



BASIC INTRODUCTION TO REACT JS, NODE JS, NPM

BASIC INTRODUCTION TO REACT JS



React JS is a JavaScript library or framework used to build user interfaces (UI) in web applications.

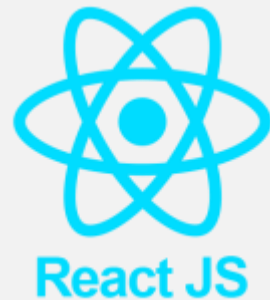
However, React can interact with the backend through various means, such as sending HTTP requests (for example, using Fetch API or axios) to the backend server to fetch or send data, or communicating via WebSocket protocol for real-time updates.

To create a backend for web applications, you would typically use technologies and frameworks such as Node.js

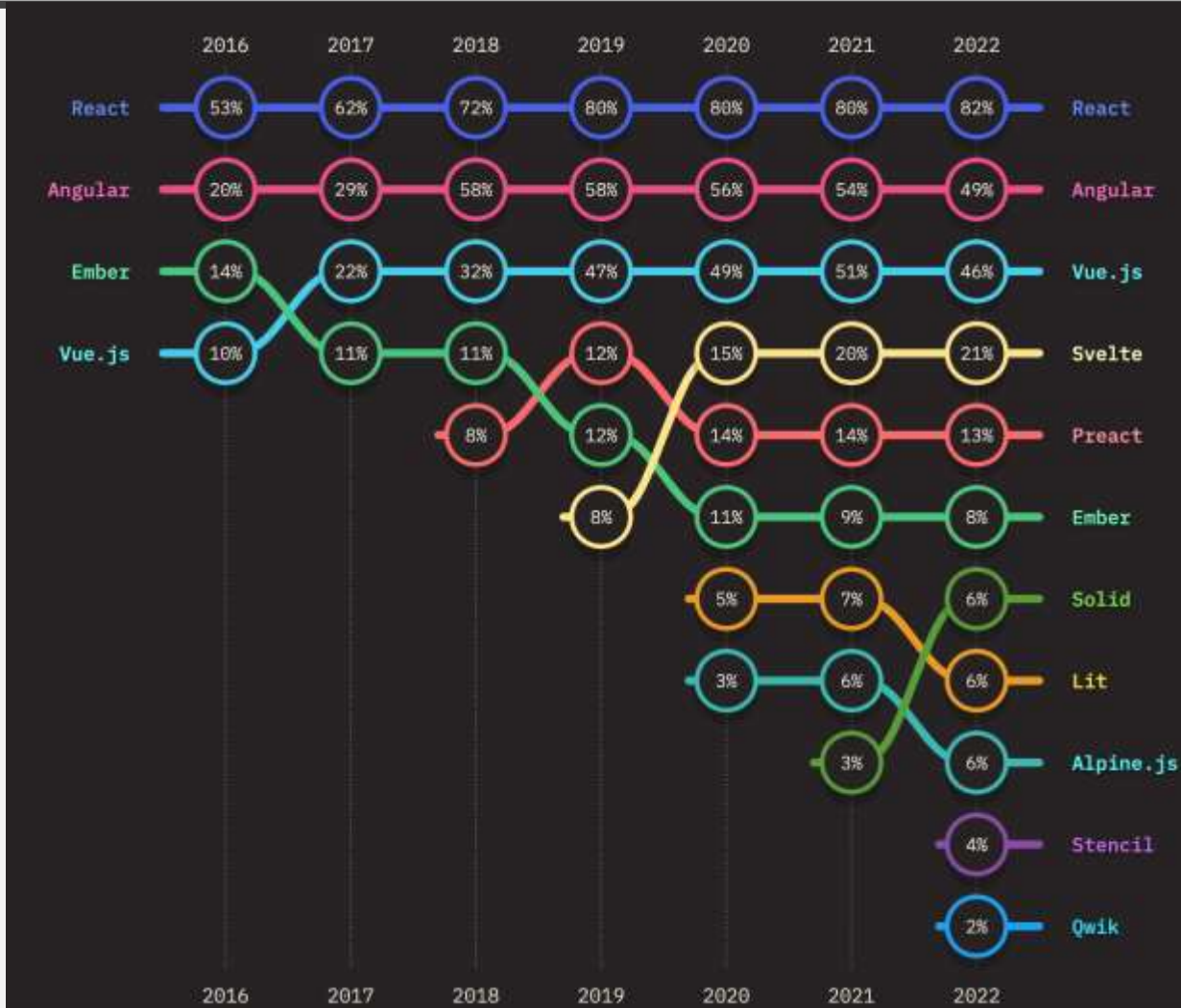
BASIC INTRODUCTION TO REACT JS



A **framework** is a **framework or a set of code that has specific functionalities**, which can include functions, objects, or classes that developers can directly use. Essentially, its purpose is to **assist developers in avoiding repetition of code across different pages**.



BASIC INTRODUCTION TO REACT JS



BASIC INTRODUCTION TO REACT JS



React JS was initially built by Jordan Walker, a software engineer at Facebook, now known as Meta, and was open-sourced in 2013.



BASIC INTRODUCTION TO REACT JS



React JS employs the concept of 'virtual DOM' to enhance application performance by minimizing the number of direct operations on the actual DOM.

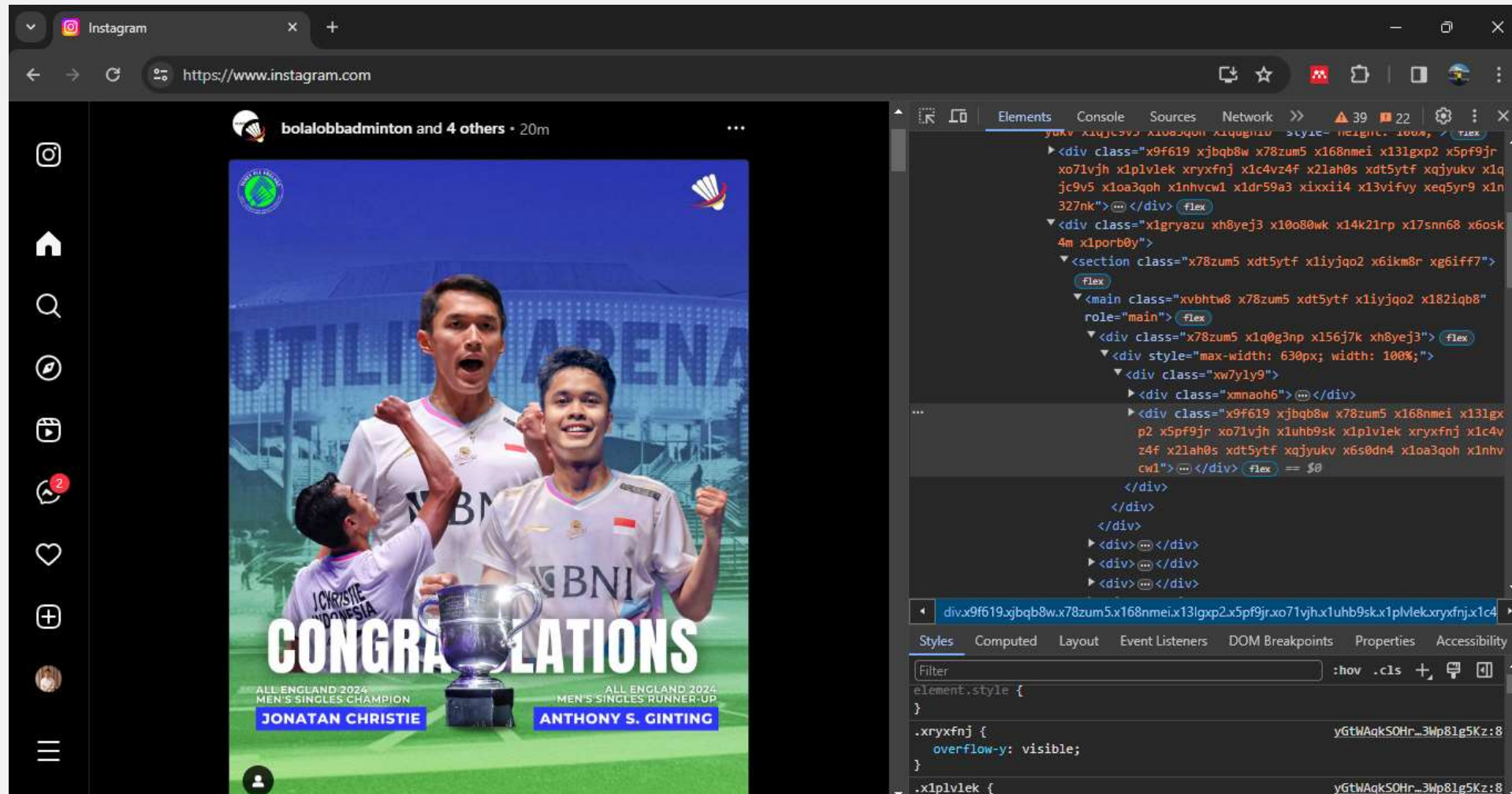
Therefore, React JS efficiently updates or re-renders a component that has real-time changes.

BASIC INTRODUCTION TO REACT JS

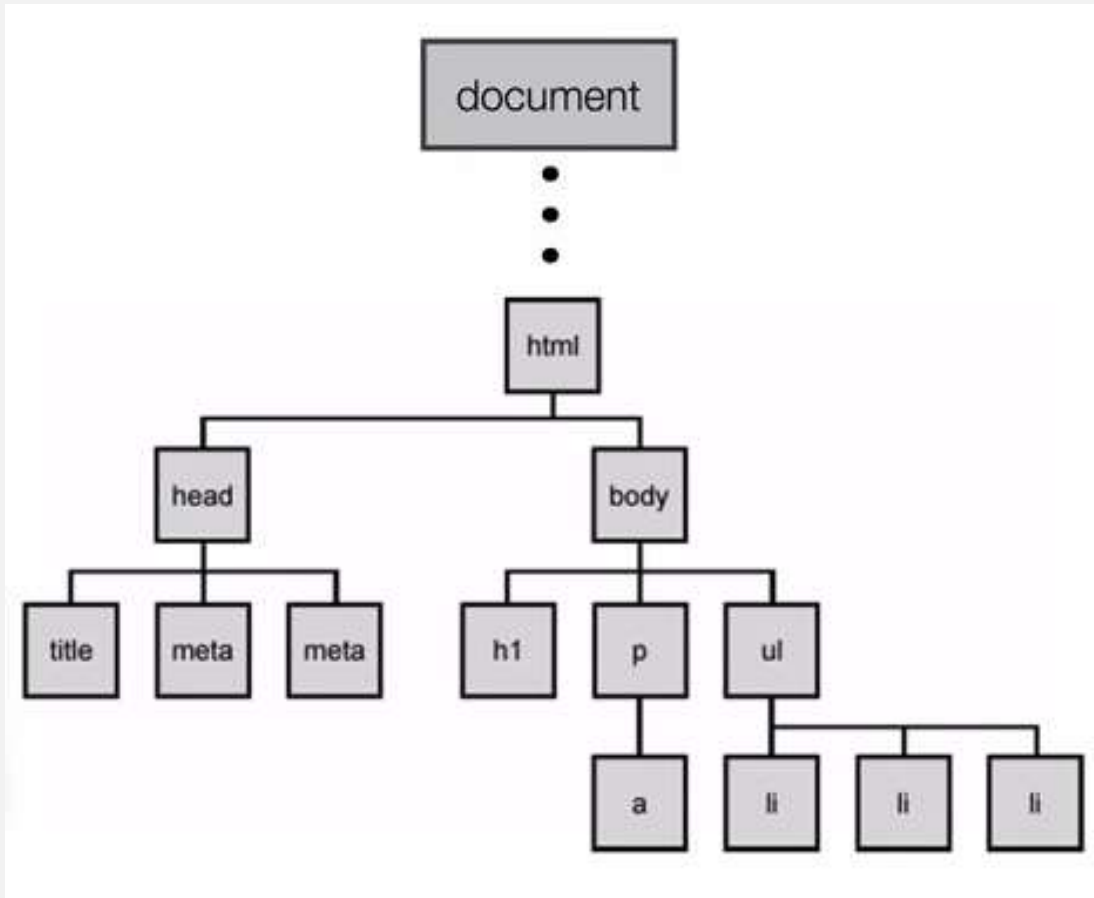


The **DOM** stands for **Document Object Model**. It's a **representation of the HTML or XML document structure, typically used to create web pages**, in the form of a structured tree within a web browser.

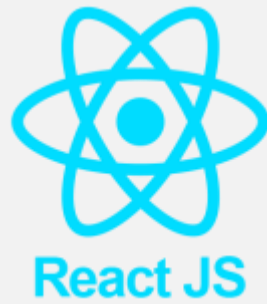
BASIC INTRODUCTION TO REACT JS



BASIC INTRODUCTION TO REACT JS

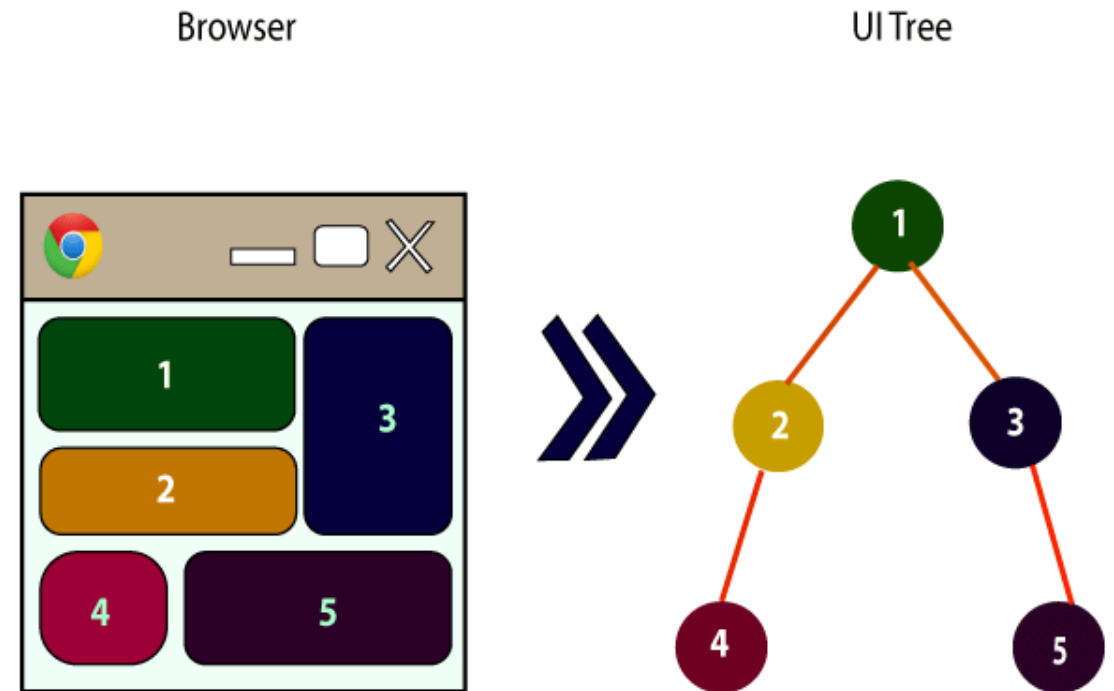


BASIC INTRODUCTION TO REACT JS



In ReactJS, components are the basic building blocks of a React application. They are reusable UI pieces that allow you to divide the user interface into independent parts and focus on specific aspects of logic and display.

Each component typically has its own properties and state. Properties are data passed to the component, while state is data controlled by the component itself. When the state or properties change, the component will automatically update.



BASIC INTRODUCTION TO REACT JS



In **ReactJS**, elements are simple objects that represent components or **HTML** code that you'll see on the screen. Elements can be created using **JSX** syntax, which is similar to **HTML**, to build the structure of your application's interface.

```
<div id="ReactJS-App0"></div>
```

BASIC INTRODUCTION TO REACT JS



What needs to be prepared to become a ReactJS developer.

The HTML logo, consisting of the text "<html>" in white on an orange rectangular background.

<html>

BASIC INTRODUCTION TO REACT JS



What needs to be prepared to become a ReactJS developer.



BASIC INTRODUCTION TO REACT JS



What needs to be prepared to become a ReactJS developer.



BASIC INTRODUCTION TO REACT JS



What needs to be prepared to become a ReactJS developer.



CDN Links



BASIC INTRODUCTION TO REACT JS

Installation -> create container -> create component -> render

Both React and ReactDOM are available over a CDN.

```
<script crossorigin src="https://unpkg.com/react@18/umd/react.development.js"></script>  
<script crossorigin src="https://unpkg.com/react-dom@18/umd/react-dom.development.js"></script>
```

The versions above are only meant for development, and are not suitable for production.

Minified and optimized production versions of React are available at:

```
<script crossorigin src="https://unpkg.com/react@18/umd/react.production.min.js"></script>  
<script crossorigin src="https://unpkg.com/react-dom@18/umd/react-dom.production.min.js"></script>
```

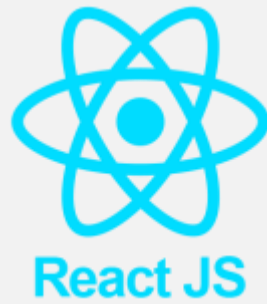
To load a specific version of `react` and `react-dom`, replace `18` with the version number.

BASIC INTRODUCTION TO REACT JS



```
<> index.html > html
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <meta name="viewport" content="width=device-width, initial-scale=1.0">
6      <title>Document</title>
7
8      <!-- CDN React Installation -->
9      <script crossorigin src="https://unpkg.com/react@18/umd/react.development.js"></script>
10     <script crossorigin src="https://unpkg.com/react-dom@18/umd/react-dom.development.js"></script>
11     <script src="https://unpkg.com/babel-standalone@6/babel.min.js"></script>
12
13     <title>ReactJs-0</title>
14 </head>
15 <body>
16     <!-- Create Container -->
17     <div id="ReactJS-App0"></div>
18
19     <!-- Create Component -->
20     <script type = "text/babel">
21         class HelloReact extends React.Component {
22             render(){
23                 return <h1>Hello ReactJs</h1>
24             }
25         }
26
27         // Render Component
28         ReactDOM.render(<HelloReact />, document.getElementById('ReactJS-App0'));
29     </script>
30 </body>
31 </html>
```

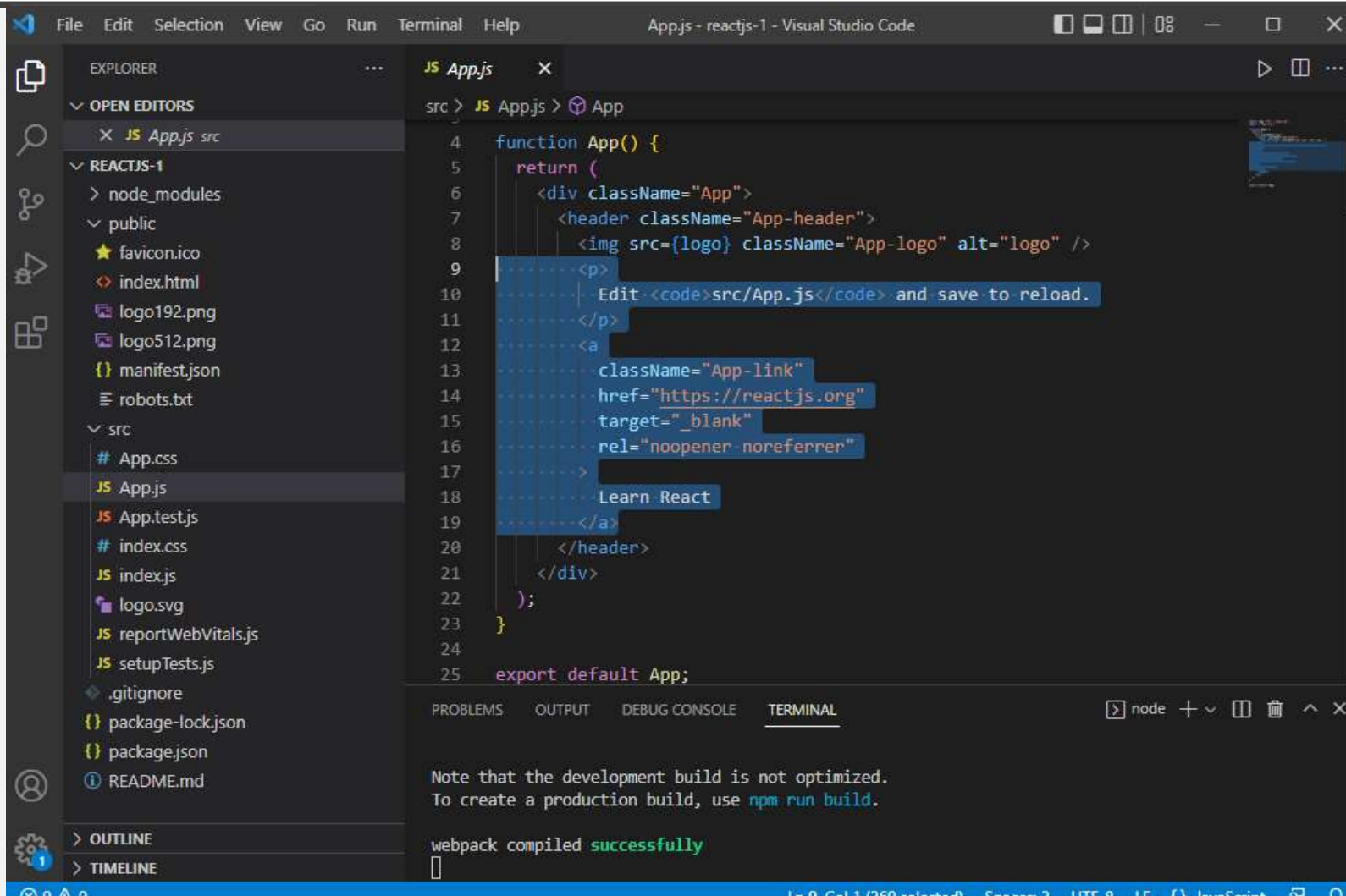
BASIC INTRODUCTION TO REACT JS



A screenshot of the Bootstrap 5.3 documentation page for the Accordion component. The page has a purple header with navigation links: Docs, Examples, Icons, Themes, Blog, and a search bar. The left sidebar shows a tree view with 'Forms' and 'Components' sections. The 'Components' section is expanded, showing 'Accordion' as the selected item. The main content area is titled 'Accordion' and includes a 'View on GitHub' link. It describes the component as a way to build vertically collapsing accordions using the Collapse JavaScript plugin. An example of an accordion is shown with a colorful illustration of people. Below this, the 'How it works' section explains that the accordion uses the 'collapse' plugin and mentions a media query for reduced motion. The 'Example' section provides instructions on how to use the component, including adding the '.show' class and removing the '.collapsed' class. On the right side of the main content, there is a 'On this page' section with links to 'How it works', 'Example', 'Flush', 'Always open', 'Accessibility', 'CSS', 'Variables', and 'See variables'.

<https://getbootstrap.com/docs/5.3/components/accordion/>

BASIC INTRODUCTION TO REACT JS

A screenshot of the Visual Studio Code editor interface. The title bar shows "App.js - reactjs-1 - Visual Studio Code". The Explorer sidebar on the left shows a project structure with folders "node_modules", "public", and "src". The "src" folder is expanded, showing files like "App.css", "App.js", "App.test.js", "index.css", "index.js", "logo.svg", "reportWebVitals.js", "setupTests.js", ".gitignore", "package-lock.json", "package.json", and "README.md". The "App.js" file is selected and open in the editor. The code in "App.js" is as follows:

```
4 function App() {  
5   return (  
6     <div className="App">  
7       <header className="App-header">  
8         <img src={logo} className="App-logo" alt="logo" />  
9         <p>  
10          Edit <code>src/App.js</code> and save to reload.  
11        </p>  
12        <a  
13          className="App-link"  
14          href="https://reactjs.org"  
15          target="_blank"  
16          rel="noopener noreferrer"  
17        >  
18          Learn React  
19        </a>  
20      </header>  
21    </div>  
22  );  
23 }  
24  
25 export default App;
```

The editor has a blue selection highlight on lines 9 through 19. The bottom of the editor shows the "TERMINAL" tab with the following output:

```
Note that the development build is not optimized.  
To create a production build, use npm run build.  
  
webpack compiled successfully
```

BASIC INTRODUCTION TO NODE JS



Downloads

Latest LTS Version: 20.11.1 (includes npm 10.2.4)

Download the Node.js source code or a pre-built installer for your platform, and start developing today.

LTS Recommended For Most Users	Current Latest Features	
 Windows Installer node-v20.11.1-x64.msi	 macOS Installer node-v20.11.1.pkg	 Source Code node-v20.11.1.tar.gz

Windows Installer (.msi)

Windows Binary (.zip)

macOS Installer (.pkg)

macOS Binary (.tar.gz)

Linux Binaries (x64)

Linux Binaries (ARM)

Source Code

32-bit	64-bit	ARM64
32-bit	64-bit	ARM64
64-bit / ARM64		
64-bit	ARM64	
64-bit		
ARMv7	ARMv8	
node-v20.11.1.tar.gz		

BASIC INTRODUCTION TO NODE JS



NodeJS was first introduced by Ryan Dahl in 2009.

NodeJS is a technology that can be used to run JavaScript code outside of a web browser.

NodeJS is a free and open-source project.



BASIC INTRODUCTION TO NODE JS



NodeJS popularized the "JavaScript Everywhere" paradigm, where we can create server-side applications using the JavaScript programming language.

This makes it so we only need to learn the JavaScript programming language to create web applications, for example, without needing to learn other programming languages like PHP or Java for the server-side web.

Currently, NodeJS is very popular and widely used in technology companies, especially to assist in frontend web development.

BASIC INTRODUCTION TO NODE JS



```
JS hello-world.js ●  
  
JS hello-world.js  
1 console.info("Hello World");
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS D:\Bahan Kuliah Del\Mata Kuliah\psw 2\2024\node-js\node-js-0> node hello-world.js  
Hello World  
PS D:\Bahan Kuliah Del\Mata Kuliah\psw 2\2024\node-js\node-js-0> █
```

```
JS hello-world.js JS module.mjs X  
  
JS module.mjs  
1 import os from "os";  
2  
3 console.info(os.platform());  
4 console.table(os.cpus());
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
>  
PS D:\Bahan Kuliah Del\Mata Kuliah\psw 2\2024\node-js\node-js-0> node "d:\Bahan Kuliah Del\Mata Kuliah\psw 2\2024\node-js\node-js-0\module.mjs"  
win32
```

(Index)	model	speed	times
0	'Intel(R) Core(TM) i7-8565U CPU @ 1.80GHz'	1992	[Object]
1	'Intel(R) Core(TM) i7-8565U CPU @ 1.80GHz'	1992	[Object]
2	'Intel(R) Core(TM) i7-8565U CPU @ 1.80GHz'	1992	[Object]
3	'Intel(R) Core(TM) i7-8565U CPU @ 1.80GHz'	1992	[Object]
4	'Intel(R) Core(TM) i7-8565U CPU @ 1.80GHz'	1992	[Object]
5	'Intel(R) Core(TM) i7-8565U CPU @ 1.80GHz'	1992	[Object]
6	'Intel(R) Core(TM) i7-8565U CPU @ 1.80GHz'	1992	[Object]
7	'Intel(R) Core(TM) i7-8565U CPU @ 1.80GHz'	1992	[Object]

```
PS D:\Bahan Kuliah Del\Mata Kuliah\psw 2\2024\node-js\node-js-0> █
```


BASIC INTRODUCTION TO NODE JS



REPL stands for Read Eval Print Loop.

That's the mechanism where a program can read code directly as it's typed, then execute it, display the result, and then repeat the process from the beginning.

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS
>
PS D:\Bahan Kuliah Del\Mata Kuliah\psw 2\2024\node-js\node-js-0> node
Welcome to Node.js v20.11.1.
Type ".help" for more information.
> 
```

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS
.save      Save all evaluated commands in this REPL session to a file

Press Ctrl+C to abort current expression, Ctrl+D to exit the REPL
>
PS D:\Bahan Kuliah Del\Mata Kuliah\psw 2\2024\node-js\node-js-0> node
Welcome to Node.js v20.11.1.
Type ".help" for more information.
> .help
.break      Sometimes you get stuck, this gets you out
.clear      Alias for .break
.editor      Enter editor mode
.exit        Exit the REPL
.help        Print this help message
.load        Load JS from a file into the REPL session
.save        Save all evaluated commands in this REPL session to a file

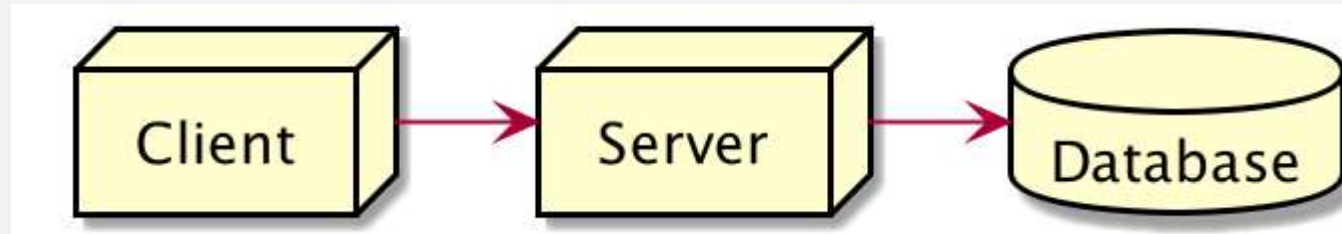
Press Ctrl+C to abort current expression, Ctrl+D to exit the REPL
> 
```

```
PS D:\Bahan Kuliah Del\Mata Kuliah\psw 2\2024\node-js\node-js-0> node
Welcome to Node.js v20.11.1.
Type ".help" for more information.
> function penjumlahan(a,b){
... return a+b;
... }
undefined
> var hasil = penjumlahan(2,3);
undefined
> console.log(hasil);
5
undefined
> 
```


BASIC INTRODUCTION TO NODE JS



When we create a web application, it is usually divided into three parts: the client, server, and database.

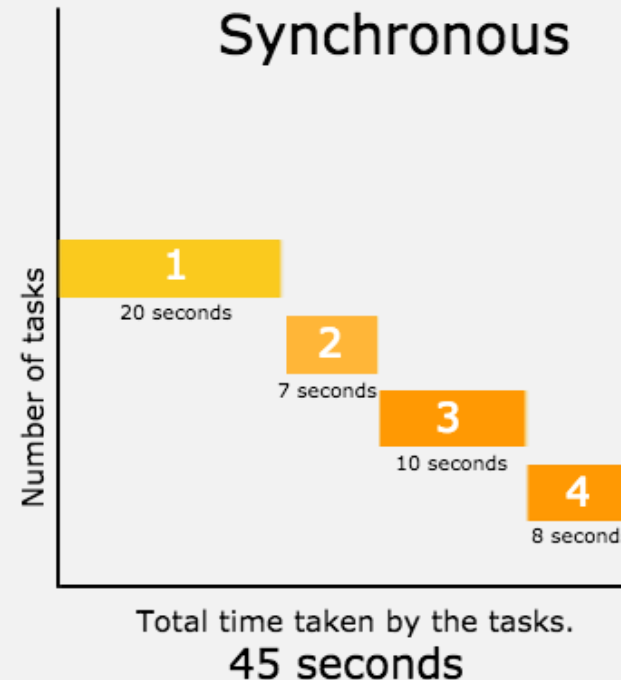


BASIC INTRODUCTION TO NODE JS



Synchronous refers to when our program code runs sequentially, and each step waits until the process is complete before moving on to the next step.

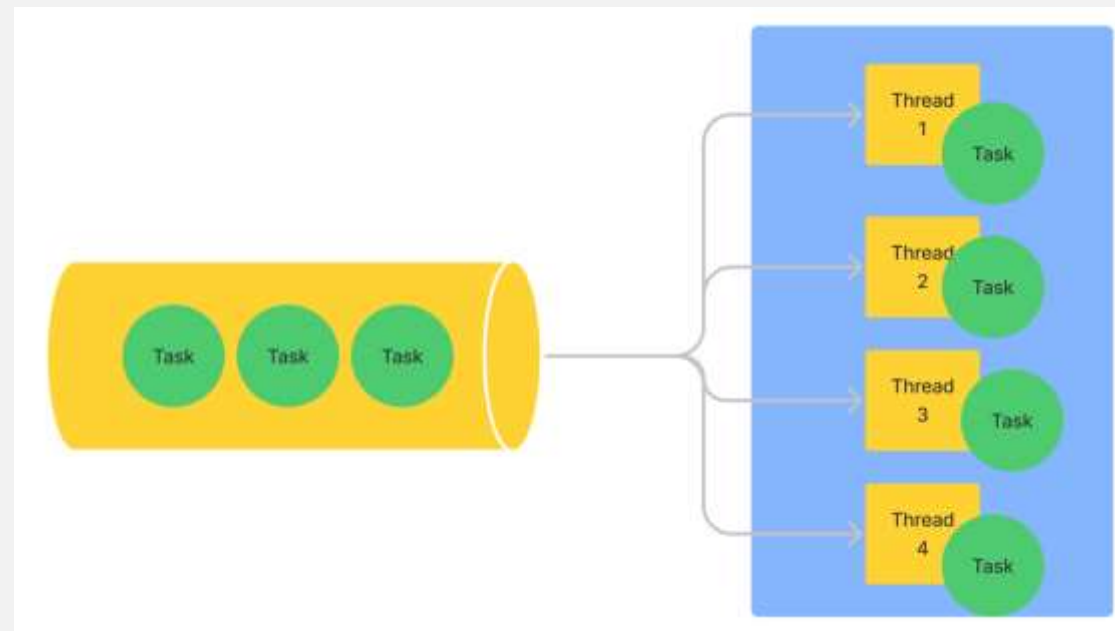
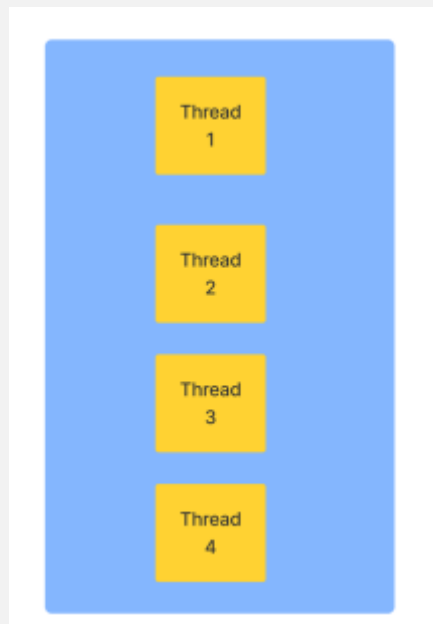
Asynchronous means that our program code runs, and we don't have to wait for the execution of that code to finish. We can move on to the next steps of the program code.



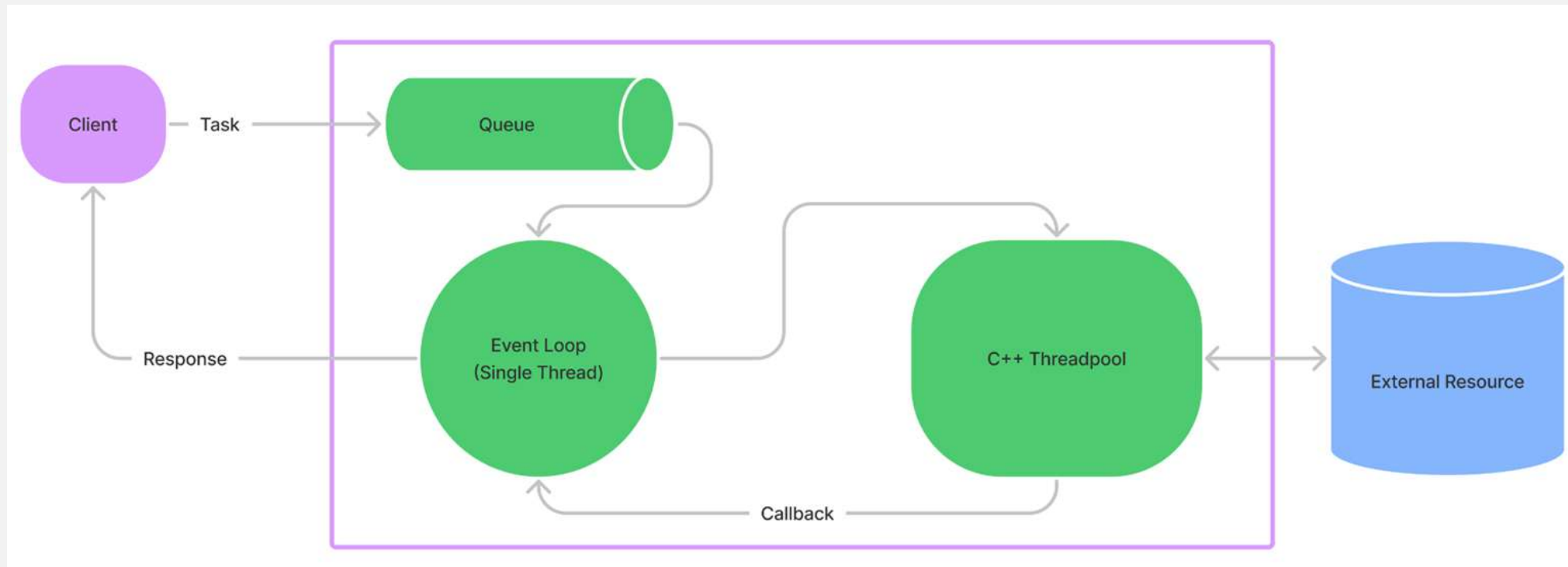
BASIC INTRODUCTION TO NODE JS



Threads are **lightweight processes commonly** created when developing applications.



BASIC INTRODUCTION TO NODE JS



BASIC INTRODUCTION TO NPM



♥ Pro Teams Pricing Documentation

npm

🔍 Search packages

Search

Sign Up

Sign In

About npm

npm, Inc. is a company founded in 2014, and was acquired by GitHub in 2020. npm is a critical part of the JavaScript community and helps support one of the largest developer ecosystems in the world.

npm is lots of things.

- npm is the package manager for [Node.js](#). It was created in 2009 as an [open source project](#) to help JavaScript developers easily share packaged modules of code.
- The npm Registry is a public collection of packages of open-source code for Node.js, [front-end web apps](#), [mobile apps](#), [robots](#), [routers](#), and countless other needs of the JavaScript community.
- npm is the command line client that allows developers to install and publish those packages.

<https://www.npmjs.com/about>



TERIMAKASIH