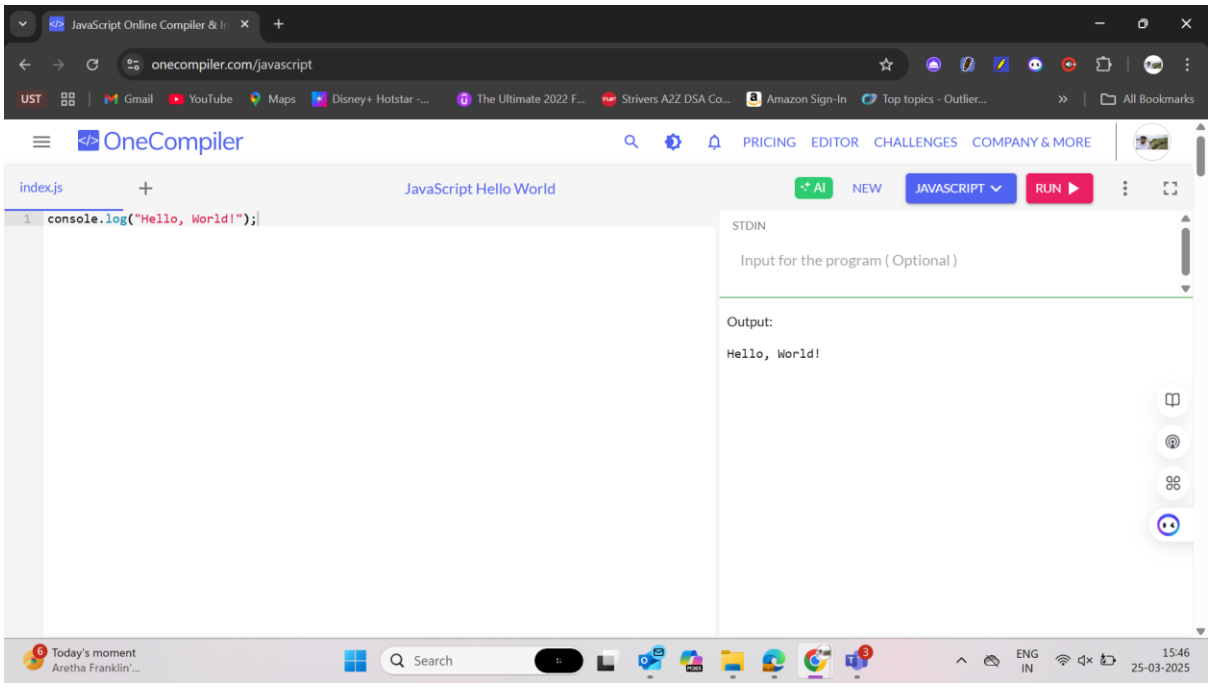
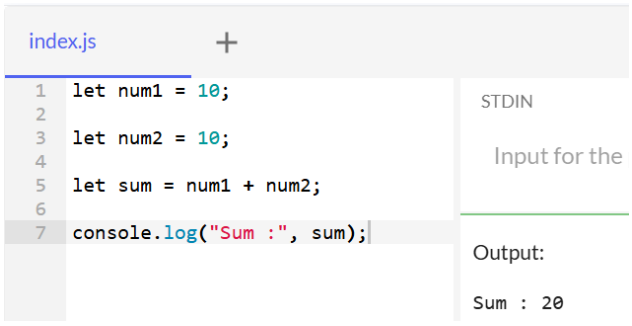
1.Create a simple web page displaying "Hello, World!" using JavaScript.

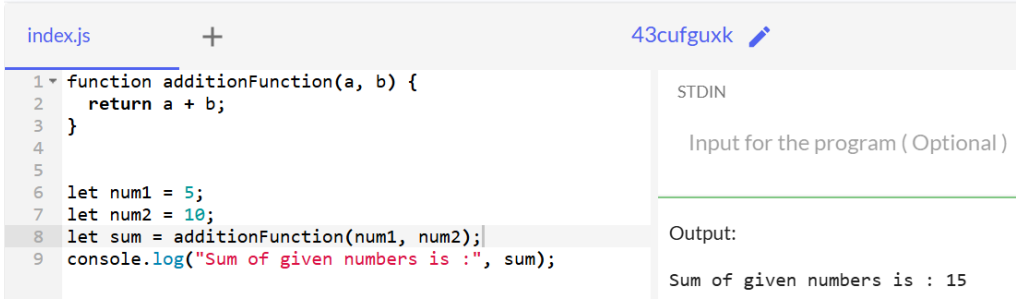


2.

Using + Operator



Using function



Using Arrow function

index.js 43cufguxk

```
let addition = (a, b) => a + b;


let num1 = 25;
let num2 = 25;
let sum = addition(num1, num2);
console.log("Sum of given numbers is :", sum);
```

STDIN

Input for the program (Optior

Output:

```
Sum of given numbers is : 50
```

Using Addition Assignment (+=) Operator

index.js 43cufguxk

```
let num1 = 15;
let num2 = 10;
num1 += num2;

console.log("Sum of the given number is :", num1);
```

STDIN

Input for the program (Optional)

Output:

```
Sum of the given number is : 25
```

3. Convert a regular function to an arrow function.

Arrow Function without Parameters

index.js 43cufguxk

```
const ust = () => {
console.log( "Hi from ust!" );
}
ust();
```

STDIN

Input for the pr

Output:

```
Hi from ust!
```

Arrow Function with Single Parameters

index.js 43cufguxk

```
const square = x => x*x;
console.log(square(4));
```

STDIN

Input for t

Output:

```
16
```

Arrow Function with Multiple Parameters

index.js 43cufguxk

```
const gfg = ( x, y, z ) => {
console.log( x + y + z )
}
gfg( 10, 20, 30 );
```

Output:

60

Arrow Function with Default Parameters

index.js 43cufguxk

```
const gfg = ( x, y, z = 30 ) => {
console.log( x + " " + y + " " + z);
}
gfg( 10, 20 );
```

Output:

10 20 30

Return Object Literals

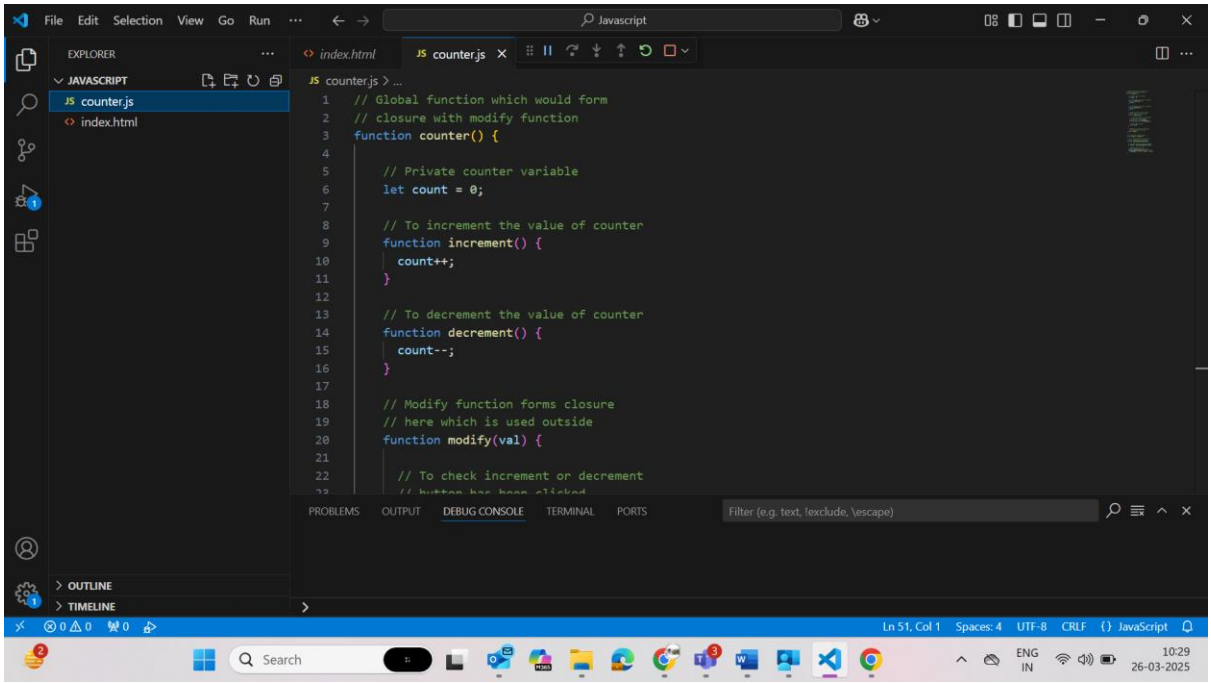
index.js 43cufguxk AI NEW

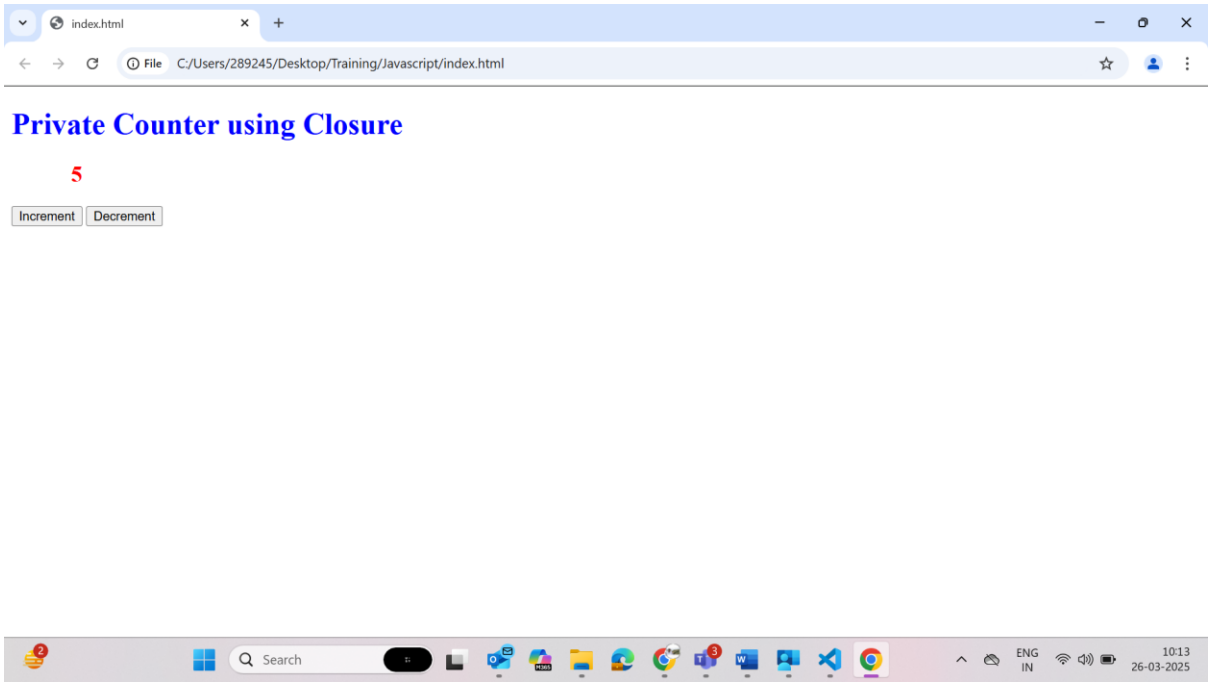
```
const makePerson = (firstName, lastName) =>({first: firstName, last: lastName});

console.log(makePerson("Pankaj", "Bind"));
```

Output:

{ first: 'Pankaj', last: 'Bind' }

4. Create a counter function using closures.





5. Define an object representing a car with properties and a method.

Creating object with a constructor

index.js + 43cufguxk

```
1  // Simple function
2  function vehicle(name, maker, engine) {
3      this.name = name;
4      this.maker = maker;
5      this.engine = engine;
6  }
7  // New keyword to create an object
8  let car = new vehicle('GT', 'BMW', '1998cc');
9  // Property accessors
10 console.log(car.name);
11 console.log(car.maker);
12 console.log(car['engine']);
```

Output:

```
GT
BMW
1998cc
```

Using object literals

index.js + 43cufguxk

```
1  // Creating js objects with object literal
2  let car = {
3      name: 'GT',
4      maker: 'BMW',
5      engine: '1998cc'
6  };
7  // Property accessor
8  console.log(car.name); //dot notation
9  console.log(car['maker']); //bracket notation
```

Output:

```
GT
BMW
```

Creating object with Object.create() Method

index.js 43cufguxk AI NEW

```
const coder = {
    isStudying : false,
    printIntroduction : function(){
        console.log(`My name is ${this.name}. Am I studying?: ${this.isStudying}`)
    }
};
const me = Object.create(coder);
me.name = 'Sahil';
me.isStudying = true;
me.printIntroduction();
```

Output:

```
My name is Sahil. Am I studying?: true
```

Using es6 classes

index.js 43cufguxk

```
class Vehicle {
  constructor(name, maker, engine) {
    this.name = name;
    this.maker =  maker;
    this.engine = engine;
  }
}

let car1 = new Vehicle('GT', 'BMW', '1998cc');

console.log(car1.name);  //GT
```

Output:

```
GT
```