

FULL STACK

Working with NPM and Node.js



You Already Know

Course(s):

Node.js Absolute Beginners Guide-Learn Node from Scratch



Recap

- Install Tools
 - Course Introduction
 - Install Node
 - Install Git bash and Sublime Text Editor
- Explain what is Node.js
 - Run a Console Log Node Test
 - Explaining the Node Server Code
 - File System and Index Page
 - Create Server Control Status Messages



- Create a Basic Website with Node and Bootstrap
 - Create a Bootstrap Navbar
 - Create an About and Resume Page
 - Add a Bootstrap Jumpbotron
- Push Node Website to Heroku Web Hosting
 - Generate SSH Keys
 - Initialize Git and GitHub
 - Create a package.json file



A Day in the Life of a MEAN Full Stack Developer

Joe has to debug a customized Node application.

In this lesson, we will learn how to solve this real-world scenario to help Joe complete his task effectively and quickly.



Learning Objectives

By the end of this lesson, you will be able to:

- 👁 Set up Node.js
- 👁 Download and setup VSC
- 👁 List features of Node.js
- 👁 Illustrate the importance of NPM
- 👁 Explain the file system in NPM and nodes



FULL STACK

Introduction to Node.js

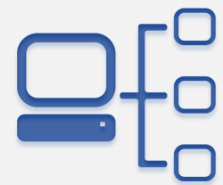
What Is Node.js?



Node.js runs on Chrome's V8 JavaScript engine.



It is a server-side JavaScript environment.



It is a high-performing application framework.



Node.js uses an event-driven, non-blocking I/O model that makes it lightweight and efficient.



Node.js applications simply deliver the output in chunks.

What Is Chrome V8?

V8 is an open source JavaScript Engine developed by Google for its web browser, Chrome.

V8 compiles JavaScript to native machine code.



The V8 engine is extremely fast.

V8 focuses on the web, so it works well with HTTP, DNS, and TCP.

Application of Node.js

Using Node.js, you can create:

DNS server

Static file server

Web chat applications like Gtalk

HTTP server

TCP server

Online games and collaboration tools

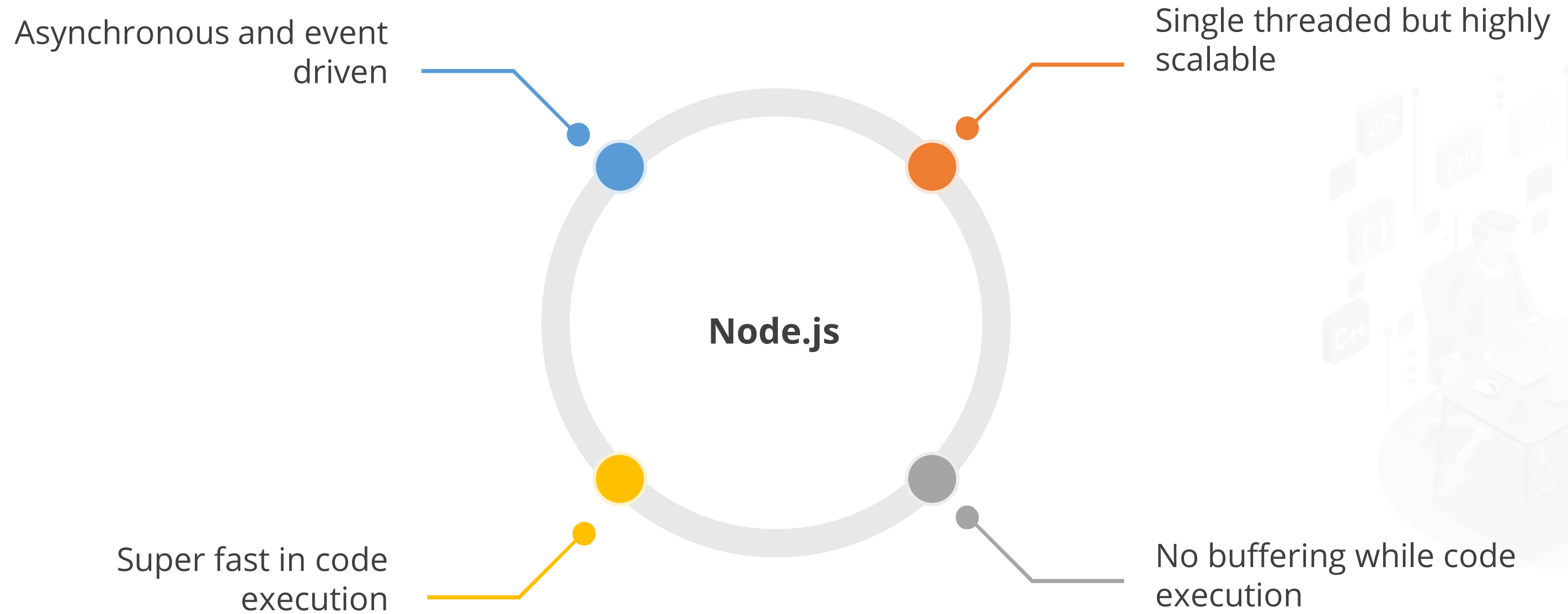
Node.js can also:

Share code between the browser and backend

Manage the load without adding additional CPU cycles, when there is a high-level concurrency

Interact with MySQL

Features of Node.js



Setting up Node.js



Duration: 15 min.

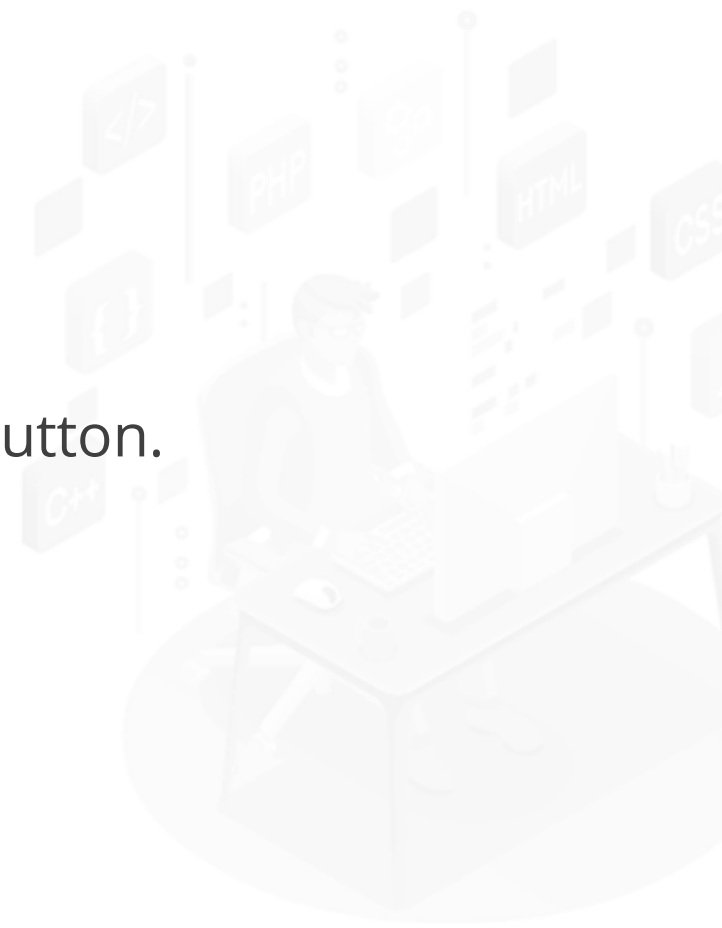
Problem Statement:

You are given a project to set up Node.js on your local system.

ASSISTED PRACTICE

Assisted Practice: Guidelines to Set up Node.js

1. Download Node.js.
2. Click the Run button on the first screen to begin the installation.
3. Click the Next button to continue with the installation.
4. Accept the license agreement and click on the Next button.
5. Choose the location where Node.js needs to be installed and then click on the Next button.
6. Accept the default components and click on the Next button.
7. Click the Install button to start the installation.
8. Click the Finish button to complete the installation.



Download and Set Up VSC



Duration: 15 min.

Problem Statement:

You are given a project to download and setup Visual Studio Code (VSC) on your local system.

ASSISTED PRACTICE

Assisted Practice: Guidelines to Download and Set up VSC

1. Download Visual Studio Code (VSC).
2. Accept the license agreement and click on the Next button.
3. Select additional tasks and then click on the Next button.
4. Click the Install button to start the installation.
5. Click the Finish button to complete the installation.



FULL STACK

Explore NPM

What Is NPM?

NPM stands for Node Package Manager, which is a package manager for JavaScript programming language.

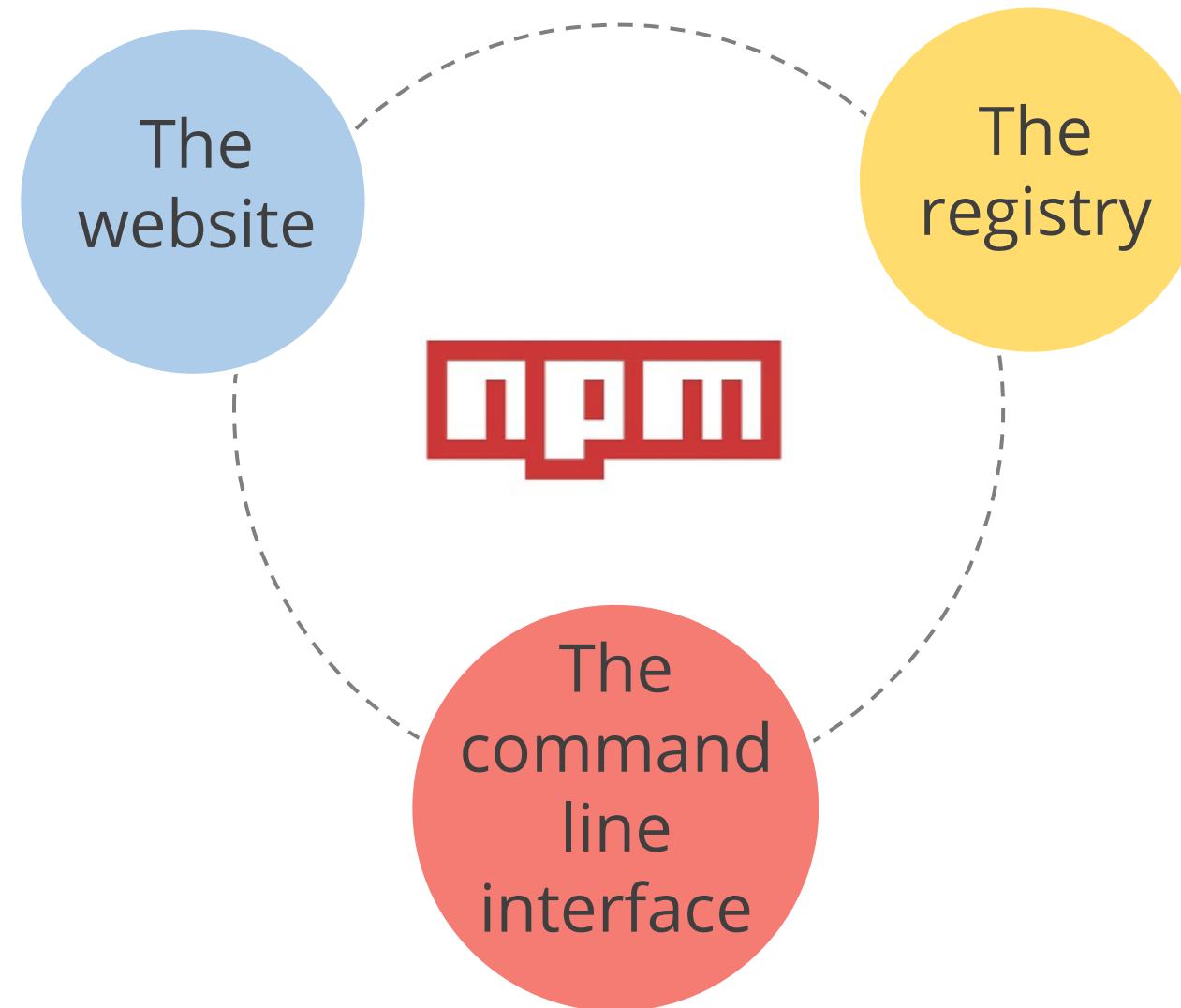


It is used as the default package manager for the JavaScript runtime environment, Node.js.

Components of NPM

NPM consists three distinct components:

The website helps to discover packages, set up profiles, and manage other aspects of npm.



The registry is a large public database of JavaScript software and the meta-information surrounding it.

The CLI runs from a terminal and is how most developers interact with npm.

Why NPM

NPM is the package manager for Node.js.

- It helps in installing various packages and resolving their dependencies.
- It greatly helps with node development.
- It helps to install various modules for web development.
- It also has a command-line tool that lets us easily install, manage, and run projects.

Uses of NPM

- Adapting packages of code for apps or incorporating packages as they are
- Downloading standalone tools so that they can be used right away
- Running packages without using [npm](#)
- Sharing code with any npm user, anywhere
- Restricting code to specific developers



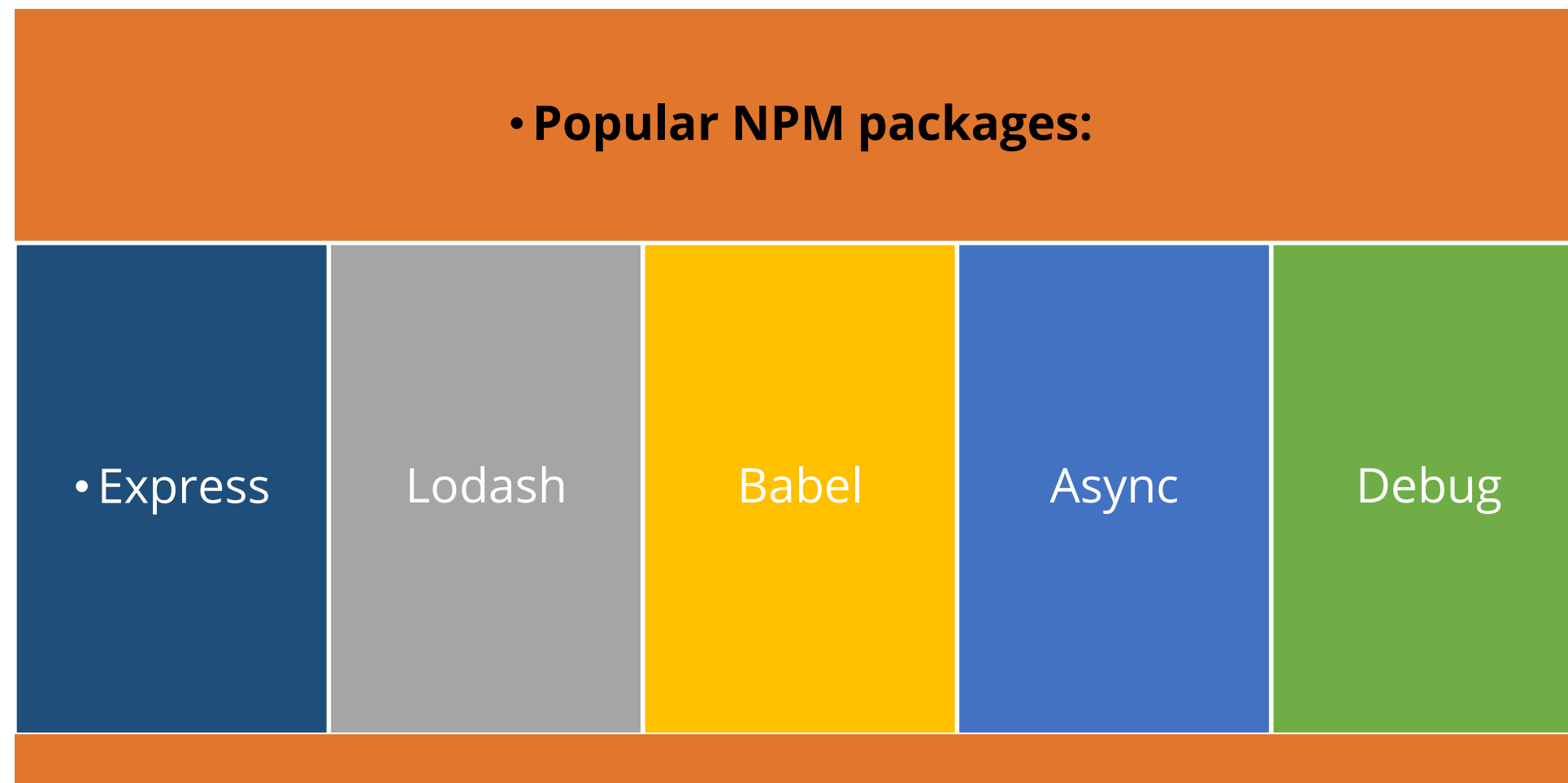
Uses of NPM

- Forming virtual teams by using modules
- Managing multiple versions of code and code dependencies
- Updating applications easily when underlying code is updated
- Discovering multiple ways to solve the same puzzle
- Finding other developers who are working on similar problems and projects



What Is a Package?

- A package is one or more modules grouped together.
- It's commonly used by other packages or a project on its own.
- It must contain a package.json file in order to be published in the npm registry.
- Node.js uses a package manager, where you can find and install packages.



Package.json File

- Package.json is the manifest file of any **Node.js** project that contains metadata of the project
- A package.json file must contain **name** and **version** fields
- **Name** field:
 - Contains the package name
 - Must be in lowercase
 - Should have one word
 - May contain hyphens and underscores.
- **Version** field must be in the form x.x.x and follow semantic version guidelines

Example:

```
{  
  "name": "my-own-package",  
  "version": "1.0.0"  
}
```


What Is a Module?

Module in Node.js can be a simple or a complex functionality organized in single or multiple JavaScript files. It can be reused throughout the Node.js application.

• **Node.js includes three types of modules:**

• Core modules

Local modules

Third-party modules

Core Modules

Core Module	Description
http	In http module, classes, methods, and events are included to create Node.js http server.
url	In url module, methods are included for URL resolution and parsing.
querystring	In querystring module, methods are included to deal with query string.
path	In path module, methods are included to deal with file paths.
fs	In fs module, classes, methods, and events are included to work with file I/O.
util	In util module, utility functions are included which is useful for programmers.

Importing NPM Modules

To install a package using NPM in a folder, you need to install **lodash**, the popular JavaScript utility library, using:

```
npm install lodash
```

This will install the package in the folder. To use it in your code, you need to import it into your program using **require**:

```
const _ = require('lodash')
```

Importing NPM Modules



Duration: 30 min.

Problem Statement:

Write a program to import NPM modules in your local system.

ASSISTED PRACTICE

Assisted Practice: Guidelines to Import NPM Modules

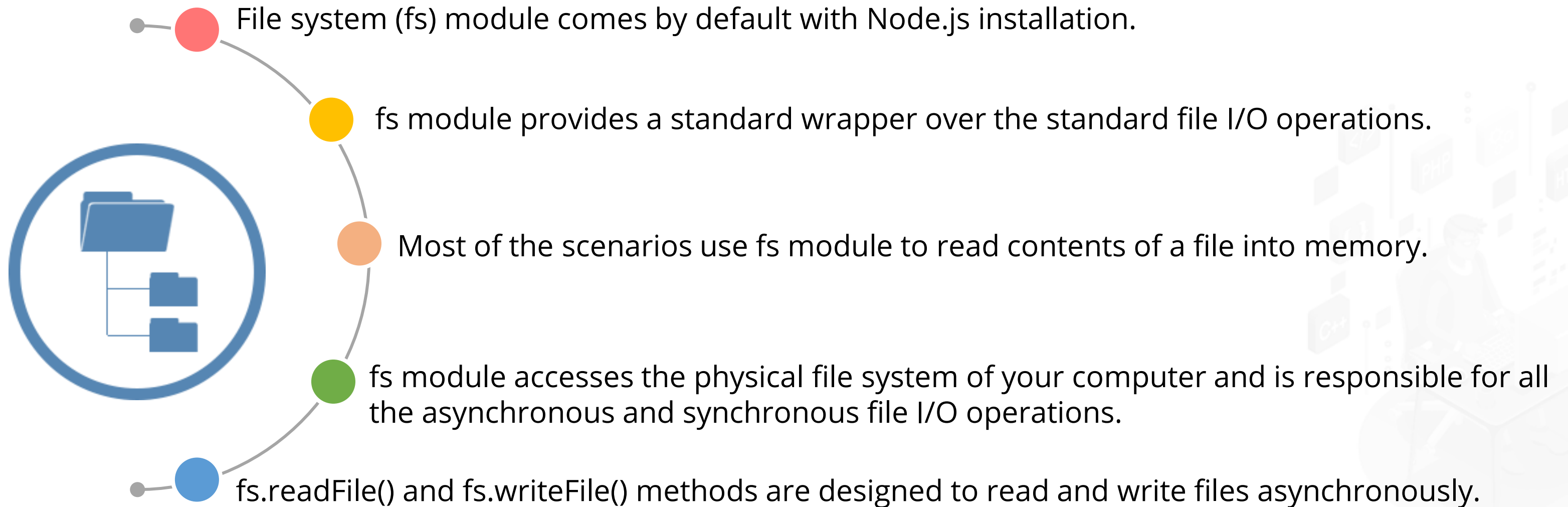
1. Create a package in the app.
2. Write commands to install NPM modules.
3. Initialize the .git file.
4. Add and commit the program files.
5. Push the code to your GitHub repository.



FULL STACK

File System in NPM and Node

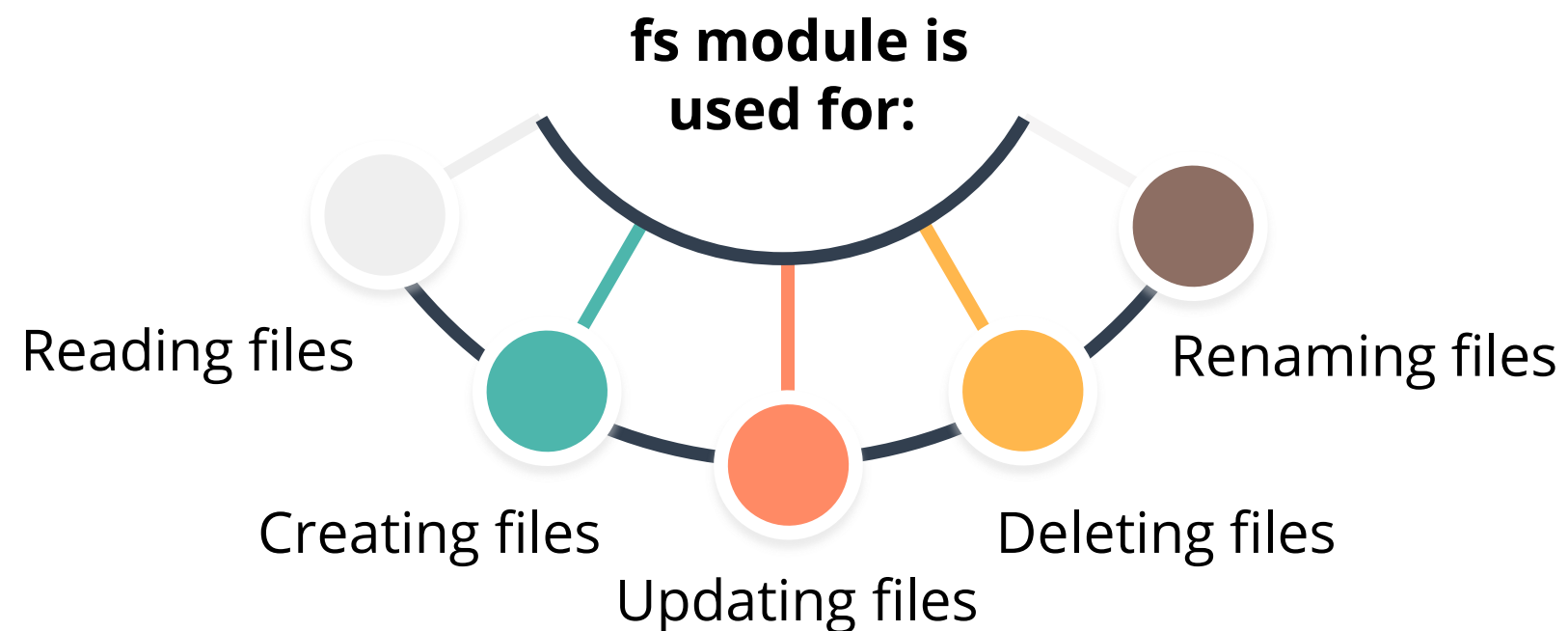
Working with Node.js File System



Standard Method Signature: *fs.readFile(sourceFName [,option], callbackfunction)*

File System

- Every method in a file system executes synchronous and asynchronous functions.
- Functions in fs module that end with **Sync** represent synchronous functions.
- It's good to choose an asynchronous method over a synchronous method, as the asynchronous method never blocks a program during execution.



Commands Used to Handle User Inputs

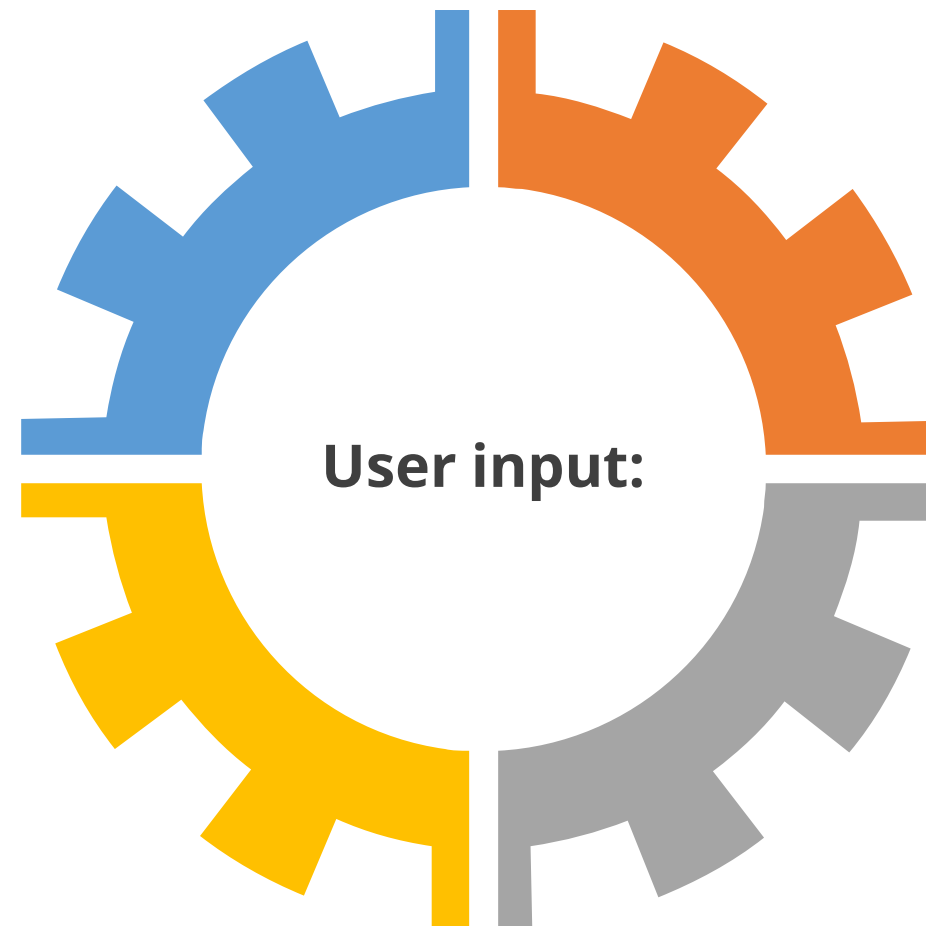
Commands	Description
console.log	Prints the message that needs to be displayed to the user
readline	Provides an interface for reading data from a readable stream
require	Reads a JavaScript file, executes the file, and then proceeds to return the exports object
prompt	Node.js gets user input from command-line prompt

Importance of User Inputs

User input is one of the most important aspects of programming concepts. Every program should have some sort of user interaction.

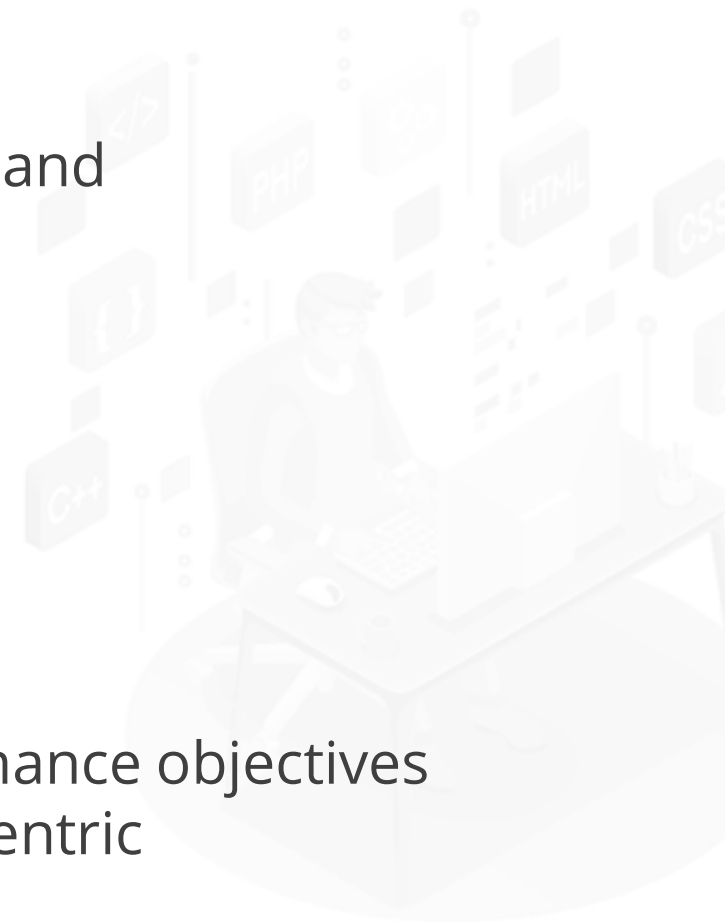
Helps users understand their needs, workflows, and environment

Engages the users in the planning process



Helps users test and evaluate

Sets the performance objectives based on user-centric requirements



Arrow Functions

An arrow function expression is a syntactically compact alternative to a regular function expression, although without its own bindings to the `this`, `arguments`, `super`, or `new.target` keywords.

() => { }



Advantages of Arrow Functions:

- Are simple and allow users to write functions with a short syntax
- Encourage use of small functions



Disadvantages of Arrow Functions:

- Cannot be used as constructors

Handle User Inputs



Duration: 30 min.

Problem Statement:

Write a program to handle user inputs on your local system.

ASSISTED PRACTICE

Assisted Practice: Guidelines to Handle User Inputs

1. Create a package in the app.
2. Write commands to handle user inputs.
3. Initialize the .git file.
4. Add and commit the program files.
5. Push the code to your GitHub repository.



FULL STACK

Debug Your Node Application

Bugs and Debugging

Mistakes that programmers make while writing code are **bugs**.
In programming, it's important to test your program and debug. **Debugging** means removing the bugs.

Types of Bugs:

- A syntax bug is an error caused by something the programmer has typed. It could be a spelling mistake or a command that the computer doesn't understand.
- A logical bug is an error which means that even though the computer can carry out its instructions, it doesn't act as the programmer intended or the user expects.

Debugging in Node.js

- Node.js includes an out-of-process debugging utility accessible via a V8 Inspector and built-in debugging client.
- To use it, start Node.js with the inspect argument followed by the path to the script to debug; a prompt will be displayed indicating successful launch of the debugger.

Install it with:

```
npm install -g node-inspector
```

Then run:

```
Node-debug projectname.js
```

Error Handling in Node.js

In Node.js, errors are handled through exceptions.

An exception is created using the throw keyword:

```
throw value
```

In Node.js, we don't throw strings; we just throw error objects. An error object is an object that is either an instance of the error object or extends the error class.

```
throw new Error('Ran out of tea')
```

or

```
class NotEnoughTeaError extends Error {  
  //...  
}  
throw new NotEnoughCoffeeError
```



Handling Exceptions

An exception handler is a **try/catch** statement.

Any exception raised in the lines of code included in the **try** block is handled in the corresponding **catch** block:

```
try {  
    //lines of code  
} catch (e) {  
  
}
```

e in this example is the exception value.
You can add multiple handlers that can catch different kinds of errors.



Debug your Node Application



Duration: 20 min.

Problem Statement:

You are given a project to debug a node application on your local system.

ASSISTED PRACTICE

Assisted Practice: Guidelines to Debug a Node Application

1. Create a package in the app.
2. Write commands to debug the app.
3. Initialize the .git file.
4. Add and commit the program files.
5. Push the code to your GitHub repository.



Key Takeaways

- Node.js is an open-source, cross-platform which runs JavaScript code outside of a browser
- Node.js is asynchronous, event-driven, single threaded, and highly scalable
- NPM stands for Node Package Manager, which is a package manager for JavaScript programming language
- A try-catch block is used to handle exceptions in Node.js



Debug a Customized Node Application

Duration: 60 min.

Problem Statement:

You are given a project to record logs and errors in a document.



LESSON-END PROJECT

Before the Next Class

You should know:

- Building an application using TypeScript
- Basics of Angular
- Progressive Web App Development with Angular
- Fundamentals of NPM and Node.js



Online Test Application

Duration: 240 min.

Project Objective:

As a Full Stack Developer, you have to develop an online test portal for assessment.



PHASE-END PROJECT

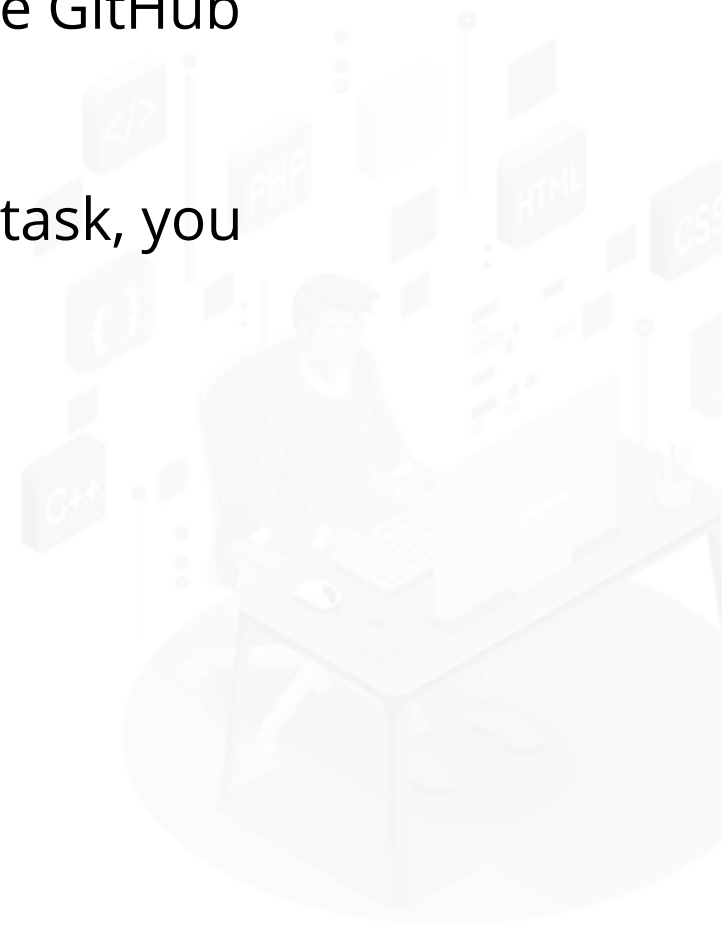
Background of the Project Statement

Create an online test application in order to help your companies recruitment team to take online tests of the candidates coming for interviews.



You Are Asked to Do

- A few of the source codes should be tracked on GitHub repositories.
- You need to document the tracked files that are ignored during the final push to the GitHub repository.
- The submission of your GitHub repository link is mandatory. In order to track your task, you need to share the link of the repository in the document.
- The step-by-step process involved in completing this task should be documented.



You must use the following



Visual Studio Code



Angular



Git and GitHub



Node.js



Specification document

