

TASK-1

```
1. What will be the output of this code?  
console.log(x);  
var x = 5;
```

Output:

```
PS C:\Users\surya\OneDrive\Desktop\js> node js2task.js  
undefined  
PS C:\Users\surya\OneDrive\Desktop\js> |
```

Explanation:

The output will be undefined. This is because of variable hoisting in JavaScript, where the declaration `var x` is moved to the top but not its initialization. So, when `console.log(x)` is executed, `x` is declared but still uninitialized, resulting in undefined.

```
2. What will be the output of this code?  
console.log(a);  
var a;
```

Output:

```
PS C:\Users\surya\OneDrive\Desktop\js> node js2task.js  
undefined
```

Explanation:

The output will be undefined. Due to variable hoisting, the declaration `var a` is moved to the top, but since it is not initialized, its value remains undefined when `console.log(a)` is called.

```
3. What will be the output of this code?  
console.log(b);  
b = 10;  
var b;
```

Output:

```
PS C:\Users\surya\OneDrive\Desktop\js> node js2task.js  
undefined
```

Explanation:

The output will be undefined. Due to hoisting, the declaration `var b` is moved to the top, but the assignment `b = 10` happens after the `console.log(b)` call. At the time of the log, `b` is declared but uninitialized, so it results in undefined.

4. What will happen here?

```
console.log(c); // Attempting to access 'c'
```

Output:

```
ReferenceError: c is not defined  
    at Object.<anonymous> (C:\Users\surya\OneDrive\Desktop\js\js2task.js:15:13)
```

Explanation:

The output will be a ReferenceError. Since c is not declared anywhere in the code, JavaScript cannot find it in the scope when console.log(c) is executed, leading to the error.

6. What will be the output of this code?

```
console.log(e);  
var e = 10;  
console.log(e);  
e = 20;  
console.log(e);
```

Output:

```
PS C:\Users\surya\OneDrive\Desktop\js> node js2task.js  
undefined  
10  
20
```

Explanation:

1. undefined (due to hoisting, e is declared but uninitialized at the first console.log).
2. 10 (after the variable e is initialized).
3. 20 (after e is reassigned to 20).

7. What will be the output of this code?

```
console.log(f);  
var f = 100;  
var f;  
console.log(f);
```

Output:

```
PS C:\Users\surya\OneDrive\Desktop\js> node js2task.js  
undefined  
100
```

Explanation:

The first console.log(f) outputs undefined because of JavaScript's hoisting; the variable f is declared but not yet assigned a value. The second console.log(f) outputs 100 since the variable f has been assigned the value 100. Redeclaring var f doesn't affect the value, as var allows redeclaration but keeps the same value.

8. What will be the output of this code?

```
console.log(g);  
var g = g + 1;  
console.log(g);
```

Output:

```
PS C:\Users\surya\OneDrive\Desktop\js> node js2task.js  
undefined  
NaN
```

Explanation:

- The first console.log(g) outputs undefined because g is hoisted but not initialized.
- Next, g = g + 1 tries to add 1 to undefined, resulting in NaN.
- The second console.log(g) outputs NaN.

9. What will be the output of this code?

```
var h;  
console.log(h);  
h = 50;  
console.log(h);
```

Output:

```
PS C:\Users\surya\OneDrive\Desktop\js> node js2task.js  
undefined  
50
```

Explanation:

The first console.log(h); prints undefined because the variable h is declared but not yet initialized. The variable h is assigned the value 50. The second console.log(h); prints 50, as h now holds that value.

10. What will be the output of this code?

```
console.log(i);  
i = 10;  
var i = 5;  
console.log(i);
```

Output:

```
PS C:\Users\surya\OneDrive\Desktop\js> node js2task.js  
undefined  
5
```

Explanation:

The first console.log(i) outputs undefined because var i is hoisted but not initialized. The variable i is then assigned 10, but immediately re-assigned to 5 due to var i = 5. The second console.log(i) outputs 5.