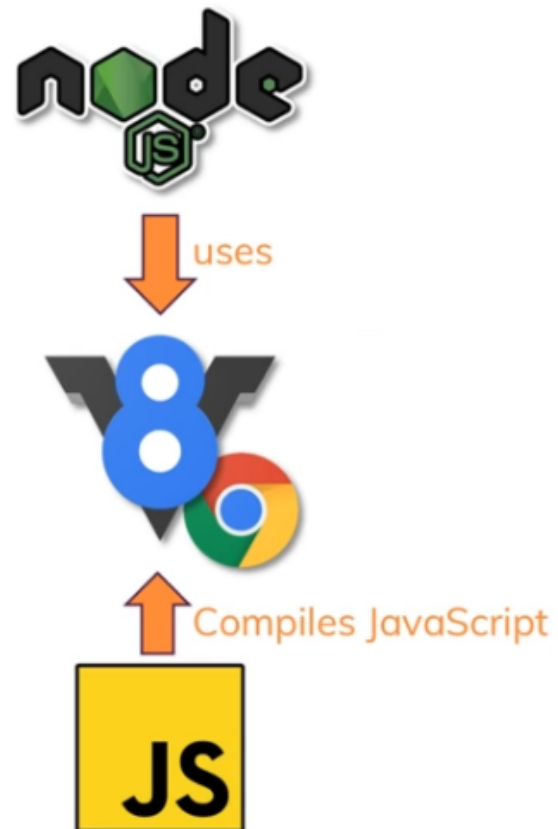
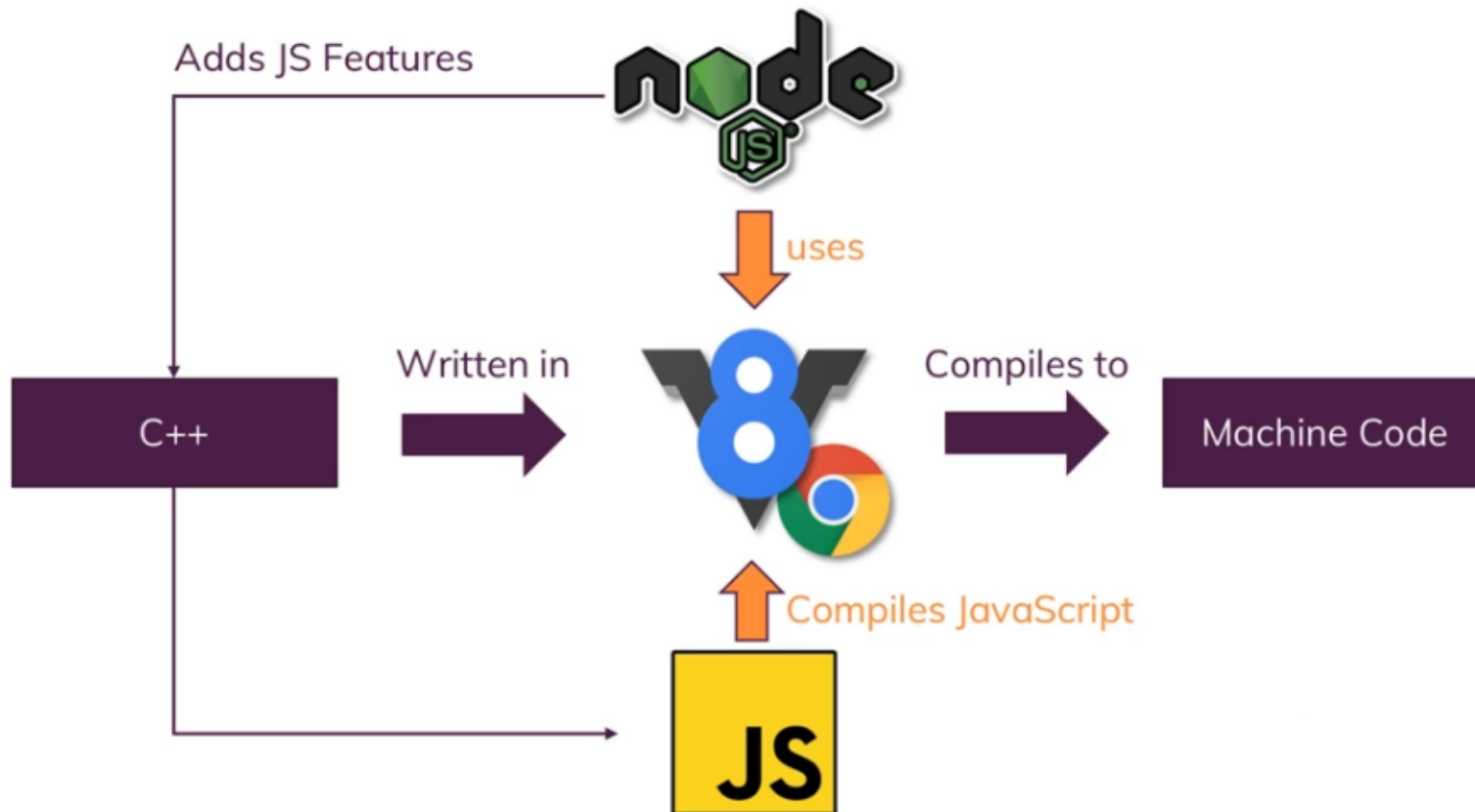


by: Zaynab AL-Ariny
zenab_khalf@eng.sohag.edu.eg

What Is Nodejs?

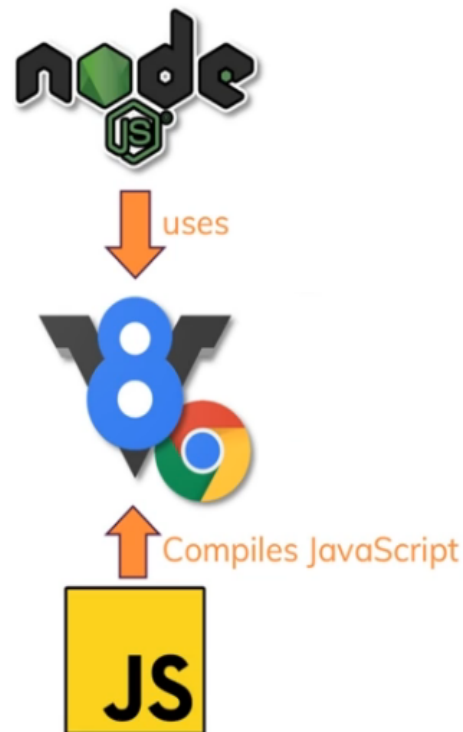


What Is Nodejs?



What Is Nodejs?

- a server-side platform built on Google Chrome's JavaScript Engine (V8 Engine).
- Allows you to build scalable network applications using JavaScript on the server-side.
- Nodejs = Runtime Environment + JavaScript Libraries !



Installation

- Visit nodejs.org
- Download and install ...

Node.js® is a JavaScript runtime built on [Chrome's V8 JavaScript engine](#).

#BlackLivesMatter

New security releases to be made available January 4, 2021

Download for macOS (x64)

14.15.2 LTS

Recommended For Most Users

15.4.0 Current

Latest Features

[Other Downloads](#) | [Changelog](#) | [API Docs](#) [Other Downloads](#) | [Changelog](#) | [API Docs](#)

Or have a look at the [Long Term Support \(LTS\) schedule](#).

- Check that node is installed successfully.

```
apple@apples-MacBook-Pro ~ % node -v  
v12.16.3
```

Use node

- Enter the interactive mode:

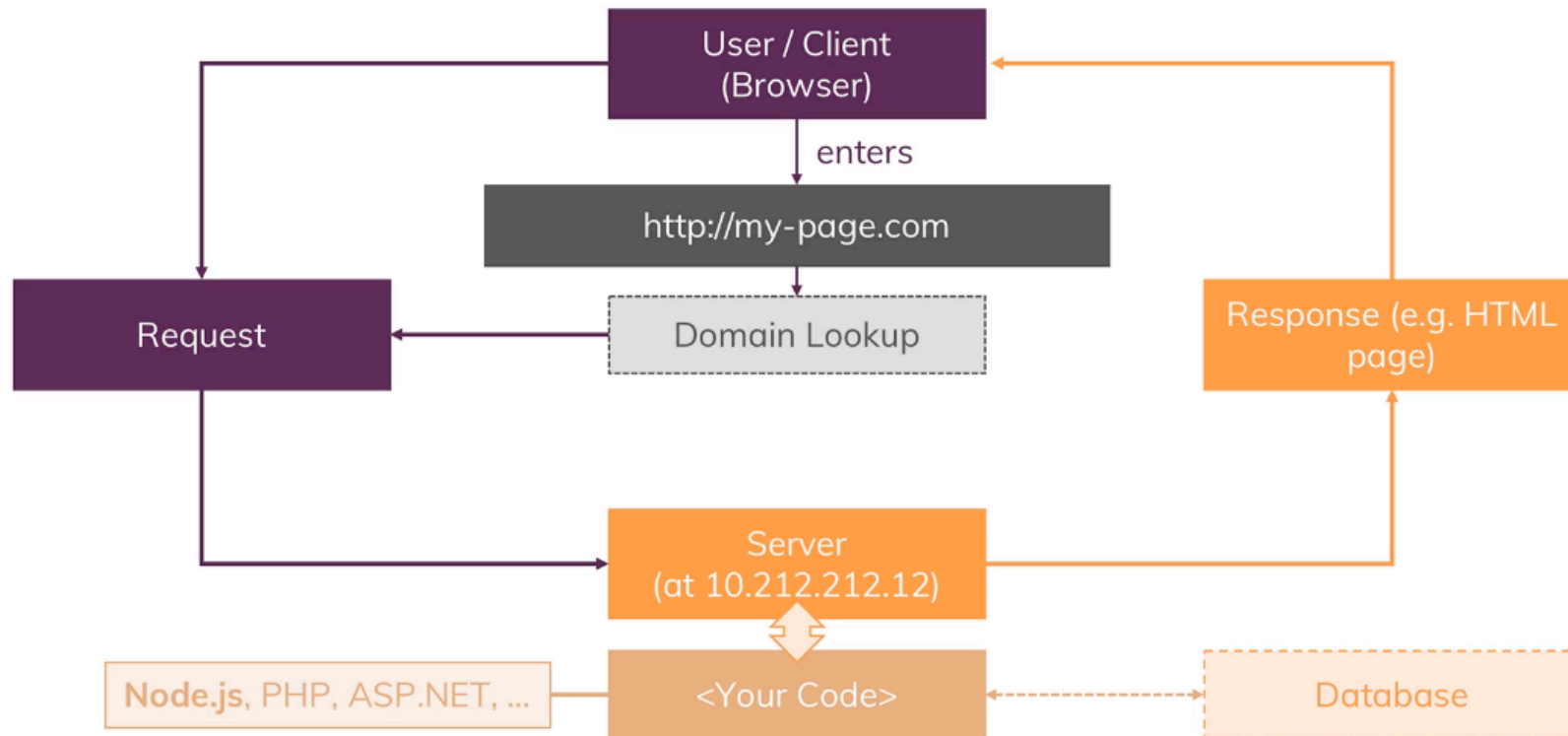
```
apple@apples-MacBook-Pro Documents % node
Welcome to Node.js v12.16.3.
Type ".help" for more information.
> console.log("Hello from Node.js !")
Hello from Node.js !
```

```
myFirstApp.js
1 console.log("Hello from Node.js!");
2
```

- Execute js file with node:

```
apple@apples-MacBook-Pro Documents % node myFirstApp.js
Hello from Node.js!
```

How web works?



How web works?

- http

Hyper Text Transfer Protocol

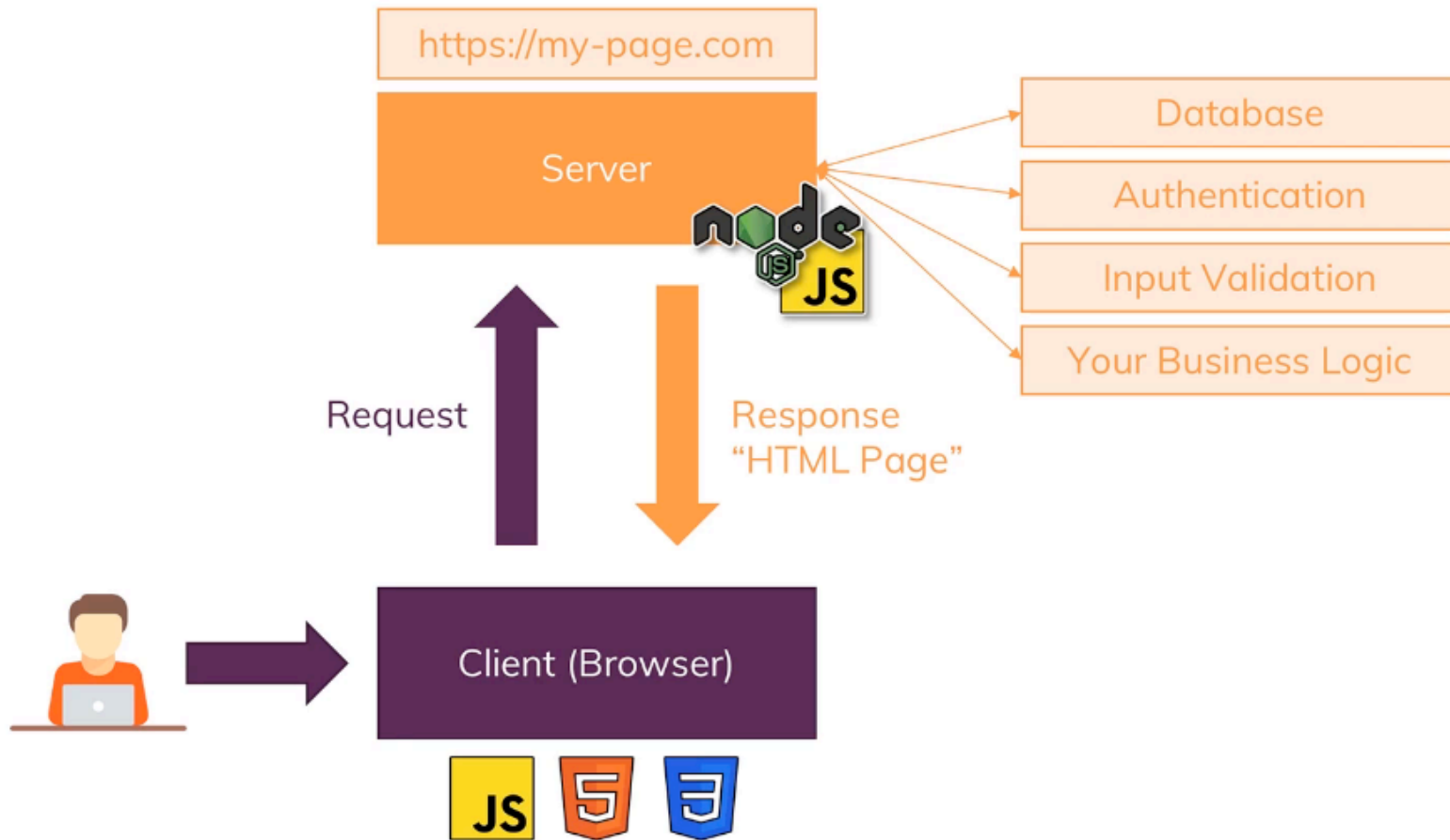
A Protocol for Transferring Data which is understood by Browser and Server

- https

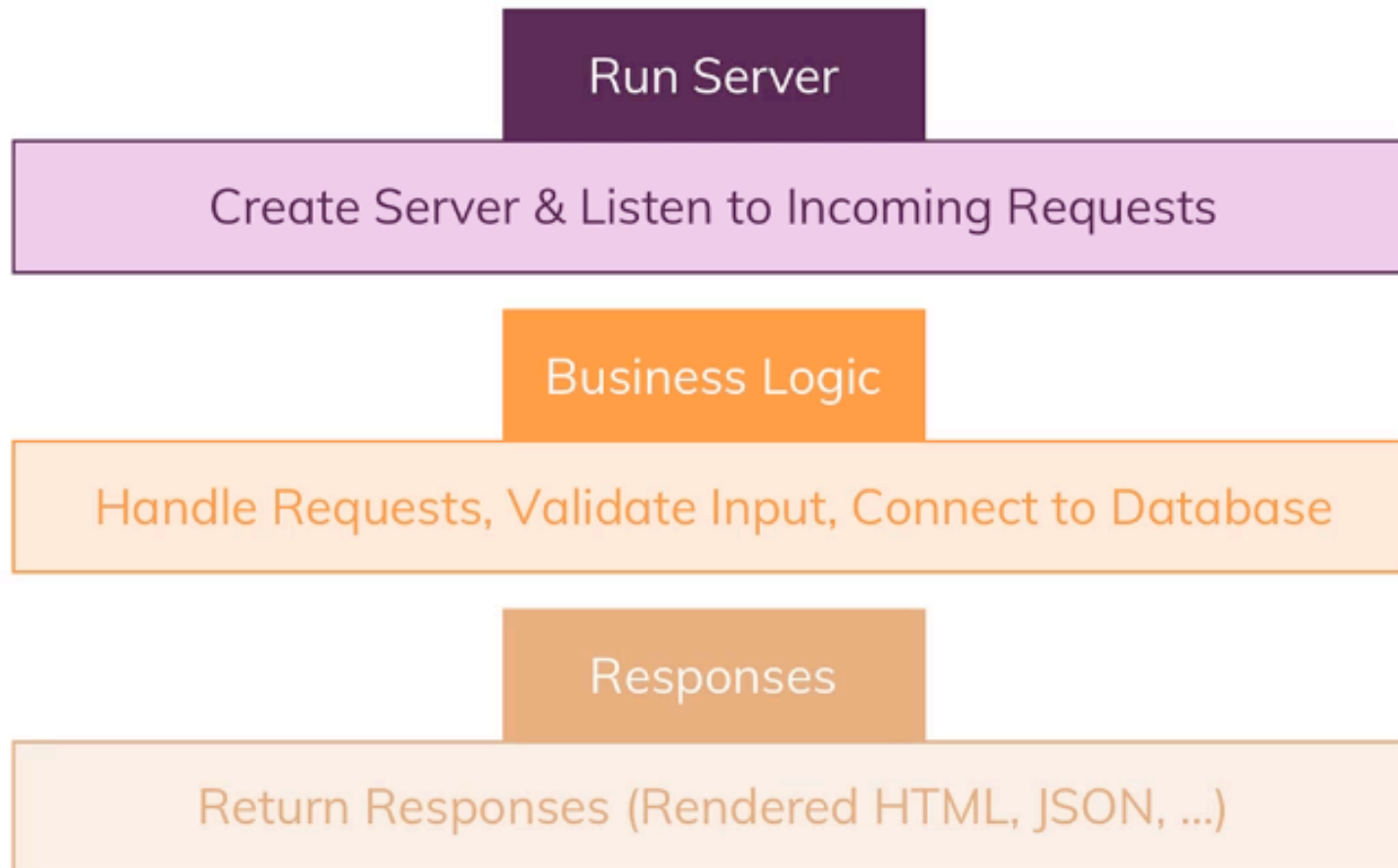
Hyper Text Transfer Protocol Secure

HTTP + Data Encryption (during Transmission)

JS on the Server

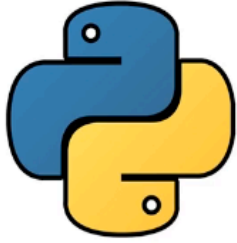


Node.js Role in Web Development



Why Node.js?

- Alternatives:



And More (Ruby, ASP.NET, ...)

- So, Why Node.js?
 - Very Fast.
 - Single Threaded but Highly Scalable.
 - No Buffering.
 - Asynchronous and EventDriven.
 - JS platform !

Create a Node server

- Write your server:

```
app.js
const http= require('http')

server = http.createServer((req, resp)=>{
  console.log(req);
})

server.listen(3000);
```

- Run the server:

```
% node app.js
```

- go to <http://localhost:3000>

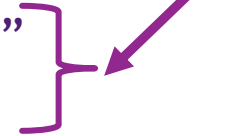
Blocking and non-Blocking code

- IF I need to write code for reading file content:

Blocking Code

1. Read file from File system, set equal to “contents”
2. Print contents
3. Do something else ...

Stop process until complete



Non-Blocking Code

1. Read file from File system, set equal to “contents”

whenever you're complete, print the contents

2. Do something else ...

This is a callback!



Blocking and non-Blocking code

- IF I need to write code for reading file content:

```
var contents = fs.readFileSync("/etc/hosts");  
console.log(contents);  
console.log("Doing something else");
```

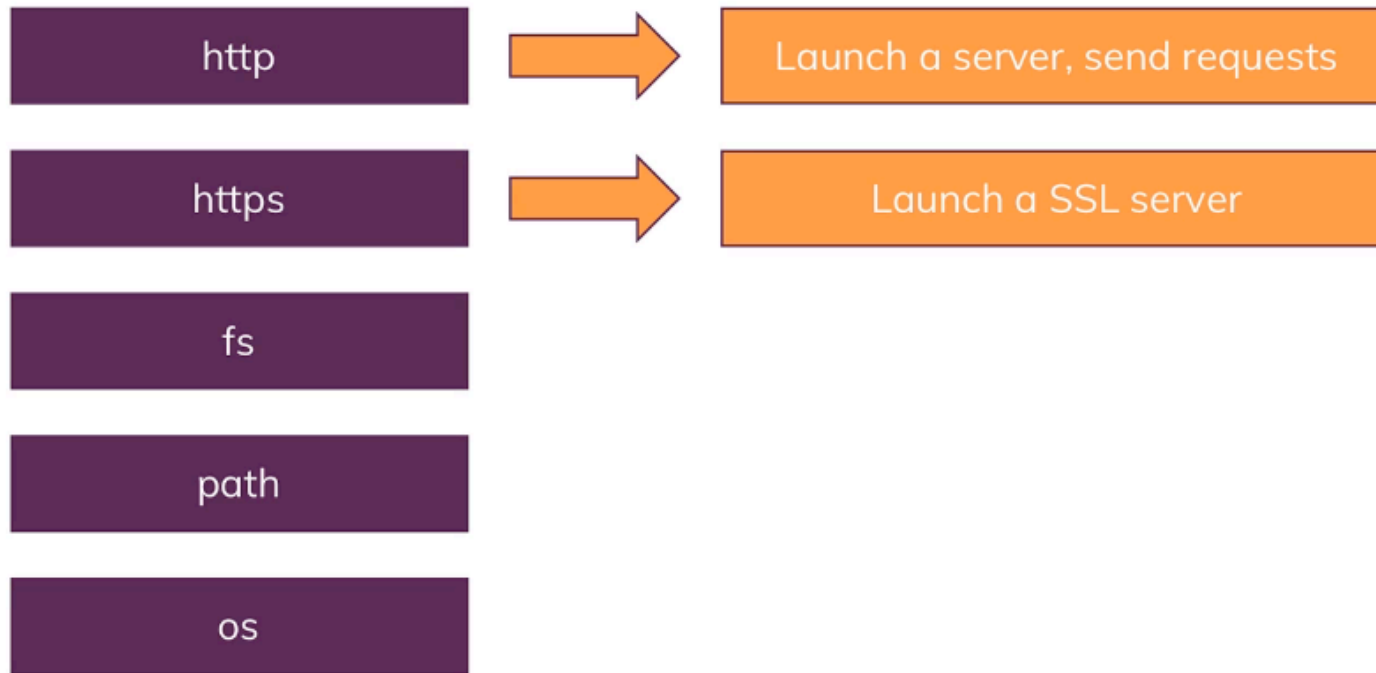
Stop process until complete

```
fs.readFile("/etc/hosts", function(err, contents){  
  console.log(contents);  
});  
console.log("Doing something else");
```

This is a callback!

Node Modules

- Import Library to your code.
- Core modules:



Node Modules

- load library once to your project.
- If you load it again will return the same object.

look in same directory

look in parent directory

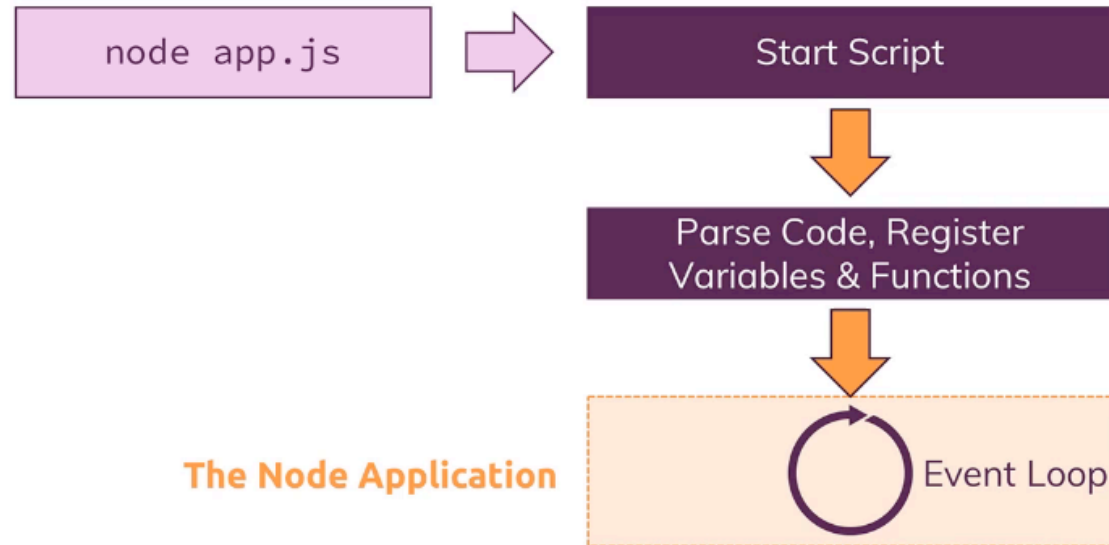
```
var library= require("./library_path");  
var library= require("../library_path");  
var library= require("/home/username/library_path");  
var library= require("library_path");
```

Search in node_modules directories

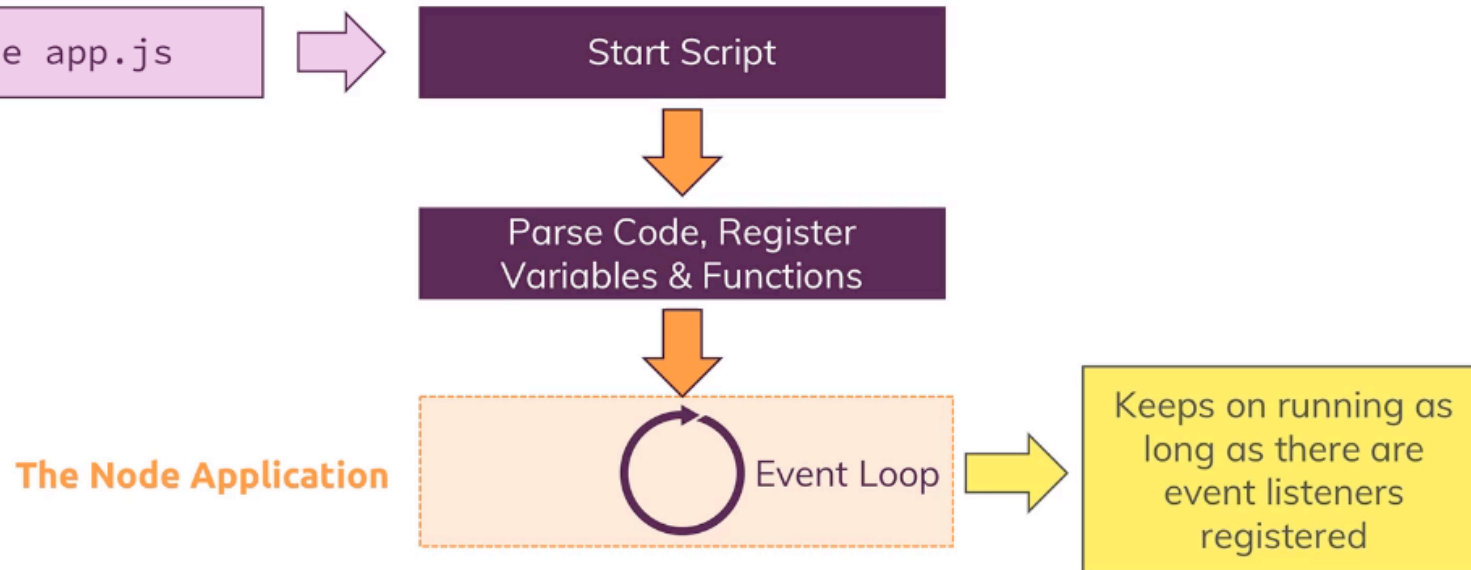
- Modules must be exported:

```
module.exports = myModule;
```


Program Lifecycle



Program Lifecycle



Event

- Many objects in Node emit events.
- You can create custom event emitters

```
var EventEmitter = require("events").EventEmitter;  
  
var logger = new EventEmitter();  
logger.on("error", function(message){  
    console.log('ERR: ' + message);  
});  
});  
  
logger.emit("error", "test message");
```

Load events module

Register callback for
event and named event
with name "error"

Fire "error" event

Event Loop

