Install Node.js

Download the latest version

Check version node -v (check if installed correctly or not)

Mistype nodee -v

Visual studio code alternatives atom, sublime text, web storm

What is Nodejs?
Javascript only run in the browser -> run as a standalone process on your machine
Limited to what browser allowed it to do -> more like a general purpose programming language

Click event to a button, redirect to a different page -> web servers that can access the file system and connect to databases

Instead of being forced to run it on the client -> run javascript code on the server

Nodejs is a javascript runtime built on Chrome's v8 JS engine. V8 is a Google open source project.
JS engine, takes JS code complies it to machine code that your machine can execute.

V8 is written in C++. Chrome and Nodejs are largely written in C++.
Nodejs is not a programming language. It provides custom functionality with functions and objects.
Both chrome and Nodejs create their own modified version of javascript. The same javascript language but with custom functions and objects injected.

Chrome and Nodejs pass the js code to V8 and get the results back.

In console (global, process) and chrome (window, document)
2+3
'Can".toUpperCase()

Nodejs extends javascript functionality by providing C++ bindings. To do anything that C++ can do such as accessing the file system.

Why Nodejs?
Nodejs is an event driven non-blocking IO model that makes it lightweight and efficient
IO is input output and your app uses IO to communicate with the outside world (with the machine it's running on (file system) or a remote server)

Non-blocking - when the application is waiting for a response, it can do other things (process other code make other requests, etc.)

Non-blocking because it would be a bad experience otherwise. When the data is being fetched, the user could not click links or buttons.
Those operations run in the background both on Nodejs and browser.

Event driven is registering callbacks and having them call when the IO operation is done.

npm is the largest open source ecosystem. npm is already installed with nodejs. You can search available packages on the npm website.

```
console.log('Hello Node.js!')
```

```
node index.js
```

Importing core modules

```
fs.writeFileSync('notes.txt', 'I live in Bursa’)
```

Error!

```
const fs = require('fs')
```

Fs can be changed. No output. Check the file.

```
fs.writeFileSync('notes.txt', 'I am Can’)
```

Rerun the code. The file is overwritten.
Append a message?

```
fs.appendFileSync('notes.txt', ' I live in Bursa’)
```

Run the script. Open the file.

Importing your own files.

Whe the project gets bigger, create multiple files to stay organized.
Delete the code.

const name = 'Can'
console.log(name)

Run the script.
Create a new file. utils.js under notes-app
console.log(‘utils.js’)
Run the script. Utils is not printed.

Add require(‘./utils.js’) at the top of app.js
Save and rerun. Require is at the top and printed earlier.

Now define a function.

Remove name variable from app.js, define it in the utils.

const name = ‘cem’

Error. Name is not defined. All files have their own scope.
To access, export it in utils.

module.exports = name

İn app.js
const name = require(‘./utils.js’)
Variable name can be different.

Now export a function in utils

const add = function (a,b) {return a+b}
Module.exports = add

Change name to add in app.js
const add = require(‘./utils.js’)
Const sum = add(4,-2)
Console.log(sum)

Rerun the script.

Challenge!
Define a new file called note.js
Create get notes function, export it
From app.js, call the function.

const getNotes = function () {return ‘your notes …’}
module.exports = getNotes

İn app.js
Delete/comment earlier codes

const getNotes = require(‘./notes.js’)
const msg = getNotes()
console.log(msg)

Remove utils.js now.

Importing npm modules

Initialize npm
Run at the root level.
node -v
npm -v (greater than 5!)

npm init (create configuration file. Accept default values). Most of the values are for those who create npm packages. It creates a package.json configuration file. JSON is javascript object notation. Invalid if remove double quotes.

Data validation does not make my app unique. Do not reinvent the wheel and focus on the app functionality. Validates email address, url, etc.

Install the data validator.
npm install validator@

Updated package.json dependencies. Created node_modules folder and package-lock.json file. Do not alter these files.

Go to app.js Add the second require.
const validator = require('validator')

console.log(validator.isEmail('can@btu.edu'))
Run the app. node app.js
Remove @ and run again.

console.log(validator.isURL('www.btu.edu'))
Run again.

Node-modules folder is missing if downloaded from a repository. To try, delete this folder. In this case, run, npm install Packages to be installed are determined according to the package.json configuration file.

Install chalk module. Chalk prints custom colored text.
npm -i chalk

Remove validator library from the app.js and load it. Then print success in green.
const chalk = require('chalk')
console.log(chalk.green.bold('Success!'))

Try chalk.bold.green('Success!')
Try chalk.bold.inverse.green('Success!')

Global npm modules. Up to here we used local modules. When we install globally, it gives us access to a new command from the terminal. It is not changing the project files. Not in the dependency list.

npm install nodemon -g

In case of an error, try sudo! Or run as admin.

nodemon -v

Run the script as follows:
nodemon app.js

Change the code from success to error and green to red.

Use ctrl C to quit.