

**COURSE** : PLATFORM BASED PROGRAMMING  
**SESSION** : II  
**SUBJECT** : Practicum Guide: JavaScript Basics in Node.js  
**STUDENT NAME** : Ujang Herlan

### Practicum Objective

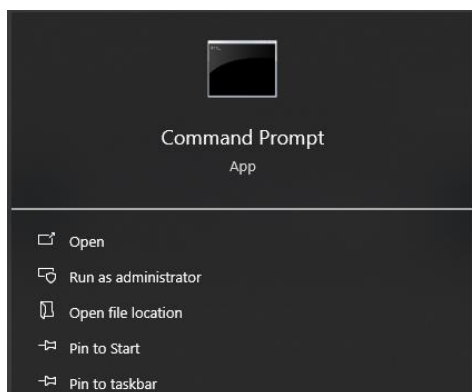
1. Understand the use of JavaScript in the Node.js environment.
2. Get to know the basics of the JavaScript programming language.
3. Understand the use of variables, data types in JavaScript.
4. Use built-in and third-party modules in Node.js.

#### 1. Create a project in the following way:

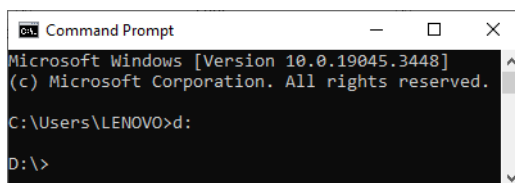
- a. Open a console window, type "command" in the search input



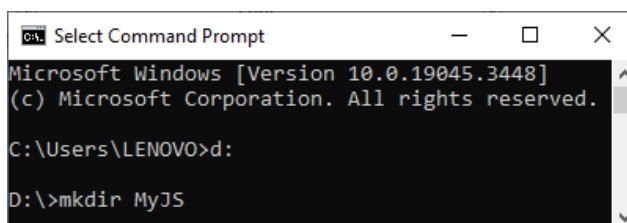
- b. then the search results will appear as follows:



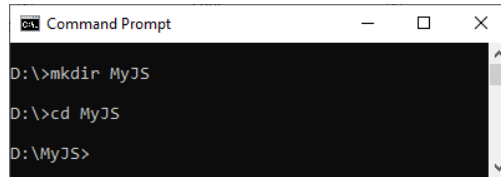
- c. then click "Open"
- d. Change the active drive according to your own hard disk, for example change it to the d:\ drive as shown below:



- e. Create the **MyJS** folder using the **mkdir name-directory** syntax as shown below:

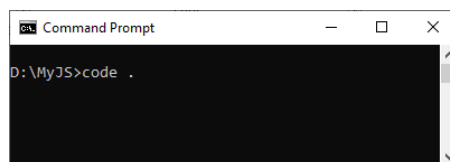


- f. then change the current directory to **MyJS**, using the **cd name-directory** (change directory) syntax, as follows:



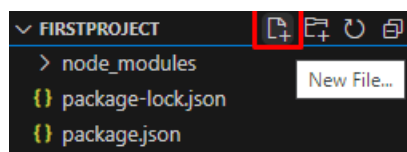
```
D:\>mkdir MyJS
D:\>cd MyJS
D:\MyJS>
```

- g. The next step is to initialize the project using the npm init syntax. Fill in some parameters that will be used to generate the package.json file
- h. type **code .** to open the project folder into visual studio code as follows



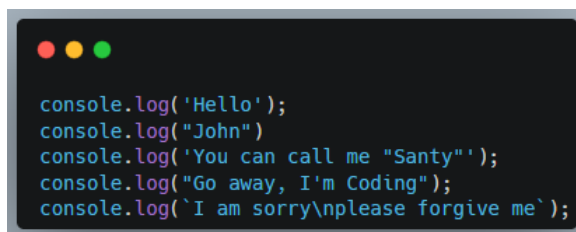
```
D:\MyJS>code .
```

- i. Finally, create a file with the name index.js in the visual studio code, by pressing the (+) sign or new file as follows:



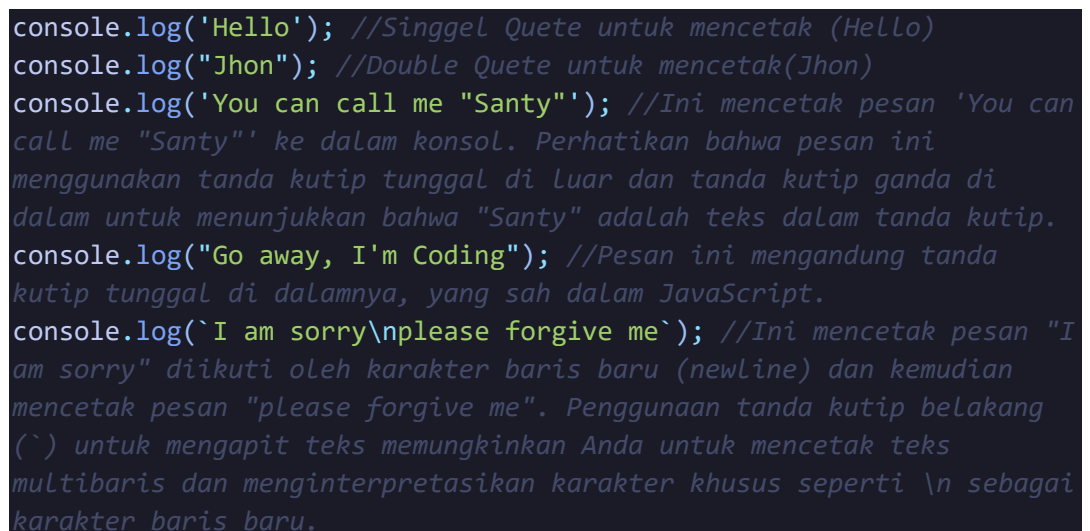
## 2. Making a program

- a. Write the following program code !



```
console.log('Hello');
console.log("John");
console.log('You can call me "Santy"');
console.log("Go away, I'm Coding");
console.log(`I am sorry\nplease forgive me`);
```

Display the results of the program code and write your analysis in the box below !



```
console.log('Hello'); //Singgel Quote untuk mencetak (Hello)
console.log("Jhon"); //Double Quote untuk mencetak (Jhon)
console.log('You can call me "Santy"'); //Ini mencetak pesan 'You can call me "Santy"' ke dalam konsol. Perhatikan bahwa pesan ini menggunakan tanda kutip tunggal di luar dan tanda kutip ganda di dalam untuk menunjukkan bahwa "Santy" adalah teks dalam tanda kutip.
console.log("Go away, I'm Coding"); //Pesan ini mengandung tanda kutip tunggal di dalamnya, yang sah dalam JavaScript.
console.log(`I am sorry\nplease forgive me`); //Ini mencetak pesan "I am sorry" diikuti oleh karakter baris baru (newline) dan kemudian mencetak pesan "please forgive me". Penggunaan tanda kutip belakang (`) untuk mengapit teks memungkinkan Anda untuk mencetak teks multibaris dan menginterpretasikan karakter khusus seperti \n sebagai karakter baris baru.
```

- b. Write the following program code !

```
//What is output display??
let number1 = 3;
let number2 = 9;
let number3 = 7;
let number4 = 6;
let number5 = 2;
console.log(number1 + number2 - number3 * number4 / number5);
console.log(number1 + (number2 - number3) * number4 / number5);
```

Display the results of the program code and write your analysis in the box below !

Hasil dari pernyataan pertama `console.log(number1 + number2 - number3 * number4 / number5);` adalah -9. Ini terjadi karena operasi matematika dijalankan sesuai dengan aturan prioritas operator matematika, sehingga kita mendapatkan hasil -9 setelah melakukan penambahan, pengurangan, perkalian, dan pembagian pada variabel `number1`, `number2`, `number3`, `number4`, dan `number5`.

Hasil dari pernyataan kedua `console.log(number1 + (number2 - number3) * number4 / number5);` adalah 9. Pada pernyataan ini, kita menggunakan tanda kurung untuk mengatur urutan operasi, sehingga terlebih dahulu menghitung pengurangan antara `number2` dan `number3`, kemudian melakukan perkalian dan pembagian dengan variabel `number4` dan `number5`, serta akhirnya menambahkan hasilnya ke `number1`, menghasilkan nilai 9.

- c. Write the following program code !

```
let conditionA = true;
let conditionB = false;
let conditionC = true;
let conditionD = false;
console.log(conditionA && conditionB || conditionC && !conditionD);
```

Display the results of the program code and write your analysis in the box below !

Kode yang diberikan mendeklarasikan empat variabel, kondisiA, kondisiB, kondisiC, dan kondisiD, lalu mencatat hasil ekspresi boolean ke konsol. Ekspresinya menggunakan operator logika untuk menggabungkan nilai variabel. Secara khusus, lalu memeriksa apakah kondisiA benar DAN kondisiB salah ATAU kondisiC benar DAN kondisiD salah.

Hasil ekspresi ditentukan sebagai berikut:

- kondisiA && kondisiB bernilai salah karena kondisiB salah.
- conditionC && !conditionD bernilai benar karena conditionC benar dan !conditionD (bukan conditionD) juga benar karena conditionD salah.
- salah || true bernilai benar karena setidaknya salah satu operan bernilai benar.

Oleh karena itu, ekspresi tersebut log true ke konsol.

d. Write the following program code !

```
let age = 9;  
let result = (age > 17) ? "Adults" : (age > 10) ? "Teenagers" : "Children";  
console.log(result);
```

Display the results of the program code and write your analysis in the box below !

Kode yang diberikan mendeklarasikan variabel age dengan nilai 9 dan kemudian memberikan nilai pada variabel result berdasarkan nilai age. Ekspresinya menggunakan operator ternary ? untuk memeriksa apakah usia lebih dari 17 tahun. Jika ya, nilai hasil disetel ke "Dewasa". Jika tidak, ia akan memeriksa apakah usia lebih besar dari 10. Jika ya, nilai hasil disetel ke "Remaja". Jika tidak, nilai result ditetapkan ke "anak-anak".

Karena usia kurang dari 17 dan 10, ekspresi bernilai false untuk kedua kondisi dan memberikan nilai "anak" pada hasil. Terakhir, kode tersebut mencatat nilai result ke konsol, yaitu "anak-anak".

- e. Write the following program code!

```
let numA = 176;
let numB = 103;

console.log(numA & numB);
console.log(numA | numB);
console.log(numA ^ numB);
console.log(numA >> 2);
console.log(numB >> 2);
console.log(numA << 2);
console.log(numB << 2);

let numC = 1;
console.log(numC << 1);
console.log(numC << 2);
console.log(numC << 3);
console.log(numC << 4);
console.log(numC << 5);
console.log(numC << 6);
console.log(numC << 7);
console.log(numC << 8);
```

```
let numA = 176
let numB = 103
console.log(numA & numB); // Operasi bitwise AND (Logika AND bit per bit)
console.log(numA | numB); // Operasi bitwise OR (Logika OR bit per bit)
console.log(numA ^ numB); // Operasi bitwise XOR (Logika XOR bit per bit)
// Pergeseran bit ke kanan sebanyak 2 posisi
console.log(numA >> 2);
console.log(numB >> 2);
// Pergeseran bit ke kiri sebanyak 2 posisi
console.log(numA << 2);
console.log(numB << 2);
```

```
let numC = 1
// Pergeseran bit ke kiri sebanyak 1, 2, 3, 4, 5, 6, 7, dan 8 posisi
console.log(numC << 1)
console.log(numC << 2)
console.log(numC << 3)
console.log(numC << 4)
console.log(numC << 5)
console.log(numC << 6)
console.log(numC << 7)
console.log(numC << 8)
```

Hasil: 32, 247, 215, 44, 25, 704, 412, 2, 4, 8, 16, 32, 64, 128, 256

- f. Write the following program code!

```
let fullName = (fn,mn,ln) => {  
  return `${fn} ${mn} ${ln}, ${getTitle()}`;  
}  
  
let getTitle = () =>{  
  return "B.Sc"  
}  
  
console.log(fullName("John", "Dream","Myung"));
```

Kode tersebut mendefinisikan fungsi getTitle yang tidak memiliki parameter dan mengembalikan string "B.Sc".

Dalam Kode tersebut mendefinisikan fungsi fullName dengan tiga parameter: fn (nama depan), mn (nama tengah), dan ln (nama belakang). Fungsi ini menggabungkan ketiga parameter ini ke dalam satu string dengan spasi di antara mereka, dan kemudian menambahkan "B.Sc" dengan memanggil fungsi getTitle().

Terakhir, memanggil fungsi fullName dengan argumen "Jhon" (nama depan), "Dream" (nama tengah), dan "Myung" (nama belakang).

Output:

Jhon Dream Myung, B.Sc

- g. what is the difference between operator == and ===, give an example of the program code.

Perbandingan dengan menggunakan simbol == hanya akan membandingkan nilai saja. Sedangkan yang menggunakan === akan membandingkan dengan tipe data juga.

```
let num = 5;  
let str = "5";  
// Menggunakan ==  
console.log(num == str); // true (konversi tipe data:  
string "5" menjadi angka 5)  
// Menggunakan ===  
console.log(num === str); // false (kesetaraan ketat: tipe  
data berbeda)
```

- h. What is the difference between creating functions using functions and arrow functions?

Cara ini sering digunakan di kode Javascript masa kini, karena lebih sederhana. Akan tetapi sulit dipahami bagi pemula. Fungsi ini mulai muncul pada standar ES6.

Contoh:

```
var namaFungsi = () => {  
  console.log("Hello World!");  
}  
  
// atau seperti ini (jika isi fungsi hanya satu baris):  
var namaFungsi = () => console.log("Hello World!");
```

Sebenarnya hampir sama dengan yang menggunakan ekspresi. Bedanya, kita menggunakan tanda panah ( $\Rightarrow$ ) sebagai ganti **function**.

Pembuatan fungsi dengan cara ini disebut *arrow function*