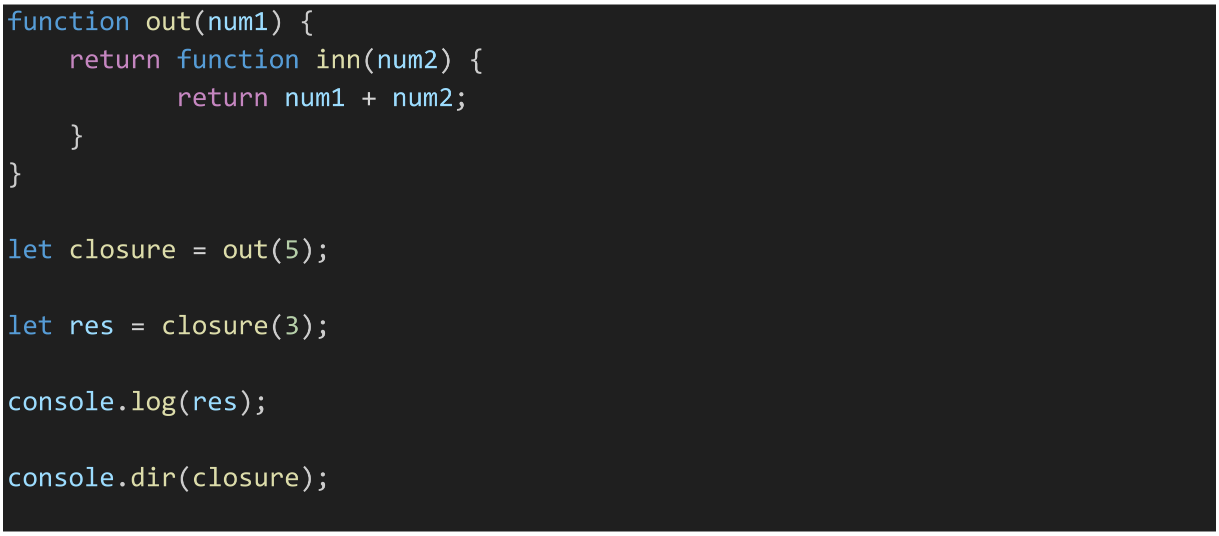
Task#1: Closures :-

Closure happends when you create one function or more (Inner Function) inside a function (Outer Function) and it allows you to access the outer scope from the inner function even after the outer function is executed.

Example :-



```
function out(num1) {
    return function inn(num2) {
           return num1 + num2;
    }
}

let closure = out(5);

let res = closure(3);

console.log(res);

console.dir(closure);
```

Task#3: Hoisting :-

Hoisting is a process that moves the declaration of all Variables (only var variables), Functions and Classes to the top of their scope. This means that you can initialize the value of a variable and call it or use it in your process before even declaring it. You can also reference a variable in its scope before declaring it and it will not appear a reference error but the value will be undefined.

Example :-

Hoisted Variable :-

```
x = 5;
console.log(x); // 5
var x;
```

Hoisted Reference :-

```
console.log(x); // undefined
var x = 5;
```

Hoisted Function :-

```
hoisted(); // "Hello World!"

function hoisted() {
  console.log("Hello World!");
}
```