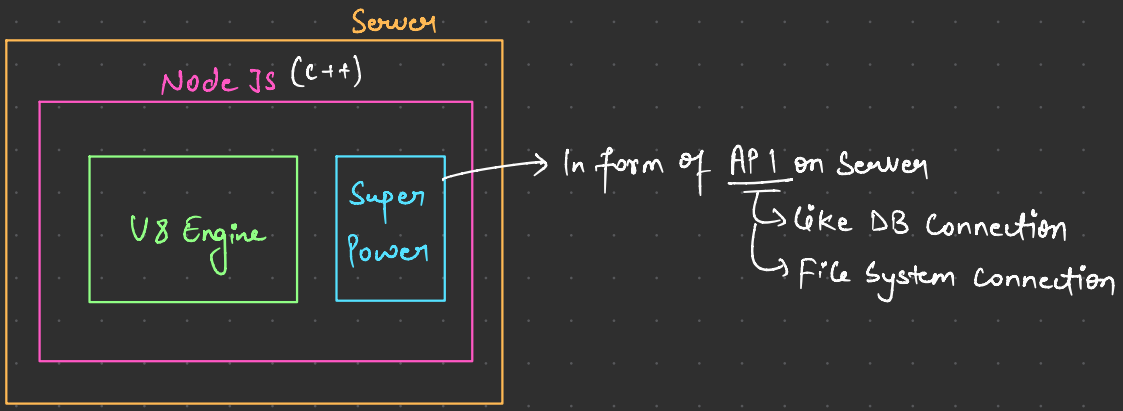
V8 is Google's open source high-performance JavaScript and WebAssembly engine, written in C++.

→ V8 can be embedded into any C++ application.

→ The job of V8 engine is to execute JS code.



→ Node JS is a C++ application with V8 embedded into it



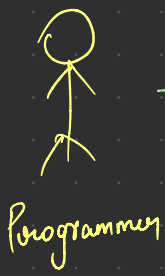
→ V8 is a JS Engine that follows ECMAScript standards.

→ ECMAScript is a Standard for Scripting languages like Javascript.

- Standards/Rules
- JS Engines follow these Standards.

→ So, V8 follows the ECMAScript standards and Node.js gives the extra Super-power.

V8 is a C++ Code. What does it do & why it is a C++ Code ?



Writes → Javascript Code (High-level Code)

↓ Processed by

C++ (JS Engine)

↓ Converts

Machine Code (Low level Code)

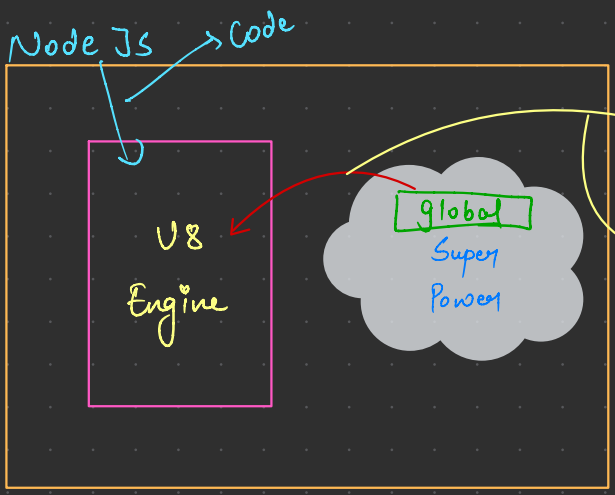
↓ This Machine Code is understandable by the Computer

Computer (Binary)

Episode-3

→ As in browsers, the global object is "Windows". Same in Node.js the global object is "global".

→ global is not the part of the V8 Engine and it is one of the Superpower of node.js. Which is given to us by node.js.



This global object access is given inside the V8 Engine

V8 Engine does not understands global. It only understands global when Node JS gives access to the V8 Engine

When we console "global" and "window" we get

This

```
app.js
1   console.log(global);
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS COD

```
pratiksingh@Pratiks-MacBook-Pro Namaste Node JS % node app.js
<ref *1> Object [global] {
  global: [Circular *1],
  queueMicrotask: [Function: queueMicrotask],
  clearImmediate: [Function: clearImmediate],
  setImmediate: [Function: setImmediate] {
    [Symbol(nodejs.util.promisify.custom)]: [Getter]
  },
  structuredClone: [Function: structuredClone],
  clearInterval: [Function: clearInterval],
  clearTimeout: [Function: clearTimeout],
  setInterval: [Function: setInterval],
  setTimeout: [Function: setTimeout] {
    [Symbol(nodejs.util.promisify.custom)]: [Getter]
  },
  atob: [Function: atob],
  btoa: [Function: btoa],
  performance: Performance {
    nodeTiming: PerformanceNodeTiming {
      name: 'node',
      entryType: 'node',
      startTime: 0,
      duration: 72.5864999294281,
      nodeStart: 18.006916046142578,
      v8Start: 31.774291038513184,
      bootstrapComplete: 66.66125011444092,
      environment: 47.724250078201294,
      loopStart: -1,
      loopExit: -1,
      idleTime: 0
    },
    timeOrigin: 1723388725832.482
  },
  fetch: [AsyncFunction: fetch]
}
```

This same will happen in case of browsers when we console window

```
console.log(window)                                   Console settings
                                                                VM140:1
Window {0: Window, window: Window, self: Window, document: document, name: '', location: Location, …} i
  0: Window {window: Window, self: Window, document:
  JSCompiler_renameProperty: ƒ (t,e)
  alert: ƒ alert()
  atob: ƒ atob()
  blur: ƒ blur()
  btoa: ƒ btoa()
  caches: CacheStorage {}
  cancelAnimationFrame: ƒ cancelAnimationFrame()
  cancelIdleCallback: ƒ cancelIdleCallback()
  captureEvents: ƒ captureEvents()
  chrome: {metricsPrivate: {…}, loadTimes: ƒ, csi: ƒ,
  clearInterval: ƒ clearInterval()
  clearTimeout: ƒ clearTimeout()
  clientInformation: Navigator {vendorSub: '', produc
  close: ƒ close()
  closed: false
  confirm: ƒ confirm()
  cookieStore: CookieStore {onchange: null}
  cr: {webUIResponse: ƒ, webUIListenerCallback: ƒ}
  createImageBitmap: ƒ createImageBitmap()
  credentialless: false
  crossOriginIsolated: false
  crypto: Crypto {subtle: SubtleCrypto}
  customElements: CustomElementRegistry {}
  devicePixelRatio: 2
  document: document
  documentPictureInPicture: DocumentPictureInPicture
  event: undefined
  external: External {}
  fence: null
  fetch: ƒ fetch()
  find: ƒ find()
  focus: ƒ focus()
  frameElement: null
  frames: Window {0: Window, window: Window, self: Wi
  getComputedStyle: ƒ getComputedStyle()
  getScreenDetails: ƒ getScreenDetails()
  getSelection: ƒ getSelection()
  history: History {length: 1, scrollRestoration: 'au
  indexedDB: IDBFactory {}
  innerHeight: 858
  innerWidth: 957
  isSecureContext: true
  launchQueue: LaunchQueue {}
```

When we console "this" in Node JS and in Browser

Node JS

```
1   console.log(this);
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS COD

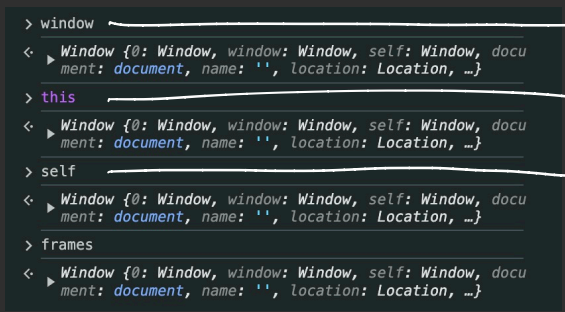
```
pratiksingh@Pratiks-MacBook-Pro Namaste Node JS % node app.js
{}
pratiksingh@Pratiks-MacBook-Pro Namaste Node JS %
```

Browser

```
console.log(this)
                                                      VM480:1
Window {0: Window, window: Window, self: Window, document: document, name: '', location: Location, …} i
  0: Window {window: Window, self: Window, document:
  JSCompiler_renameProperty: ƒ (t,e)
  alert: ƒ alert()
  atob: ƒ atob()
  blur: ƒ blur()
  btoa: ƒ btoa()
  caches: CacheStorage {}
  cancelAnimationFrame: ƒ cancelAnimationFrame()
  cancelIdleCall  function cancelAnimationFrame() { [native code] }
  captureEvents: ƒ
  chrome: {metricsPrivate: {…}, loadTimes: ƒ, csi: ƒ,
  clearInterval: ƒ clearInterval()
  clearTimeout: ƒ clearTimeout()
  clientInformation: Navigator {vendorSub: '', produc
  close: ƒ close()
  closed: false
  confirm: ƒ confirm()
  cookieStore: CookieStore {onchange: null}
  cr: {webUIResponse: ƒ, webUIListenerCallback: ƒ}
  createImageBitmap: ƒ createImageBitmap()
  credentialless: false
  crossOriginIsolated: false
  crypto: Crypto {subtle: SubtleCrypto}
  customElements: CustomElementRegistry {}
  devicePixelRatio: 2
  document: document
  documentPictureInPicture: DocumentPictureInPicture
  event: undefined
  external: External {}
  fence: null
  fetch: ƒ fetch()
  find: ƒ find()
  focus: ƒ focus()
  frameElement: null
  frames: Window {0: Window, window: Window, self: Wi
  getComputedStyle: ƒ getComputedStyle()
  getScreenDetails: ƒ getScreenDetails()
  getSelection: ƒ getSelection()
  history: History {length: 1, scrollRestoration: 'au
  indexedDB: IDBFactory {}
  innerHeight: 858
  innerWidth: 1090
  isSecureContext: true
  launchQueue: LaunchQueue {}
```

In Browsers "this" refers to the same object, "window" but its not the same case, and it refers to empty object

* In the Browser if we write "window", "this", "self", "frames" all of these gives you the global object.



→ Browsers named it "window"

→ Concept "this" pointing to global object

→ In web workers "self" points to global object

→ Node JS started using "global" as global object

→ Because of this there is lot of confusion and after that Open JS Foundation comes up with a Standard global object for all the runtime environment and there should be a single way to represent it.

→ So, finally Open JS Foundation comes up with "globalThis", and globalThis refers to global object in all javascript runtime.

Node JS

```
1  console.log(globalThis);
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS COD

```
pratiksinghe@Pratiks-MacBook-Pro Namaste Node JS % node app.js
<ref *1> Object [global] {
  global: [Circular *1],
  queueMicrotask: [Function: queueMicrotask],
  clearImmediate: [Function: clearImmediate],
  setImmediate: [Function: setImmediate] {
    [Symbol(nodejs.util.promisify.custom)]: [Getter]
  },
  structuredClone: [Function: structuredClone],
  clearInterval: [Function: clearInterval],
  clearTimeout: [Function: clearTimeout],
  setInterval: [Function: setInterval],
  setTimeout: [Function: setTimeout] {
    [Symbol(nodejs.util.promisify.custom)]: [Getter]
  },
  atob: [Function: atob],
  btoa: [Function: btoa],
  performance: Performance {
    nodeTiming: PerformanceNodeTiming {
      name: 'node',
      entryType: 'node',
      startTime: 0,
      duration: 72.42704200744629,
      nodeStart: 17.885541915893555,
      v8Start: 31.32854199409485,
      bootstrapComplete: 66.40279197692871,
      environment: 47.43129205703735,
      loopStart: -1,
      loopExit: -1,
      idleTime: 0
    },
    timeOrigin: 1723400554763.945
  },
  fetch: [AsyncFunction: fetch]
}
```

Browsers

```
console.log(globalThis)
```

VM744:1

```
Window {0: Window, window: Window, self: Window, document: document, name: '', location: Location, …}
  0: Window {window: Window, self: Window, document: document, name: '', location: Location, …}
  JSCompiler_renameProperty: ƒ (t,e)
  alert: ƒ alert()
  atob: ƒ atob()
  blur: ƒ blur()
  btoa: ƒ btoa()
  caches: CacheStorage {}
  cancelAnimationFrame: ƒ cancelAnimationFrame()
  cancelIdleCallback: ƒ cancelIdleCallback()
  captureEvents: ƒ captureEvents()
  chrome: {metricsPrivate: {…}, loadTimes: ƒ, csi: ƒ, …}
  clearInterval: ƒ clearInterval()
  clearTimeout: ƒ clearTimeout()
  clientInformation: Navigator {vendorSub: '', product
  close: ƒ close()
  closed: false
  confirm: ƒ confirm()
  cookieStore: CookieStore {onchange: null}
  cr: {webUIResponse: ƒ, webUIListenerCallback: ƒ}
  createImageBitmap: ƒ createImageBitmap()
  credentialless: false
  crossOriginIsolated: false
  crypto: Crypto {subtle: SubtleCrypto}
  customElements: CustomElementRegistry {}
  devicePixelRatio: 2
  document: document
  documentPictureInPicture: DocumentPictureInPicture
  event: undefined
  external: External {}
  fence: null
  fetch: ƒ fetch()
  find: ƒ find()
  focus: ƒ focus()
  frameElement: null
```