Hands-on lab on Node JS

The purpose of this lab is to brush up on Node.js framework before you set off doing server side coding with Node JS. This lab presumes that you have completed the IBM HTML CSS and JS for Web development. All the tasks in this lab are intended to reinforce what you have already learnt.

Duration (15 mins)

Objective

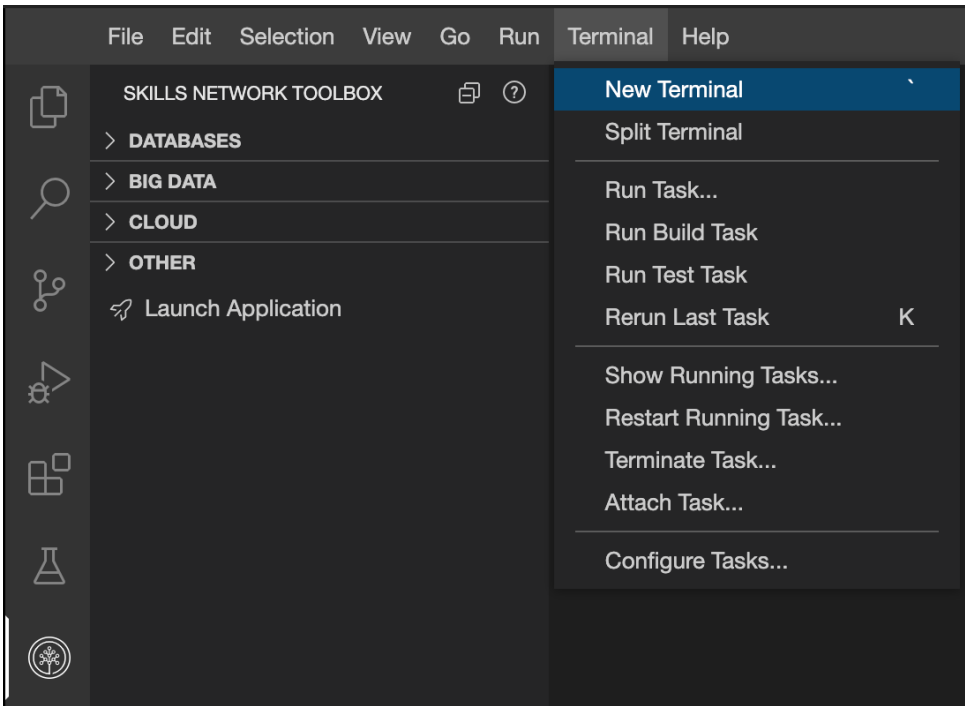
After completing this lab you will be able to:

1. Run Node JS commands in the terminal
2. Write Node JS applications

Task 1 - Running Node JS commands

To run the commands we will use the terminal. Type or paste the command and press **enter** to run the command.

1. Open a new terminal



2. Run the following command to open the **node** prompt

```
1
node
```

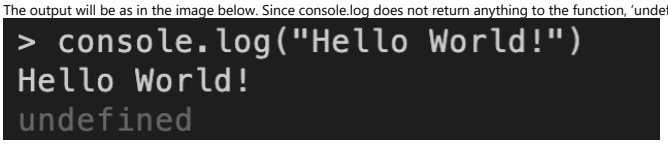
Copied!

3. Let's start with a simple code to print **Hello World!** to the console. Run the following command.

```
1
console.log("Hello World!")
```

Copied!

The output will be as in the image below. Since console.log does not return anything to the function, 'undefined' is returned by default



4. Let's create some variable and print them. Run the following command.

```
1
2
3
4
```

```
let num = 5
var mystr = "John"
console.log(num)
console.log(mystr)
```

Copied!

Both let and var can be used to create variables. `var` is used when you want the variable to have global scope and `let` is used when you want the variable to have scope within the block where it is created.

5. Let's create a constant and print it. Run the following command.

```
1
2
```

```
const pi_val = 3.147
console.log(pi_val)
```

Copied!

Const is used to declare variable whose values can never change

6. Let's create function which prints any value that is input to it.

```
1
2
```

```
3. 3
```

```
1. function printMyInput(user_input) {
2.     console.log("The parameter passed is "+user_input)
3. }
```

Copied!

7. Call the function you created in the previous step once with a number and once with a string.

```
1. 1
2. 2
```

```
1. printMyInput(9)
2. printMyInput("John")
```

Copied!

8. Let's rewrite the function printMyInput according to the ES6 standard. This syntax is also called `arrow functions` and provide a shorthand to write functions.

```
1. 1
2. 2
3. 3
```

```
1. let printMyInputES6 = (user_input)=>{
2.     console.log(user_input)
3. }
```

Copied!

9. Call the function you created in the previous step once with a number and once with a string.

```
1. 1
2. 2
```

```
1. printMyInputES6(9)
2. printMyInputES6("John")
```

Copied!

Since the function is passed a single value and the body of the function is a single line, the brackets can be omitted. The code can also be written as below.

```
1. 1
```

```
1. let printMyInputES6Short = user_input => console.log(user_input)
```

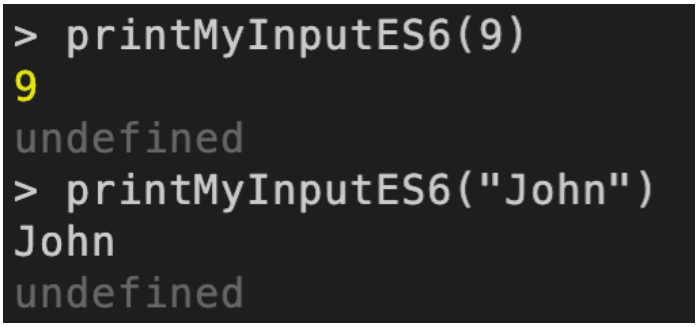
Copied!

Now when we call it, the output should remain the same.

```
1. 1
2. 2
```

```
1. printMyInputES6(9)
2. printMyInputES6("John")
```

Copied!



10. Press Ctrl+D to exit the prompt.

Task 2 - Writing JS file

1. At the command prompt, in the terminal window, run the following command to install a node package called `prompt-sync`.

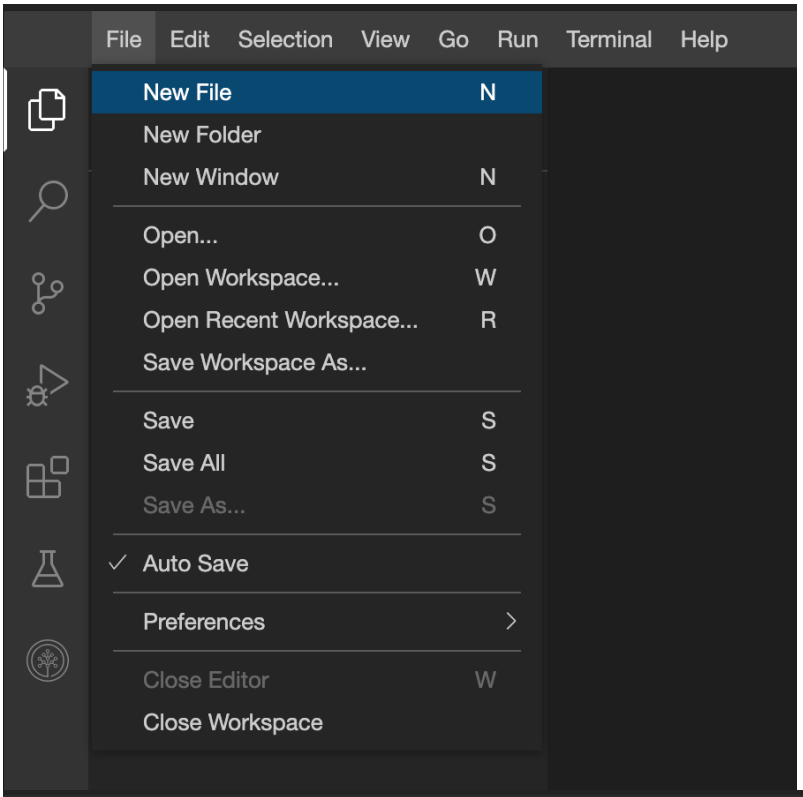
```
1. 1
```

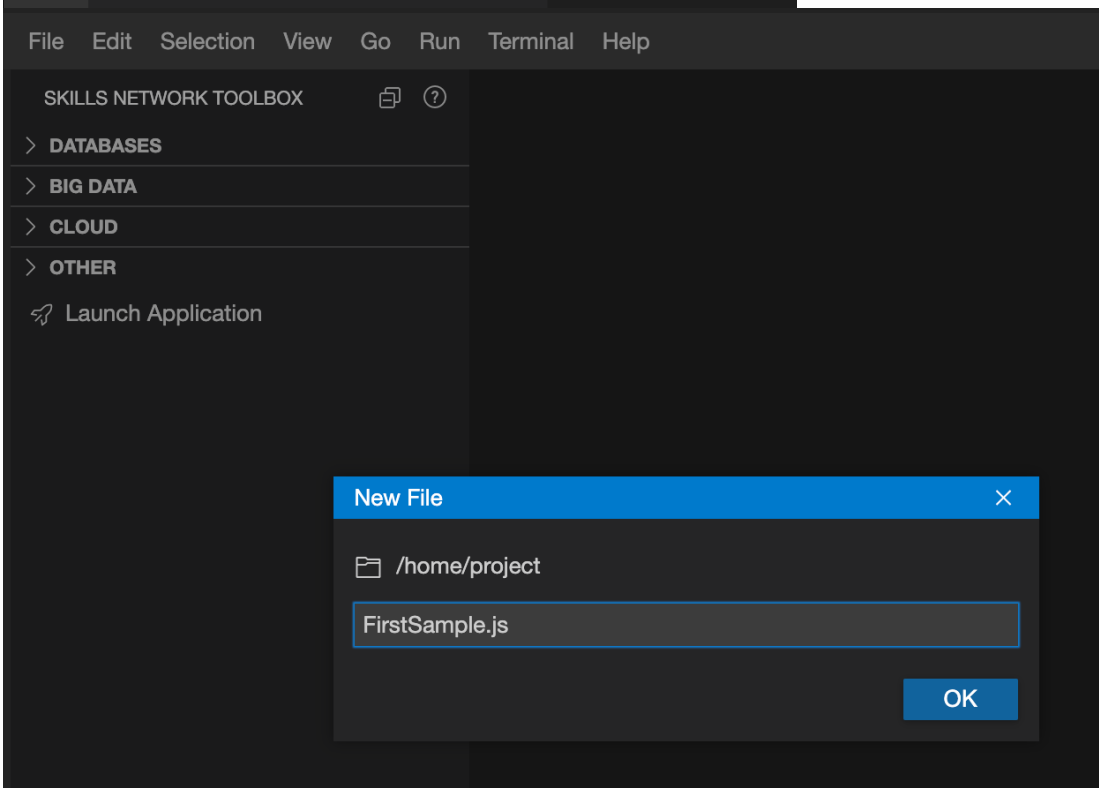
```
1. npm install -s prompt-sync
```

Copied!

`npm install` command installs the packages that are not available by default. Once installed these packages can be imported in the code. In the code sample above, installs the latest version of the package **prompt-syn** in silent mode (**-s**).

2. Create a new file named FirstSample.js





3. Paste the following code and save the file.

```
1
2
3
4
5
6
7
8
9
10
11
12
//require the  package to be used in the code and assign it to a variable name
const prompt = require("prompt-sync")();

//Create an ES6 arrow function that prints any parameter that is passed
let printMyInput = (user_input) => {
    console.log("The parameter passed is "+user_input)
}

let user_input = prompt("Enter some string or number: ");

//Call the method passing the user_input as parameter
printMyInput(user_input)
```

Copied!

4. Run the code. Enter a string when prompted and observe the output.

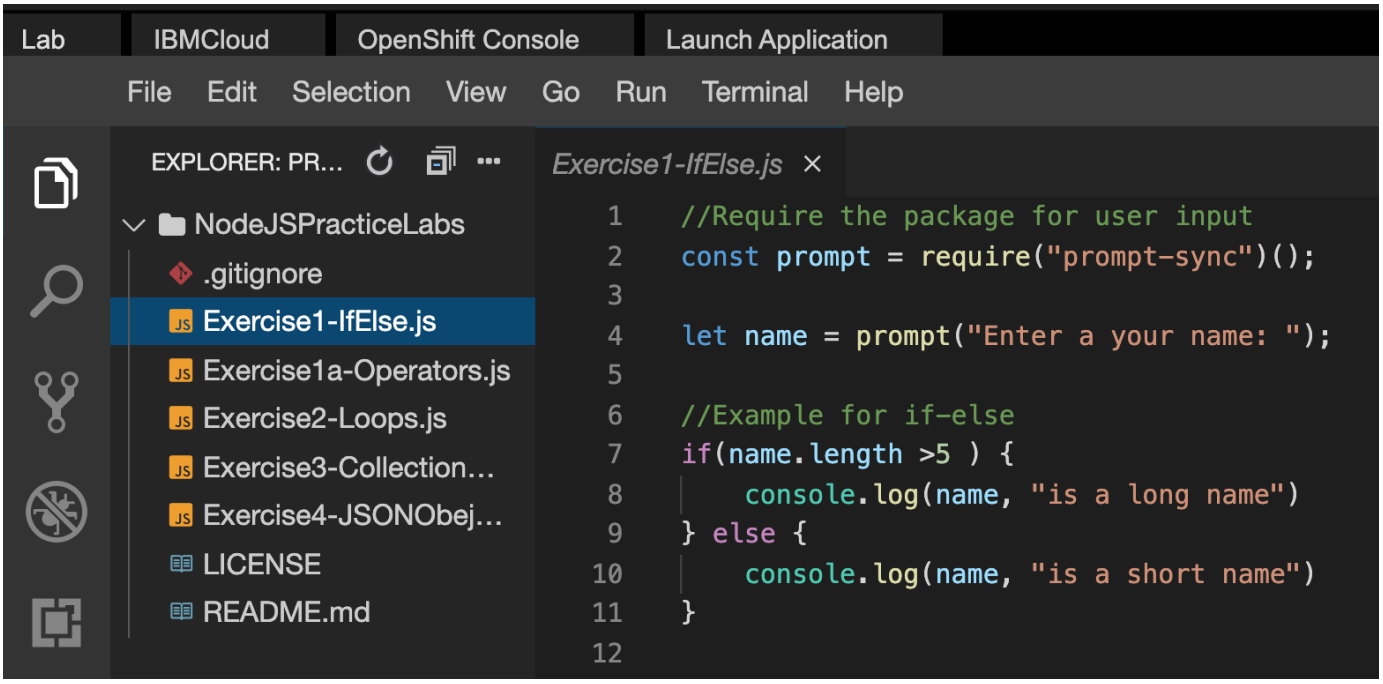
```
1
```

```
node FirstSample.js
```

Copied!

Task 3 - Operators, Controls, Loops

In this task you will be running some javascripts from which you can learn how to use operators, controls and loops. After cloning you can view the code on the file explorer.



Ensure that you understand the code in each file. These are primitive and foundational for your understanding of Node JS

1. Clone the remote repository.

```
1. 1
1. git clone https://github.com/ibm-developer-skills-network/NodeJSPracticeLabs.git
```

Copied!

2. Change to the exercise directory.

```
1. 1
1. cd NodeJSPracticeLabs
```

Copied!

3. View and run Exercise1-IfElse.js. This is focussed on `if-else`, `if-else if-else` and `switch-case`.

```
1. 1
1. node Exercise1-IfElse.js
```

Copied!

Practice Exercises

1. Write code which accepts user input as a number. Depending on the number input, between 1 to 7, prints days of the week starting from Sunday for 1.

   Hint - You can use if-else if- else or switch-case

▸ Click here for the solution

2. View and run Exercise1a-Operators.js. This is focussed on operators. You can refer to the cheatsheet to get more ideas. We also use `parseInt` (convert a string to an int) and `isNaN` (Checks if not a number) methods.

```
1. 1
1. node Exercise1a-Operators.js
```

Copied!

3. Write a code which accepts an input from the user and prints whether it is a number or not.

▸ Click here for the solution

4. View and run Exercise2-Loops.js. This is focussed on `for loop`, `while loop` and `do while`.

```
1. 1
1. node Exercise2-Loops.js
```

Copied!

5. Write a code which accepts a number and loops from 1 until that number and prints the value.

▸ Click here for the solution

   Additional Challenge: Try doing the same with while and do while loops

6. View and run Exercise3-CollectionObjects.js. This is focussed on arrays and dictionary objects.

```
1. 1
1. node Exercise3-CollectionObjects.js
```

Copied!

7. Write a code which stores the colours of the rainbow in a string array and prints the values using a for-in loop.

▸ Click here for the solution

8. View and run Exercise4-JSONObjects.js.This is focussed on JSON Objects and traversing through them and filtering them.

```
1. 1
1. node Exercise4-JSONObjects.js
```

Copied!

9. Write a code which store the following JSON object as birthday register and searches through it based on the name input by the user and print the birthday.

```
1. 1
2. 2
3. 3
4. 4
5. 5
6. 6
```

```
1. {"friends":[
2.     {"name":"Wendy","Birthday":"12th October"},
3.     {"name":"Jacob","Birthday":"15th March"},
4.     {"name":"Nicolas","Birthday":"3rd June"},
5.     {"name":"Marcus","Birthday":"16th December"}
6. ]}
```

Copied!

▸ Click here for the solution

Author(s)

Lavanya

Changelog

Date	Version	Changed by	Change Description
1-Sep-2021	1.0	Lavanya	Created the lab
18-Jan-2023	2.0	Lavanya	Updated the lab

© IBM Corporation 2023. All rights reserved.