



Presentation 2023

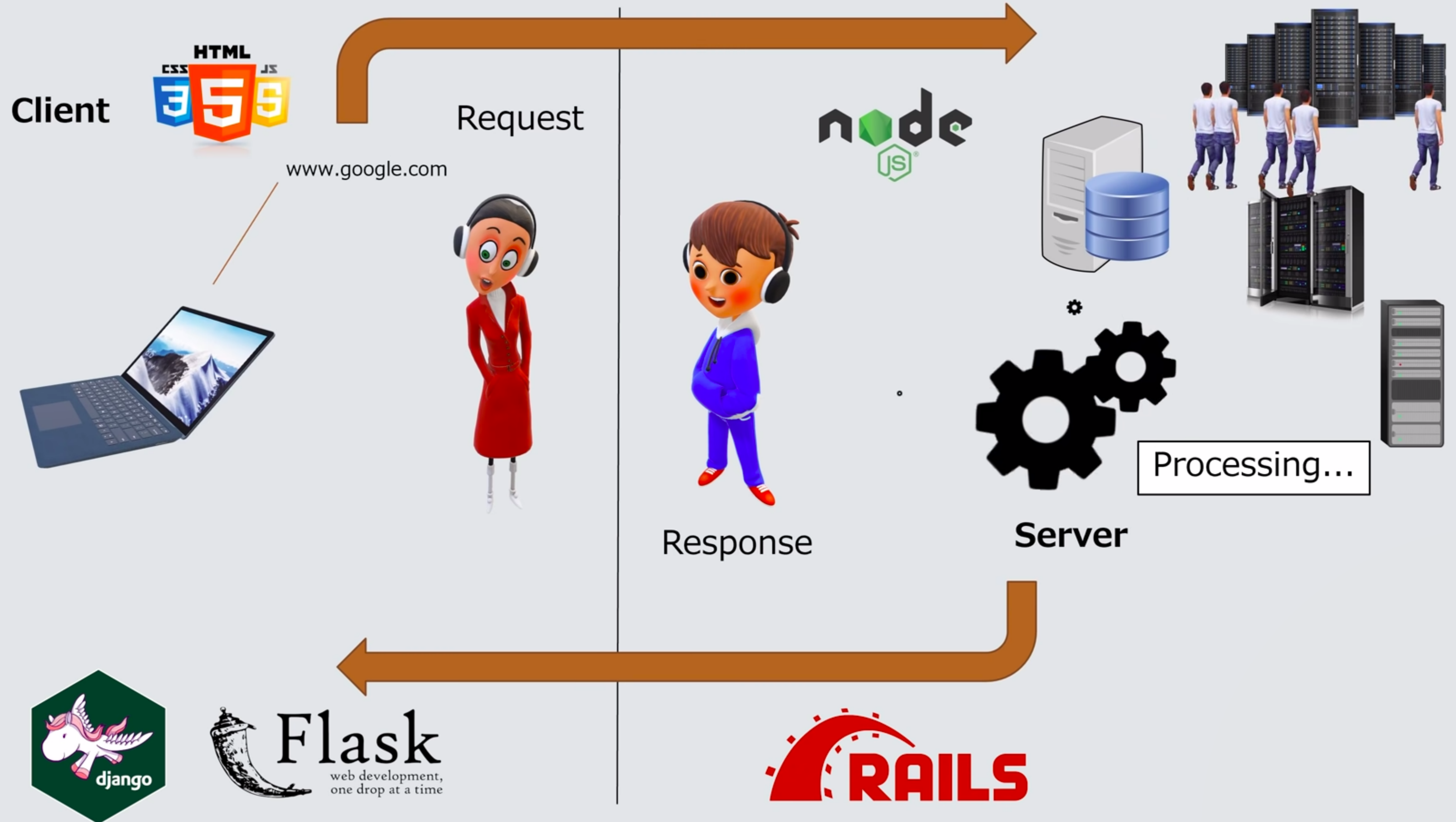
# INTRODUCTION TO NODE JS

Presented By:

**Vishal Thapa**

[thapavishal07.vercel.app](https://thapavishal07.vercel.app)

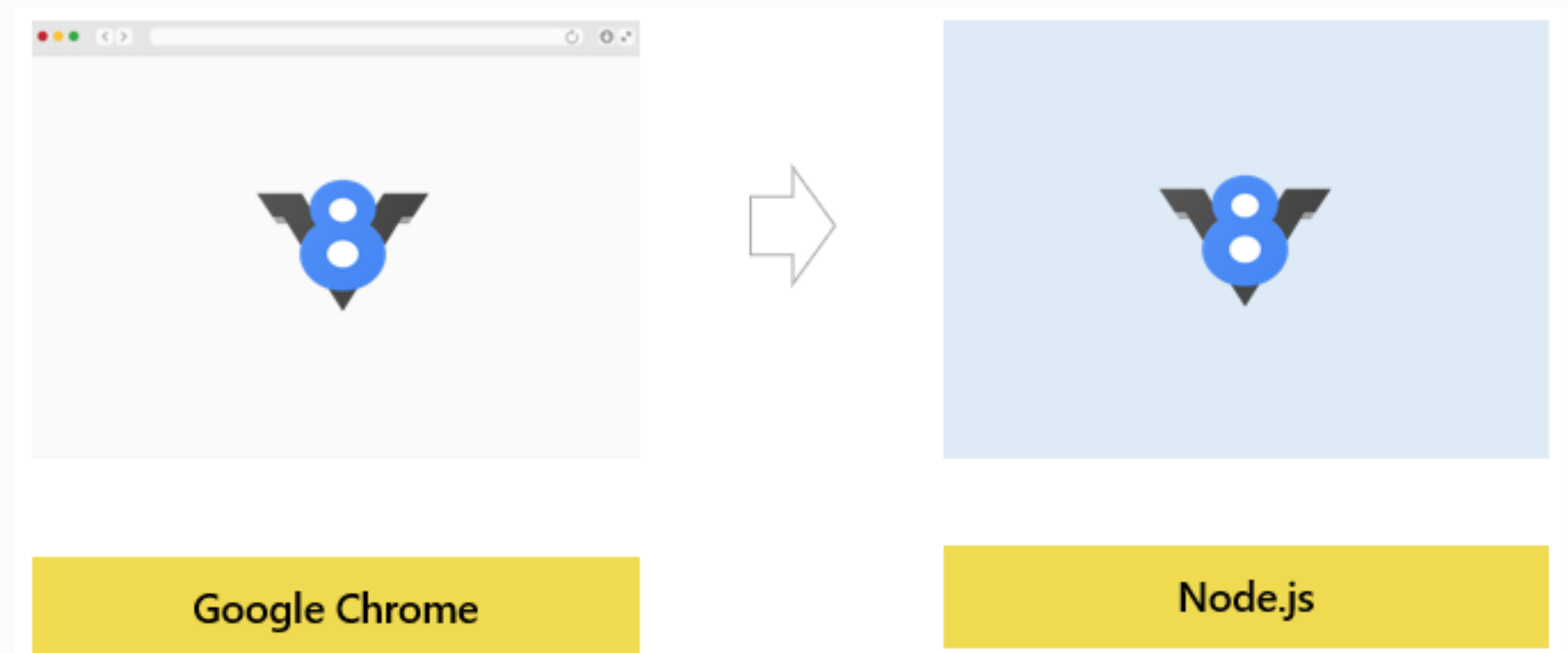
# The basics of frontend and backend



# History

## Node.js

- In 2009, Ryan Dahl (Google Engineer) saw Javascript doesn't run outside browser
- Took **V8 engine** and embedded in a **C++ program** and called it **Node.exe**, which later become **Node.js**



# Node JS

## Key Features

01



Open-source & cross-platform javascript environment

- **Open source:** Free
- **Cross-platform:** windows, mac, linux
- **JavaScript runtime environment:**
  - Node js allows you to run JavaScript code outside of a web browser.
  - When we run our code i.e npm start
  - Throws error on the spot

```
JS index.js filesysandcommandLine X JS index.js \ JS index.js jsondatawithfile
fileysandcommandLine > JS index.js > chalk
4 const name = chalk.blue("Hi I am blue color");
5
6 yargs.command({
7   command: "add",
8   describe: "adding notes",
9   builder: {
10
11   }
12 });

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
PS E:\Nodejs\fileysandcommandLine> npm run server

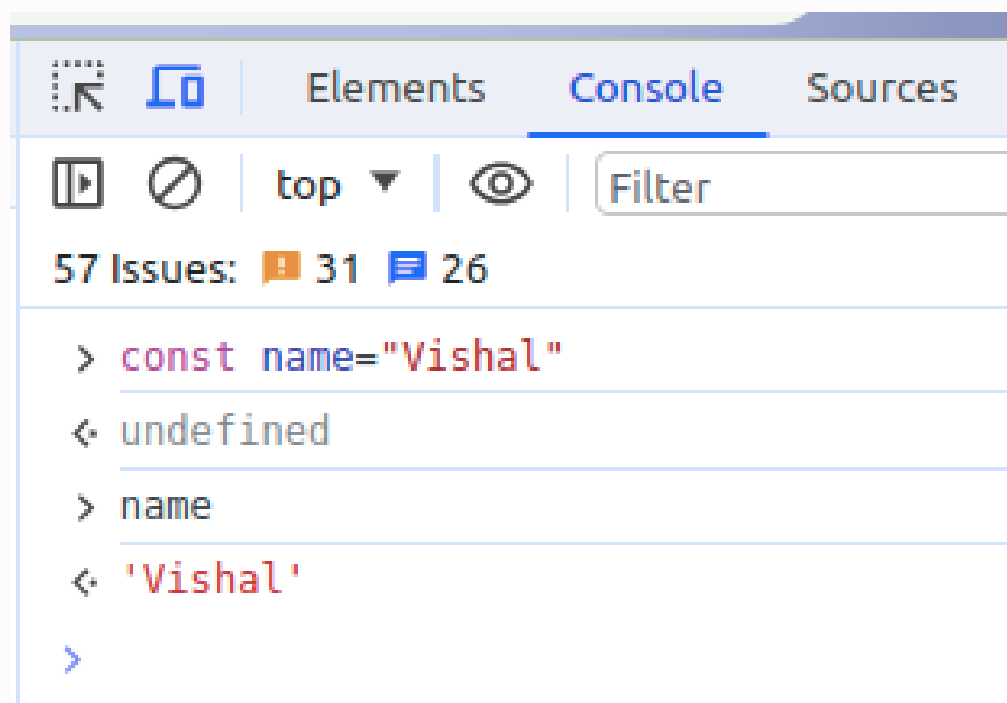
> nodejswork@1.0.0 server E:\Nodejs\fileysandcommandLine
> nodemon server.js

[nodemon] 2.0.4
[nodemon] to restart at any time, enter `rs`
[nodemon] watching path(s): *.*
[nodemon] watching extensions: js,mjs,json
[nodemon] starting `node server.js index.js`
[nodemon] clean exit - waiting for changes before restart
```

```
JS index.js filesysandcommandLine X JS index.js \ JS index.js jsondatawithfile
fileysandcommandLine > JS index.js > ...
1 const chalk = require("chalk");
2 const yargs = require("yargs");
3
4 cont name = chalk.blue("Hi I am blue color");
5
6 yargs.command({

PROBLEMS 1 OUTPUT DEBUG CONSOLE TERMINAL
E:\Nodejs\fileysandcommandLine\index.js:4
cont name = chalk.blue("Hi I am blue color");
    ^^^^^
SyntaxError: Unexpected identifier
    at wrapSafe (internal/modules/cjs/loader.js:1054:16)
    at Module._compile (internal/modules/cjs/loader.js:1102:27)
    at Object.Module._extensions..js (internal/modules/cjs/loader.js:1158:10)
    at Module.load (internal/modules/cjs/loader.js:986:32)
    at Function.Module._load (internal/modules/cjs/loader.js:879:14)
    at Function.executeUserEntryPoint [as runMain] (internal/modules/run_main.js:71:12)
    at internal/main/run_main_module.js:17:47
[nodemon] app crashed - waiting for file changes before starting...
```

- V8 is the **JavaScript engine** that powers **Google Chrome**.
- It's the thing that takes our JavaScript and executes it while browsing with Chrome.
- javascript code **runs only** in **browser before node js**
- Other JS engines
  - **Other browsers** have their **own JavaScript engine**:
    - **Firefox** has SpiderMonkey
    - **Safari** has JavaScriptCore (also called Nitro)
    - **Edge** was originally based on Chakra but has more recently been rebuilt using Chromium and the V8 engine.



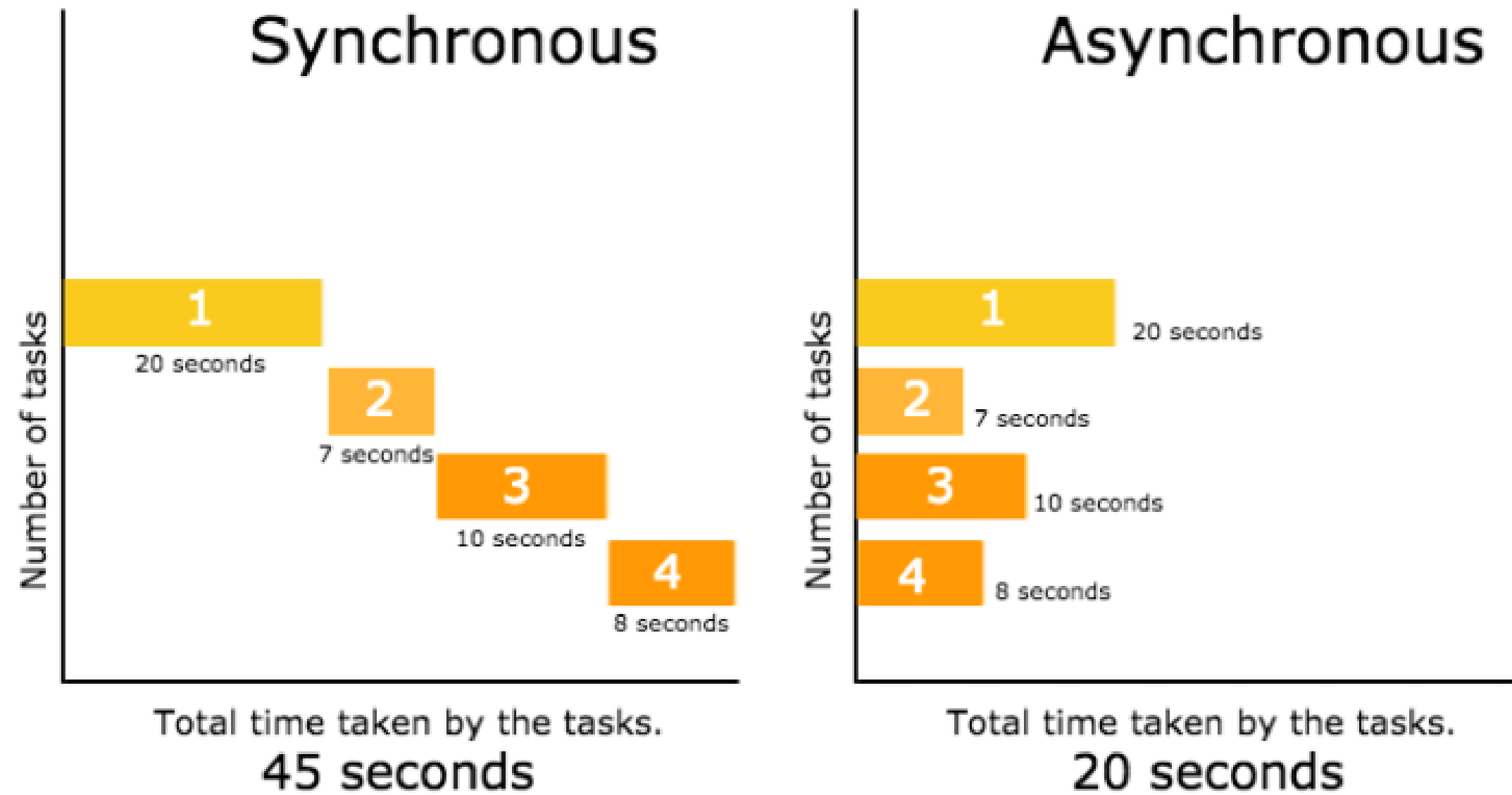


03



Asynchronous

- Non-blocking I/O model



- Node.js runs **JavaScript code** in a **single/main thread**, which means that your code can only do one task at a time.
- Node.js can do multithreading using libuv library , which handles I/O operations like reading files from a disk or network requests (**event loop**)
- Through the use of **hidden threads**, Node.js provides **asynchronous** methods that allow your code to make I/O requests **without blocking the main thread**.

# Working

## Mechanism

- Example 1
- **Question 1.** What will be the output of following code?

Answer

```
console.log("starting up")

setTimeout(() => {
  console.log("2 second log")
}, 2000)

setTimeout(() => {
  console.log("0 second log")
}, 0)

console.log("finishing up")
```

Answer



# Advantages

Node js

- **JavaScript Everywhere**

Allows **developers** to use **JavaScript** for **both front-end and back-end** site.

- **Cross-Platform Compatibility**

Node.js **applications** can be **deployed** across **various platforms**, ensuring **flexibility** and **ease of deployment**.

- **Community and Ecosystem**

Boasts a **large and active community** contributing **modules and libraries** to the **Node Package Manager (npm)**, providing a wide range of functionalities and solutions.

- **Cost-Effective**

Can handle a **large number of concurrent connections** with **less hardware**, making it a cost-effective choice for scalable applications.

# Dive In

Node js

- **Node.js Frameworks:**

**Express.js:** Perhaps the **most well-known and widely used** framework for Node.js.

Koa.js, Sails.js, NestJS, Hapi.js, etc

- **Programming language supported by node.js:**

**Javascript (ES6+), Typescript, CoffeeScript, Rust (through Neon), etc**

Answer

# Demo

Node js

- Demo 1

Answer



**THANK  
YOU**

