**Node Modules**

In the last previous chapters, we’ve look to the variables and functions that are globally available.

One of those functions are require() function. Require function is common JS module pattern. But only represents half of the pattern. The half that load the modules. The other half of the pattern is module.exports or the mechanism by which we make our modules consumable.

**Path Module:**

In that one we looked at how to load the **path** module by passing it in this function as an argument.

Ex:

// *Path Module*

const *path* = *require*("path");

*console.log*(*path.basename*(\_\_filename));

// *Any string passing here, will appended to the dir path.*

const *dirUploads* = *path.join*(\_\_dirname,"www","files","uploads");

*console.log*(dirUploads);

**Utilities Module:**

Util module has a log method as well. It gives litter more info. about our file name.

Ex:

// *Path Module*

const *path* = *require*("path");

*console.log*(*path.basename*(\_\_filename));

// *Any string passing here, will appended to the dir path.*

const *dirUploads* = *path.join*(\_\_dirname,"www","files","uploads");

*console.log*(dirUploads);

// *Utilities Module*

const *util* = *require*("util");

// *it gives litter more info. about our file name.*

*util.log*(*path.basename*(\_\_filename));

**v8 Module:**

It gives information about how much memory our app is consuming using getHeapStatistics() method.

Ex:

// *Path Module*

const *path* = *require*("path");

// *Utilities Module*

const *util* = *require*("util");

// *v8 Module*

const *v8* = *require*("v8");

*console.log*(*path.basename*(\_\_filename));

// *Any string passing here, will appended to the dir path.*

const *dirUploads* = *path.join*(\_\_dirname,"www","files","uploads");

*console.log*(dirUploads);

// *it gives litter more info. about our file name.*

*util.log*(*path.basename*(\_\_filename));

*util.log*(*v8.getHeapStatistics*());

**Readline Module:**

Readline is a module that allows us to ask questions of our terminal user. It a wrapper around the standard input & output process that allows us to easily control the user prompt without having to work directly with stdin and stdout.

Ex:

// *Readline Module*

const *readLine* = *require*("readline");

const *rl* = *readLine.createInterface*({

*input*: *process.stdin*,

*output*: *process.stdout*

});

*rl.question*("How do you like Node? ", (answer) => {

*console.log*(`Your Answer: ${answer}`);

})

**Output:**How do you like Node? It's Great!  
Your Answer: It's Great!

**Using of Readline module:**

Ex:

// *Readline Module*

const *readLine* = *require*("readline");

const *rl* = *readLine.createInterface*({

*input*: *process.stdin*,

*output*: *process.stdout*

});

const *questions* =[

"What is your Name? ",

"Where do you live? ",

"What are you going to do with Node.js? "

];

function *collectAnswers*(questions, done) {

    const *answers* =[];

    const *questionAnswered* =(*answer*)=>{

*answers.push*(*answer.trim*());

if(*answers.length* < *questions.length*){

*rl.question*(questions[*answers.length*],questionAnswered);

}

else

*return done*(answers);

};

*rl.question*(questions[0], questionAnswered);

}

*collectAnswers*(questions, (answers) => {

*console.log*("Thank You for your Answers!");

*console.log*(answers);

*process.exit*();

})

**Output:**

What is your Name? Chandan

Where do you live? Jharkhand

What are you going to do with Node.js? Doing good things, make good profits.

Thank You for your Answers!

[ 'Chandan', 'Jharkhand', 'Doing good things, make good profits.' ]