

# Node.js Intro

Zia Khan

Download Code:

[https://drive.google.com/drive/folders/1LDniUXgY52ZKE6TQ\\_Kf111P4gmUHCpil?usp=sharing](https://drive.google.com/drive/folders/1LDniUXgY52ZKE6TQ_Kf111P4gmUHCpil?usp=sharing)

# Install Node.js

<https://nodejs.org/en/download/current/>

Install Version 18.10.0+

node -v

# Hello Node.js

```
mkdir zia
```

```
cd zia
```

```
npm init
```

```
// Read https://nodejs.org/api/esm.html#modules-ecmascript-modules
```

Open VSCode and create File hello.mjs

```
console.log("Hello from Zia");
```

```
node hello.mjs
```

## addTwo.mjs File

```
// addTwo.mjs
```

```
function addTwo(num) {
```

```
  return num + 2;
```

```
}
```

```
export { addTwo };
```

## app.mjs File

```
// app.mjs
```

```
import { addTwo } from './addTwo.mjs';
```

```
// Prints: 6
```

```
console.log(addTwo(4));
```

## .mjs versus .js

In this course upto now, we've used .js extensions for our module files, but from now on you may see the .mjs extension used instead. V8's documentation recommends this ( <https://v8.dev/features/modules#mjs> ) , The reasons given are:

- It is good for clarity, i.e. it makes it clear which files are modules, and which are regular JavaScript.
- It ensures that your module files are parsed as a module by runtimes such as Node.js, and build tools such as Babel.
- On the Web, the file extension doesn't really matter, as long as the file is served with the JavaScript MIME type text/javascript. The browser knows it's a module because of the type attribute on the script element.

# Http Simple Server in Node.js

```
mkdir simpleServer
```

```
cd simpleServer
```

```
npm init
```

```
npm install express
```

```
// Check the node_modules directory
```

Open VSCode and create File index.mjs

```
node index.mjs
```

# index.mjs

```
//index.mjs
```

```
import express from 'express';
```

```
const app = express();
```

```
app.get('/',(req,res) => {  
  res.send('Zia Responding from HTTP Server');  
})
```

```
const PORT = 5001;
```

```
app.listen(PORT,() => {  
  console.log(`Running on PORT ${PORT}`);  
})
```