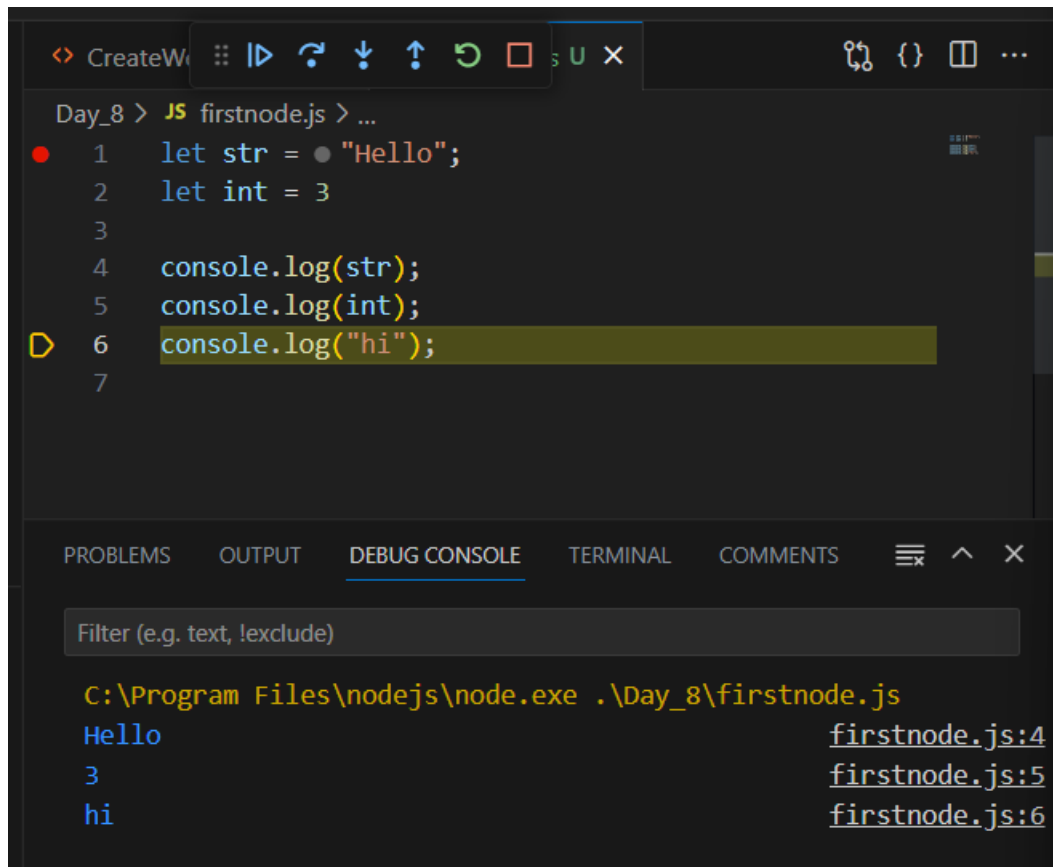


Lab Assignment [WPT] – Day 8

I Write your first nodejs Program. Run the program using command prompt. Run the program using Run And Debug option Debug the program, apply breakpoint.



The screenshot shows the VS Code editor with a file named `firstnode.js` open. The code in the editor is as follows:

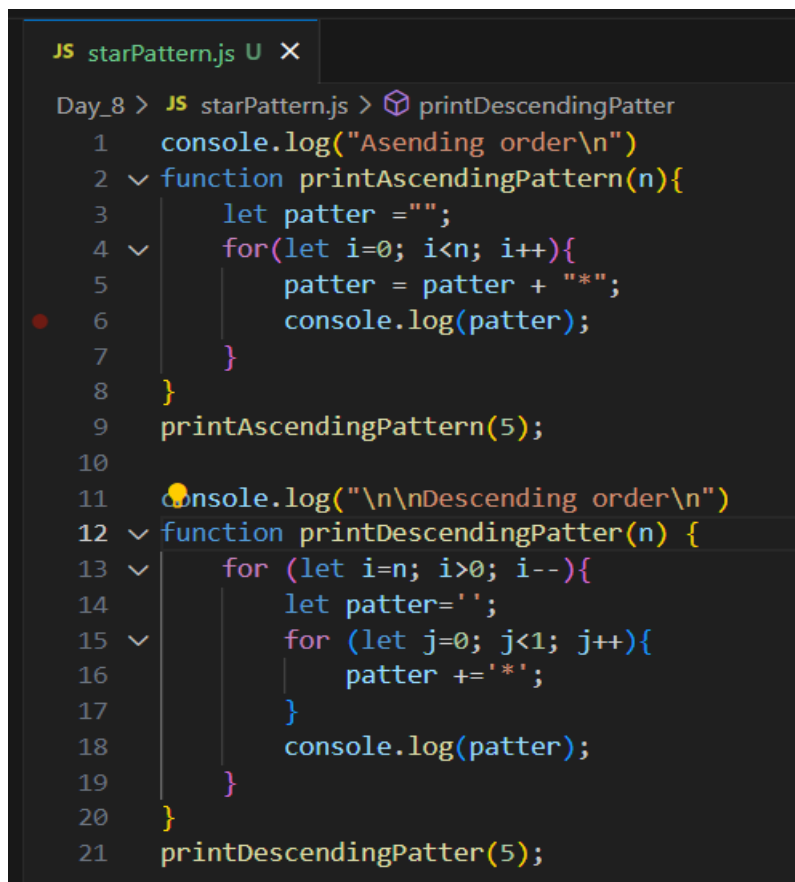
```
Day_8 > JS firstnode.js > ...
1 let str = "Hello";
2 let int = 3
3
4 console.log(str);
5 console.log(int);
6 console.log("hi");
7
```

The `console.log("hi");` line is highlighted. Below the editor, the **DEBUG CONSOLE** tab is active, showing the output of the program:

```
C:\Program Files\nodejs\node.exe .\Day_8\firstnode.js
Hello
3
hi
```

Each output line is followed by a link to the corresponding line in the source file: `firstnode.js:4`, `firstnode.js:5`, and `firstnode.js:6`.

II Pattern Program - Print Star Ascending, Descending.



The screenshot shows the VS Code editor with a file named `starPattern.js` open. The code in the editor is as follows:

```
Day_8 > JS starPattern.js > printDescendingPatter
1 console.log("Asending order\n")
2 function printAscendingPattern(n){
3     let patter = "";
4     for(let i=0; i<n; i++){
5         patter = patter + "*";
6         console.log(patter);
7     }
8 }
9 printAscendingPattern(5);
10
11 console.log("\n\nDescending order\n")
12 function printDescendingPatter(n) {
13     for (let i=n; i>0; i--){
14         let patter='';
15         for (let j=0; j<1; j++){
16             patter += '*';
17         }
18         console.log(patter);
19     }
20 }
21 printDescendingPatter(5);
```

```
TERMINAL    DEBUG CONSOLE    PROBLEMS    OUTPUT

Ascending order

*
**
***
****
*****

Descending orde

*****
****
***
**
*
```

III. Practice and display

Q.1 Variable Declaration [Var Let Const Var vs let vs const]

```
Day_8 > JS variableDeclaration.js > ...
1  //variable declaration
2  var j=10;
3  var j=101;
4  j=j+1;
5  console.log("The value of j is ", j);
6
7  //Let does not allow duplicate declaration
8  let i =100;
9  i=i+1;
10 console.log("The value of i is ",i);
11
12 //const does not allow to chnage its value by any c
13 const id = 100;
14
```

```
PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL    COMMENTS

PS D:\cdac_tvm\5. Web programming technology> cd Day_8
PS D:\cdac_tvm\5. Web programming technology\Day_8> node variable
The value of j is 102
The value of i is 101
PS D:\cdac_tvm\5. Web programming technology\Day_8> █
```

2 String Single quote Double Quote Backtick

```
JS variableDeclaration.js U JS 2.js U X
Day_8 > JS 2.js > ...
1 let n="hihi";
2 let n1 = `Tim David's unbeaten 32 off 19 went in vain as Lucknow Super Giants defeated Mumbai Indians by 5 runs on Tuesday.
3 With Mumbai needing 11 runs off the final over, Mohsin Khan conceded only five runs.
4 Earlier, Marcus Stoinis' fiery 89-run knock guided LSG to 177 for 3 in 20 overs.
5 Getting an invitation to bat first, LSG were 35 for 3 in 6.1 overs but Stoinis and Krunal Pandya (49) lifted the team with a solid patnership.
6 Jason Behrendorff (2 for 30) was pick of the bowlers for MI.`;
7 let msg = `Hello, ${n} ${n1}`;
8 console.log(msg);

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL COMMENTS
PS D:\cdac_tvm\5. Web programming technology\Day_8> node 2.js
Hello, hihi Tim David's unbeaten 32 off 19 went in vain as Lucknow Super Giants defeated Mumbai Indians by 5 runs on Tuesday.
With Mumbai needing 11 runs off the final over, Mohsin Khan conceded only five runs.
Earlier, Marcus Stoinis' fiery 89-run knock guided LSG to 177 for 3 in 20 overs.
Getting an invitation to bat first, LSG were 35 for 3 in 6.1 overs but Stoinis and Krunal Pandya (49) lifted the team with a solid patnership.
Jason Behrendorff (2 for 30) was pick of the bowlers for MI.;
PS D:\cdac_tvm\5. Web programming technology\Day_8> █
```

3 Writing if condition with Boolean String Number object

```
JS 3.js U X JS 4.js U
Day_8 > JS 3.js > [?] person > [?] name
1 //Boolean Condition
2 const flag = true;
3 if(flag){
4     console.log("Boolean condition is true");
5 }else{
6     console.log("Boolean condition is false");
7 }
8
9 //String Condition
10 const text = "Hello";
11 if (text == "Hello"){
12     console.log("String condition is true");
13 }else{
14     console.log("Number condition is false");
15 }
16
17 // Number Condition
18 const num = 10;
19 if (num > 5){
20     console.log("Number condition is true");
21 }else{
22     console.log("Number condition is false");
23 }
24
25
26 //Object Condition
27 const person = {
28     name: "Harsh",
29     age: 22
30 };
31
32 if(person.age >=18){
33     console.log("Object Condition is true");
34 }else{
35     console.log("Object condition is false");
36 }
```

```
PS D:\cdac_tvm\5. Web programming technology\Day_8> node 3.js
Boolean condition is true
String condition is true
Number condition is true
Object Condition is true
PS D:\cdac_tvm\5. Web programming technology\Day_8> █
```

4 Basic for loop Advanced for loop forEach using callback

```
for(let i = 0; i<5; i++){
  console.log(i);
}
console.log("\n");
// Advanced for loop
let a = [1, 2, 3, 4, 5];
for (const num of a) {
  console.log(num);
}

// forEach with callback
let names = ['Harsh', 'Kiran', 'Chaudhari'];
names.forEach((name) => {
  console.log(name);
});
```

```
PS D:\cdac_tvm\5. Web programming technology> cd Day_8
PS D:\cdac_tvm\5. Web programming technology\Day_8> node 4.js
0
1
2
3
4

1
2
3
4
5
Harsh
Kiran
Chaudhari
PS D:\cdac_tvm\5. Web programming technology\Day_8> █
```

5 Create Named Function Overloaded Named Function Named function with default value.

```
// Named Function
function namedFun(p1) {
    console.log(`Named Function`+p1);
}
namedFunDemo(15);

// Overloaded Named Function
function namedFunDemo(p1, p2) {
    console.log("Overloaded Named Function: " + p1 + ", " + p2);
}

// Named Function with Default Value
function namedFunDemoWithDefault(p1 = "Default") {
    console.log("Named Function with Default Value: " + p1);
}
namedFun(15);
namedFunDemo(10, 20);
namedFunDemoWithDefault();
```

```
PS D:\cdac_tvm\5. Web programming technology\Day_8> node 5.js
Overloaded Named Function: 15, undefined
Named Function15
Overloaded Named Function: 10, 20
Named Function with Default Value: Default
PS D:\cdac_tvm\5. Web programming technology\Day_8> █
```

6 Create a named function - Call function with, 2 input param 0 param 1 param 2 param 3 param

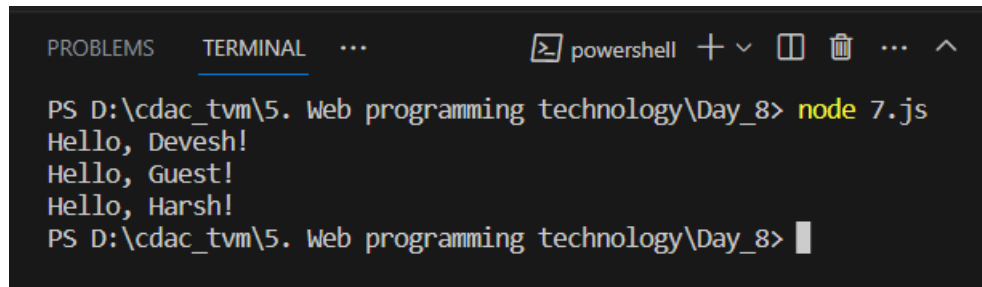
```
function add(a, b) {
    console.log(a + b);
}

add();
add(1);
add(2, 3);
add(4, 5, 6);
```

```
PS D:\cdac_tvm\5. Web programming technology\Day_8> node 6.js
NaN
NaN
5
9
PS D:\cdac_tvm\5. Web programming technology\Day_8> █
```

7 Putting Condition Putting shortcut using || By Default Value

```
function greetUser(name) {  
  const userName = name || 'Guest';  
  console.log(`Hello, ${userName}!`);  
}  
greetUser(`Devesh`, `Harsh`);  
greetUser();  
greetUser(`Harsh`);
```



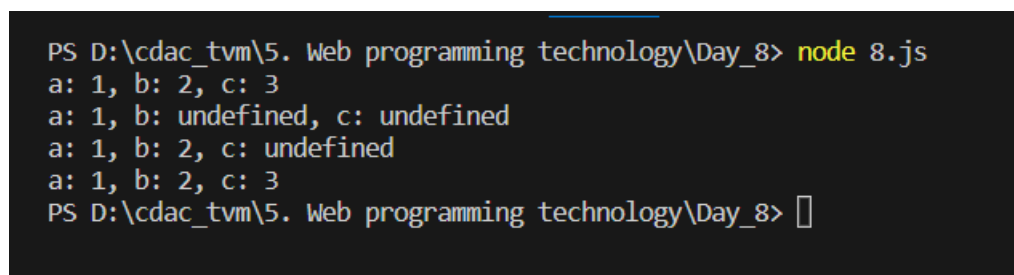
The screenshot shows a PowerShell terminal window with the title bar 'powershell'. The terminal content is as follows:

```
PS D:\cdac_tvm\5. Web programming technology\Day_8> node 7.js  
Hello, Devesh!  
Hello, Guest!  
Hello, Harsh!  
PS D:\cdac_tvm\5. Web programming technology\Day_8> █
```

8 Define function that accepts unordered param - Pass all the param -Pass less param

Define with default value - Accepts less param Pass more param.

```
function processParams(a, b, c) {  
  console.log(`a: ${a}, b: ${b}, c: ${c}`);  
}  
processParams(1, 2, 3); // Pass all parameters  
processParams(1); // Pass less parameters  
processParams(1, 2); // Define with default value  
processParams(1, 2, 3, 4); // Pass more parameters
```



The screenshot shows a PowerShell terminal window with the title bar 'powershell'. The terminal content is as follows:

```
PS D:\cdac_tvm\5. Web programming technology\Day_8> node 8.js  
a: 1, b: 2, c: 3  
a: 1, b: undefined, c: undefined  
a: 1, b: 2, c: undefined  
a: 1, b: 2, c: 3  
PS D:\cdac_tvm\5. Web programming technology\Day_8> █
```

9 Create Anonymous Function Overloaded Anonymous Function Anonymous function with default value

```
// Anonymous Function
let anonymousFun = function(p) {
  console.log("Anonymous Function: " + p);
};
// Overloaded Anonymous Function
const overloadedAnonymousFun = function(p1, p2) {
  console.log("Overloaded Anonymous Function: " + p1 + ", " + p2);
};
// Anonymous Function with Default Value
const anonymousFunWithDefault = function(p = 0) {
  console.log("Anonymous Function with Default Value: " + p);
};
anonymousFun(10);
overloadedAnonymousFun(20, 30);
anonymousFunWithDefault();
anonymousFunWithDefault(40);
```

```
PS D:\cdac_tvm\5. Web programming technology\Day_8> node 9.js
Anonymous Function: 10
Overloaded Anonymous Function: 20, 30
Anonymous Function with Default Value: 0
Anonymous Function with Default Value: 40
PS D:\cdac_tvm\5. Web programming technology\Day_8> █
```

10 Create Arrow Function , Overloaded Arrow Function , Arrow function with default Value

```
// Arrow Function
let arrowFunction = (p) => {
  console.log("Arrow Function: " + p);
};
// Overloaded Arrow Function
let overloadedArrowFunction = (p1, p2) => {
  console.log("Overloaded Arrow Function: " + p1 + ", " + p2);
};
// Arrow Function with Default Value
let arrowFunctionWithDefault = (p = 0) => {
  console.log("Arrow Function with Default Value: " + p);
};
arrowFunction(10);
overloadedArrowFunction(20, 30);
arrowFunctionWithDefault();
arrowFunctionWithDefault(100);
```

```
PS D:\cdac_tvm\5. Web programming technology\Day_8> node 10.js
Arrow Function: 10
Overloaded Arrow Function: 20, 30
Arrow Function with Default Value: 0
Arrow Function with Default Value: 100
PS D:\cdac_tvm\5. Web programming technology\Day_8> █
```