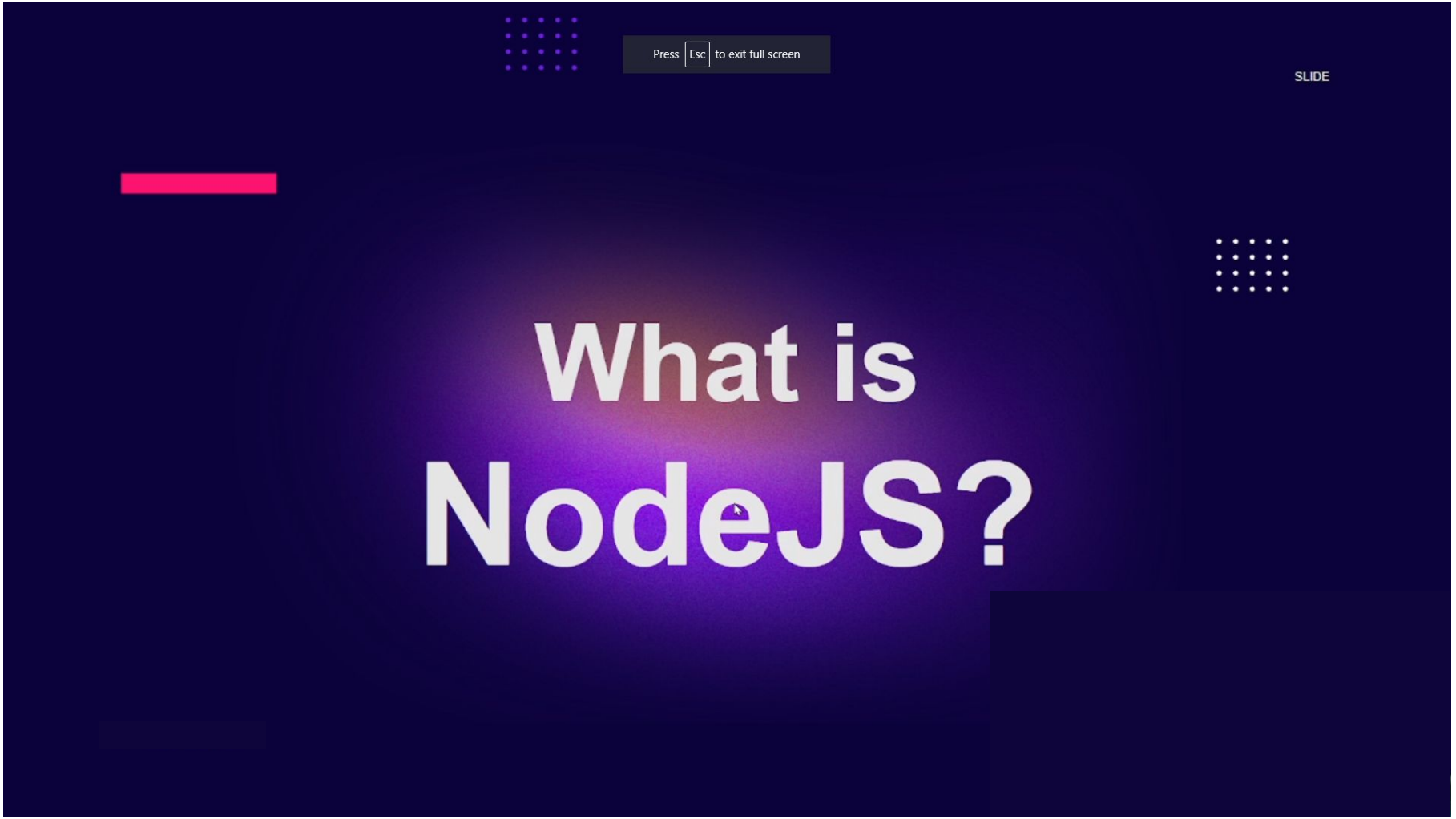
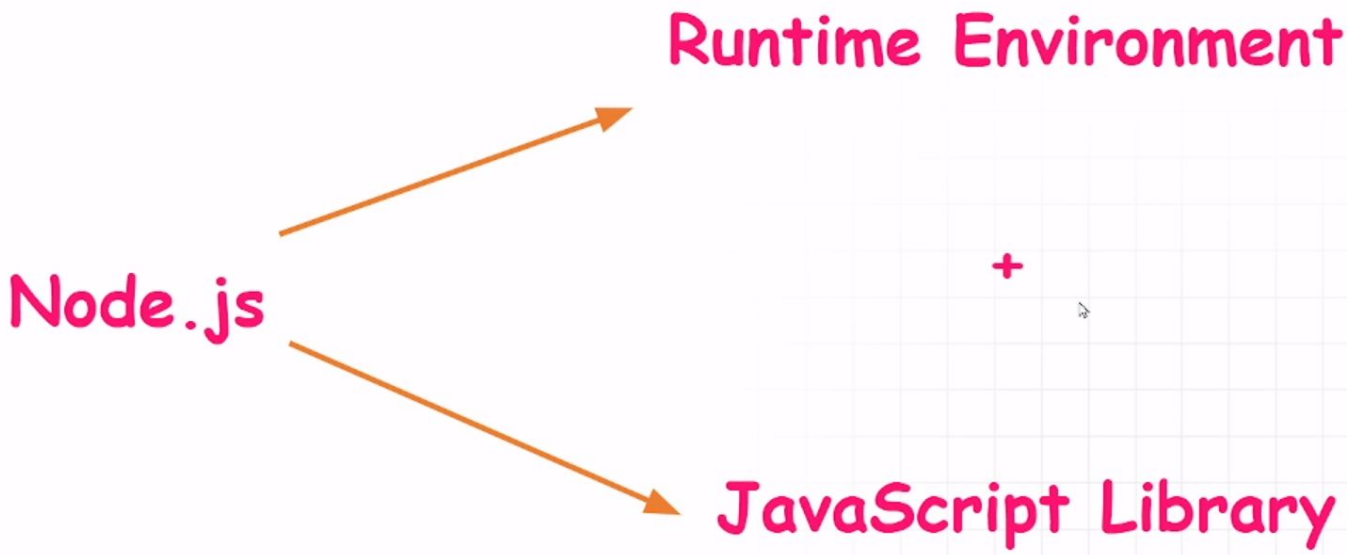
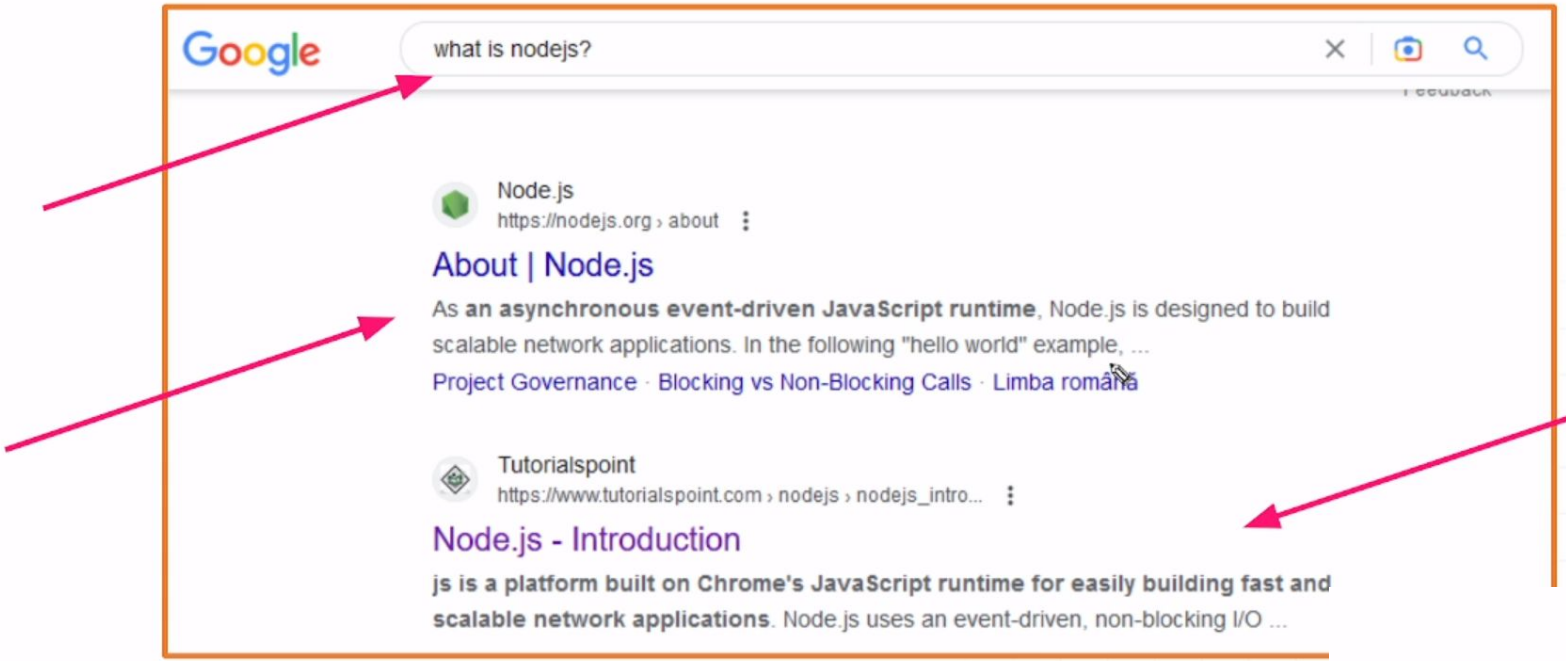
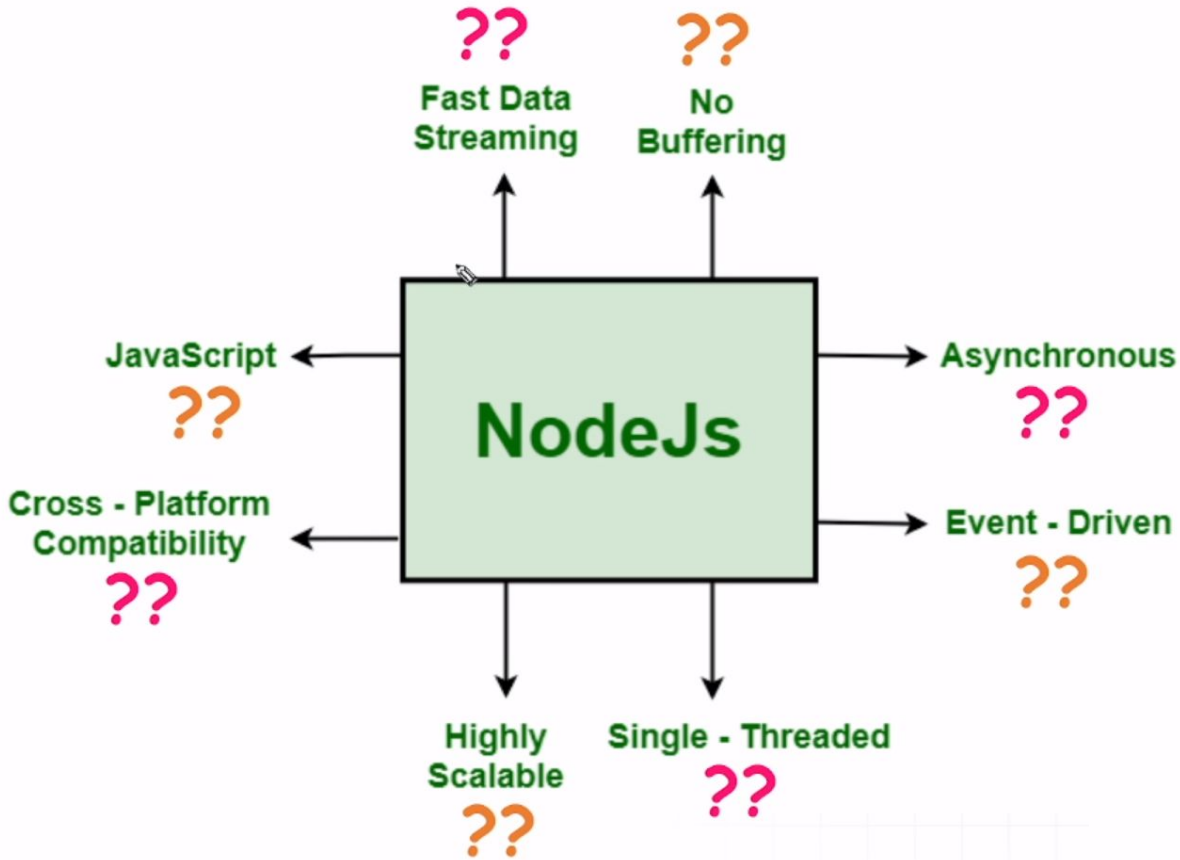

Press Esc to exit full screen
SLIDE
What is
NodeJS?


Runtime Environment
+
Node.js
JavaScript Library

Node JS

- Free
- Uses JavaScript on the server
- An open source server environment
- Runs on various platforms (Windows, Linux, Unix, Mac OS X, etc.)


Google
what is nodejs?
Node.js
https://nodejs.org › about
About | Node.js
As an asynchronous event-driven JavaScript runtime, Node.js is designed to build scalable network applications. In the following "hello world" example, ...
Project Governance · Blocking vs Non-Blocking Calls · Limba română
Tutorialspoint
https://www.tutorialspoint.com › nodejs › nodejs_intro...
Node.js - Introduction
js is a platform built on Chrome's JavaScript runtime for easily building fast and scalable network applications. Node.js uses an event-driven, non-blocking I/O ...


??
Fast Data Streaming
??
No Buffering
JavaScript
??
Asynchronous
??
NodeJs
Cross - Platform Compatibility
??
Event - Driven
??
Highly Scalable
??
Single - Threaded
??

Features

- Very Fast
- No Buffering
- Asynchronous and Event Driven
- Single Threaded but Highly Scalable

Who uses?

- eBay
- General Electric
- GoDaddy
- Microsoft
- PayPal
- Uber
- Wikipins
- Yahoo!

Do you watch movie on ↓

NETFLIX

WELL!!
It's also using NodeJS

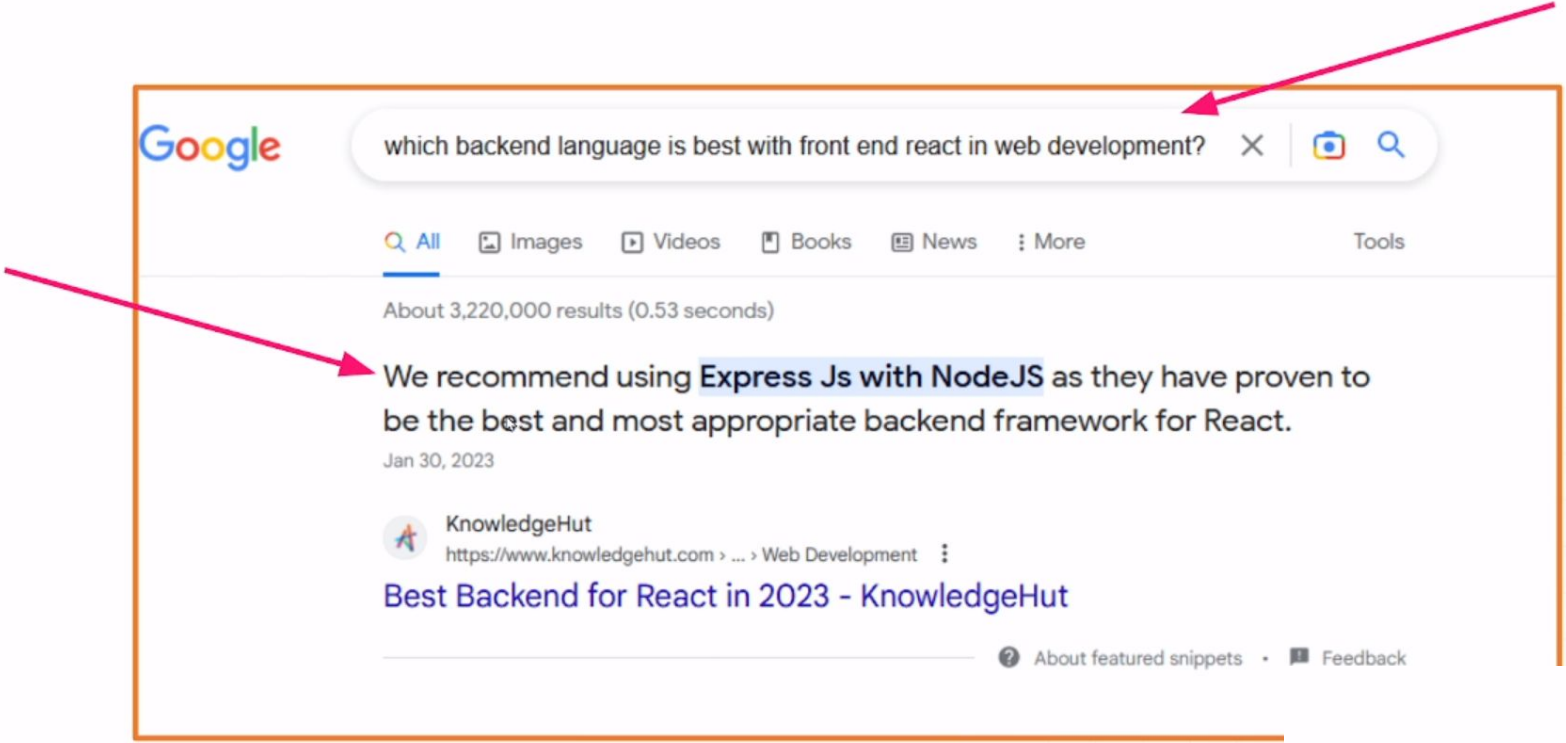
🤣 🤣 🤣

Where to use?

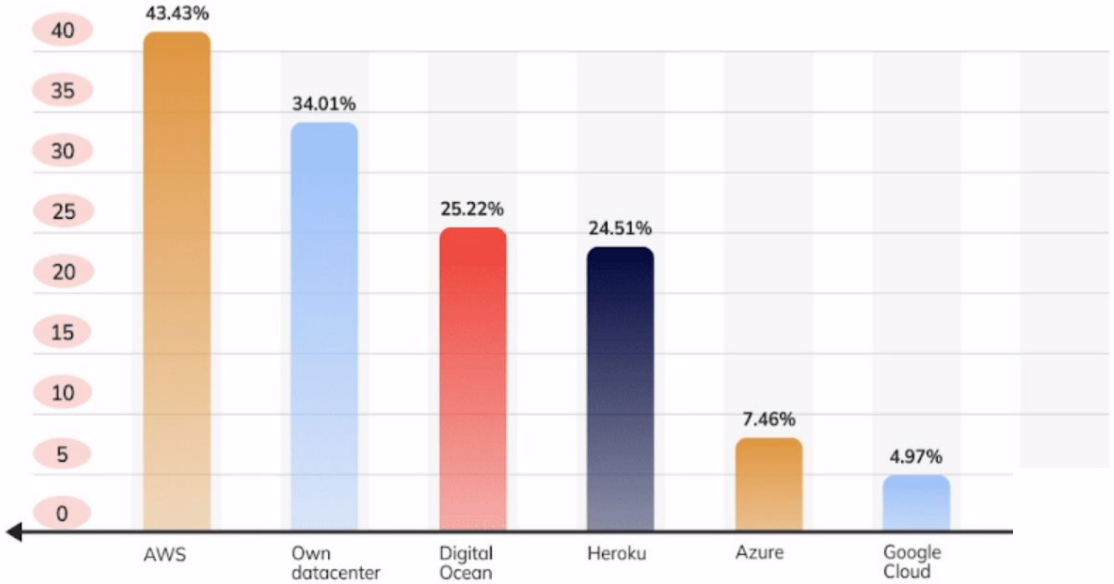
- I/O bound Applications
- Data Streaming Applications
- Data Intensive Real-time Applications (DIRT)
- JSON APIs based Applications
- Single Page Applications

NodeJS এর এতো alternatives থাকলে আমরা NodeJS ই কেন শিখবো?

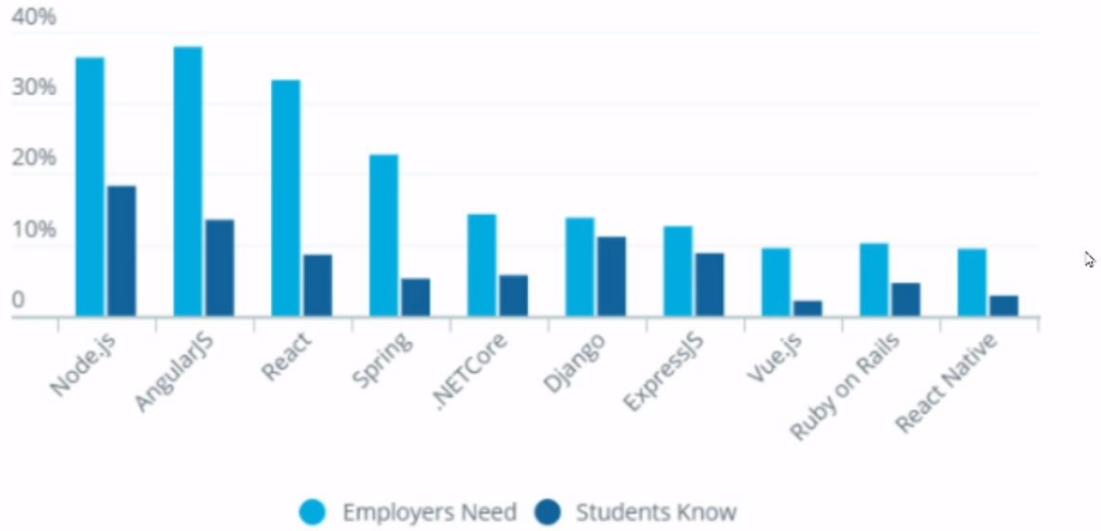
Why???


Google
which backend language is best with front end react in web development?
All
Images
Videos
Books
News
More
Tools
About 3,220,000 results (0.53 seconds)
We recommend using Express Js with NodeJS as they have proven to be the best and most appropriate backend framework for React.
Jan 30, 2023
KnowledgeHut
https://www.knowledgehut.com › ... › Web Development
Best Backend for React in 2023 - KnowledgeHut
About featured snippets
Feedback

Where can you run NodeJS apps?



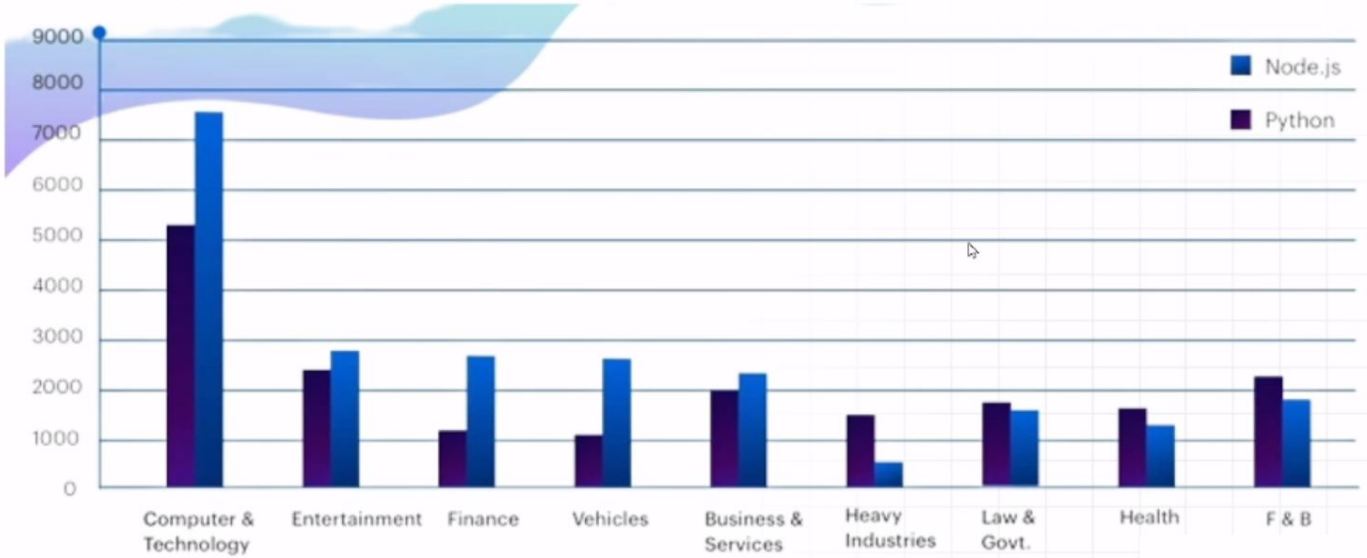
Employers need vs Our learning



তাই বলে কি NodeJS সব ধরনের website এর জন্যই best?

No

NodeJs vs Python : Website categories



What is express & why should you learn express is?

What is ExpressJS?

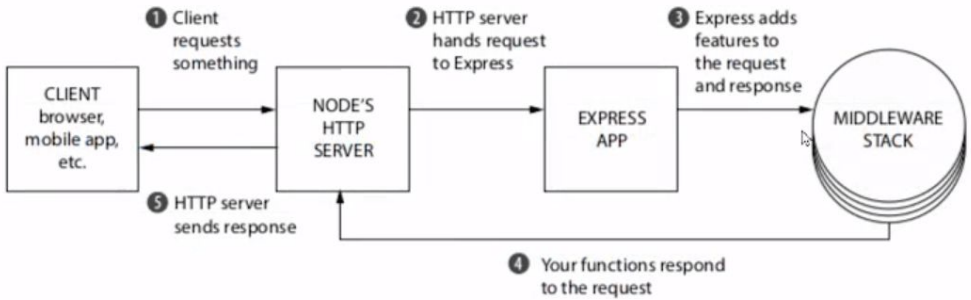
- Express is a node js web application framework that provides broad features for building web and mobile applications.
- It is used to build a single page, multipage, and hybrid web application.
- It's a layer built on the top of the Node js that helps manage servers and routes.

NodeJS vs ExpressJs

Feature	ExpressJS	NodeJS
Purpose	Web framework	JavaScript runtime
Built on	NodeJS	Chrome's V8 JavaScript engine
Focus	Server-side development	General-purpose JavaScript programming
Key features	Routing, middleware, templating	Non-blocking I/O, event-driven programming
Use Cases	Building web applications, RESTful APIs	Server-side scripting, creat… line tools

What is ExpressJS?

Node.js APIs can get complex and writing how to handle a single request can end up being over 50 lines of code. Express makes it easier to write Node.js web applications.



What is ExpressJS Middleware?

- Middleware in Express are functions that come into play **after the server receives the request and before the response is sent to the client**.
- They are arranged in a chain and are called in sequence.

What is ExpressJS Middleware?

We can use middleware functions for different types of processing tasks required for fulfilling the request like

- database querying,
- making API calls,
- preparing the response, etc,
- and finally calling the next middleware function in the chain.

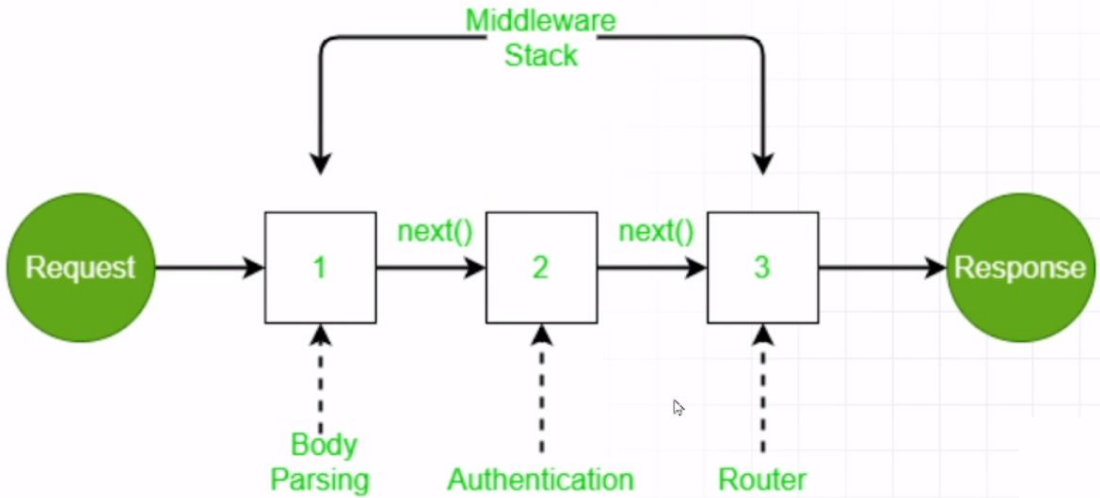
What is ExpressJS Middleware?

- We call app.use() to add a middleware function to our Express application.
- Under the hood, when we call app.use(), the Express framework adds our middleware function to its internal middleware stack.
- Express executes middleware in the order they are added, so if we make the calls in this order:

Express will first execute function1 and then function2.

What is ExpressJS Middleware?

The **next()** function in the express is responsible for calling the next middleware function if there is one.



What is ExpressJS Middleware?

Some middleware functions do not require the next function to be called, because they are specifically designed to handle the request and response on their own.

app.use(cors());

the cors() middleware function is used to enable Cross-Origin Resource Sharing (CORS) for an Express application, and it sets the appropriate headers in the response to allow cross-origin requests.

app.use(express.json());

the express.json() middleware function is used to parse incoming JSON requests.

Why ExpressJS?

Providing a robust set of features for building web app's:

- Configuration
- Routing
- Middleware
- View & Templates
- Session
- Security

The reason behind creating an express framework for nodejs is

Fast, asynchronous, unopinionated, minimalist

Express generally requires less code than building an application with the Node.js HTTP module alone.

Create server without ExpressJS

```
const http = require('http');

const server = http.createServer((req, res) => {
  res.writeHead(200, {'Content-Type': 'text/plain'});
  res.write('Hello, world!');
  res.end();
});

server.listen(3000, () => {
  console.log('Server is running on port 3000');
});
```

Create server with ExpressJS

```
const express = require('express');

const app = express();

app.get('/', (req, res) => {
  res.send('Hello, world!');
});

app.listen(3000, () => {
  console.log('Server is running on port 3000');
});
```

Routing without ExpressJS

```
const http = require('http');

const hostname = 'localhost';
const port = 3000;

const server = http.createServer((req, res) => {
  // Parse the request URL to get the path
  const path = req.url;

  // Route the request based on the path
  if (path === '/') {
    res.statusCode = 200;
    res.setHeader('Content-Type', 'text/plain');
    res.end('Hello, world!');
  } else if (path === '/about') {
    res.statusCode = 200;
    res.setHeader('Content-Type', 'text/plain');
    res.end('This is the about page');
  } else {
    res.statusCode = 404;
    res.setHeader('Content-Type', 'text/plain');
    res.end('Page not found');
  }
});

server.listen(port, hostname, () => {
  console.log(`Server running at http://${hostname}:${port}/`);
});
```

Routing with ExpressJS

```
const express = require('express');

const app = express();

app.get('/', (req, res) => {
  res.send('Hello, world!');
});

app.get('/about', (req, res) => {
  res.send('This is the about page');
});

app.use((req, res) => {
  res.status(404).send('Page not found');
});

app.listen(3000, () => {
  console.log('Server is running on port 3000');
});
```

Is ExpressJS better than other frameworks?

In recent years, ExpressJS has become the de facto choice for building web applications as it is fast, flexible, and minimal.

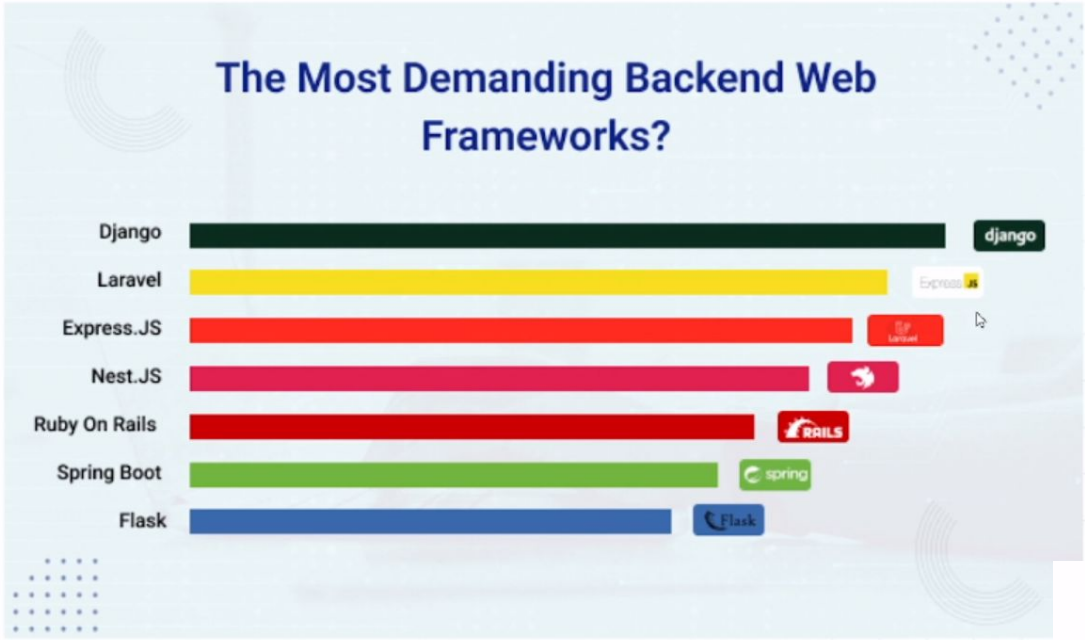
ref: SOLUTELABS

Is ExpressJS good for larger applications?

ExpressJS has a simple and intuitive API that makes it easy to learn and use, even for beginner web developers. ExpressJS is designed to be scalable and efficient, making it suitable for building large and complex web applications.

ref: Intellipaat

In 2023



ref: GrafferID

Companies using ExpressJS

See more: expressjs

Thank you!