Java Script Assignment

```
console.log("Hello, World!");
```

Output:

Hello, World!

On Client Side

Using the document.write() method in JavaScript allows you to display “Hello World” on the client side HTML document.

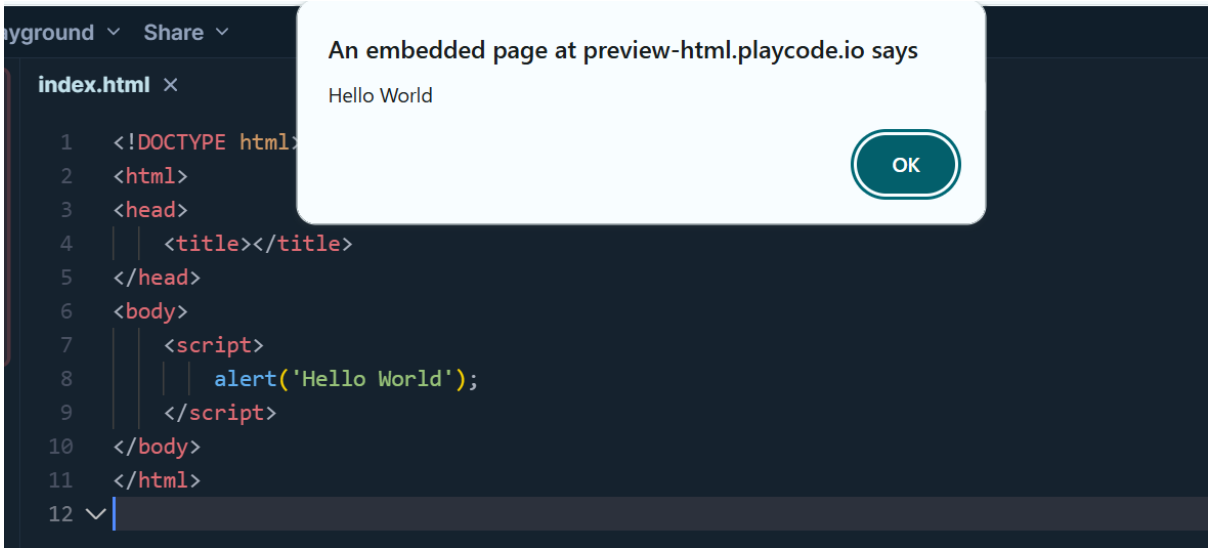
index.html ×

```
<!DOCTYPE html>
<html>
<head>
    <title></title>
</head>
<body>
    <script>
        document.write('Hello World');
    </script>
</body>
</html>
```

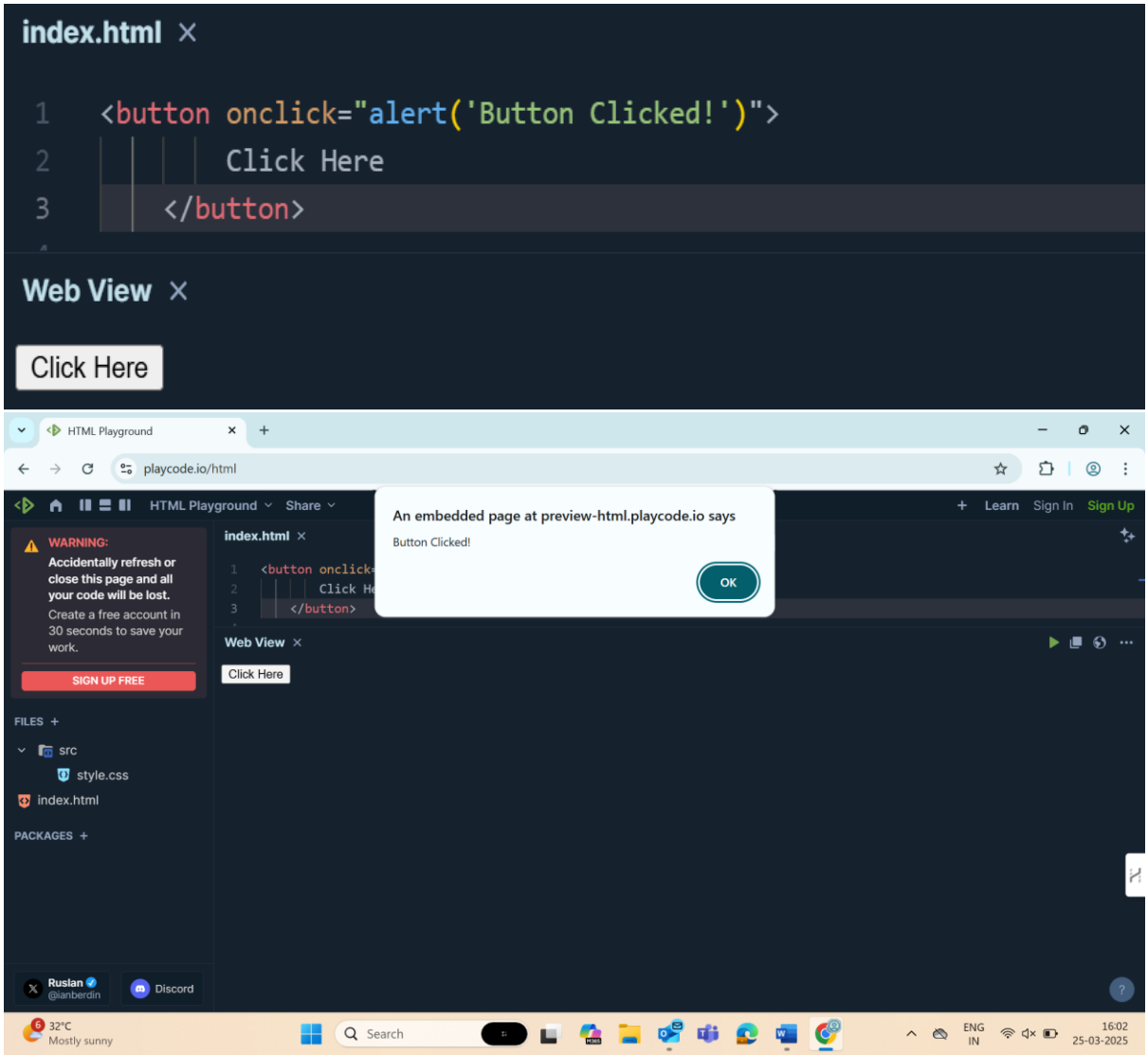
Web View ×

Hello World

Pop-Up on Client Side

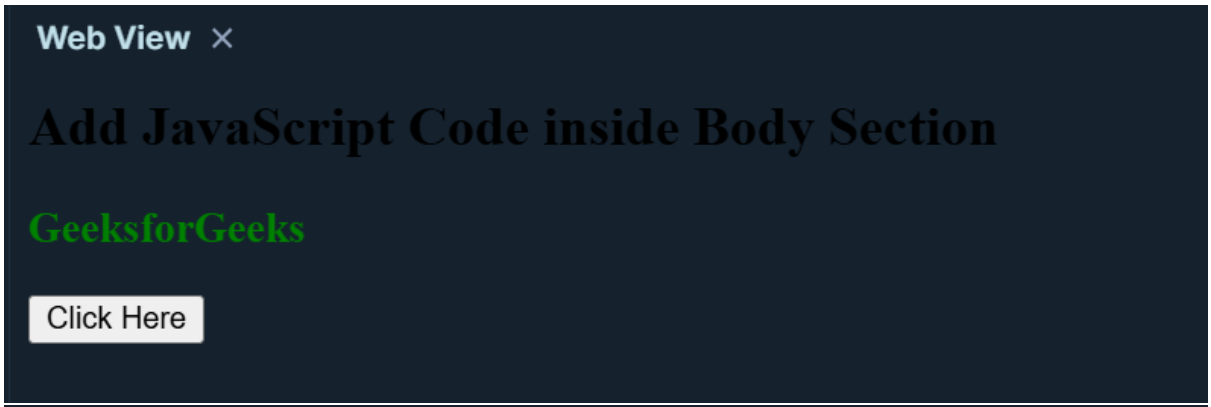


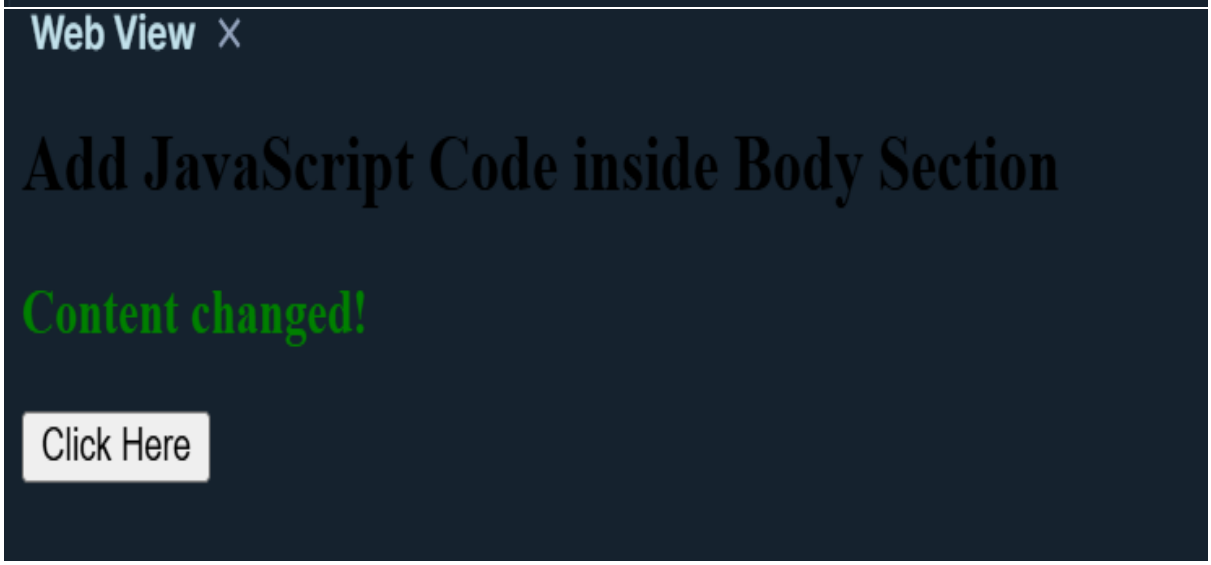
Inline JavaScript



JavaScript Code Inside <body> Tag

```html
<body>
    <h2>
        Add JavaScript Code
        inside Body Section
    </h2>
    <h3 id="demo" style="color:green;">
        GeeksforGeeks
    </h3>
    <button type="button" onclick="myFun()">
        Click Here
    </button>
    <script>
        function myFun() {
            document.getElementById("demo")
                .innerHTML = "Content changed!";
        }
    </script>
</body>
```





Sum of two numbers using + Operator

```
let num1 = 10;
let num2 = 10;
let sum = num1 + num2;
console.log("Sum :", sum);
```

Output:

```
Sum : 20
```

Using function

```
function additionFunction(a, b) {
    return a + b;
}

let num1 = 5;
let num2 = 10;
let sum = additionFunction(num1, num2);
console.log("Sum of given numbers is :", sum);
```

Output:

```
Sum of given numbers is : 15
```

Using Arrow function

```
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);
```

Output:

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

Using Addition Assignment (+=) Operator

```
let num1 = 15;
let num2 = 10;
num1 += num2;
console.log("Sum of the given number is :", num1);
```

Output:

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

3. Convert a regular function to an arrow function.

```
const add = (a, b) => a + b;
console.log(add(5, 3));
```

Output:

```
8
```

1. Arrow Function without Parameters

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

Output:

```
Hi from Barathkumar!
```

2. Arrow Function with Single Parameters

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

Output:

16

3. Arrow Function with Multiple Parameters

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

Output:

60

4. Arrow Function with Default Parameters

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

Output:

10 20 30

5. Return Object Literals

```
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.

```
function outer() {
  let str = "Barath";
  function inner()
  {
    console.log(str);

  }
  return inner;

}
const fun = outer(); fun();
```

Output:

Barath

1. **Create a counter function using closures.**

File Edit Selection View Go Run ... Codes

EXPLORER

CODES

counter.js

index.html

Welcome | index.html | counter.js

index.html > html > body

```html
<!DOCTYPE html>
<html lang="en">

<head>
    <script src="counter.js"></script>
</head>

<body>
    <h1 style="color: blue;">
        Private Counter using Closure
    </h1>

    <div id="counter_div" style=
        "margin-left: 5%; color: red;">

        <h2>0</h2>
    </div>

    <button onclick="counterHandler(this)"
        value="1">
        Increment
    </button>

    <button onclick="counterHandler(this)"
        value="0">
        Decrement
    </button>
</body>
```

OUTLINE

TIMELINE

Ln 18, Col 3 Spaces: 4 UTF-8 CRLF HTML

32°C Mostly sunny Search ENG IN 16:48 25-03-2025

File Edit Selection View Go Run ... Codes

EXPLORER

CODES

counter.js

index.html

Welcome | index.html | counter.js

counter.js > counterHandler

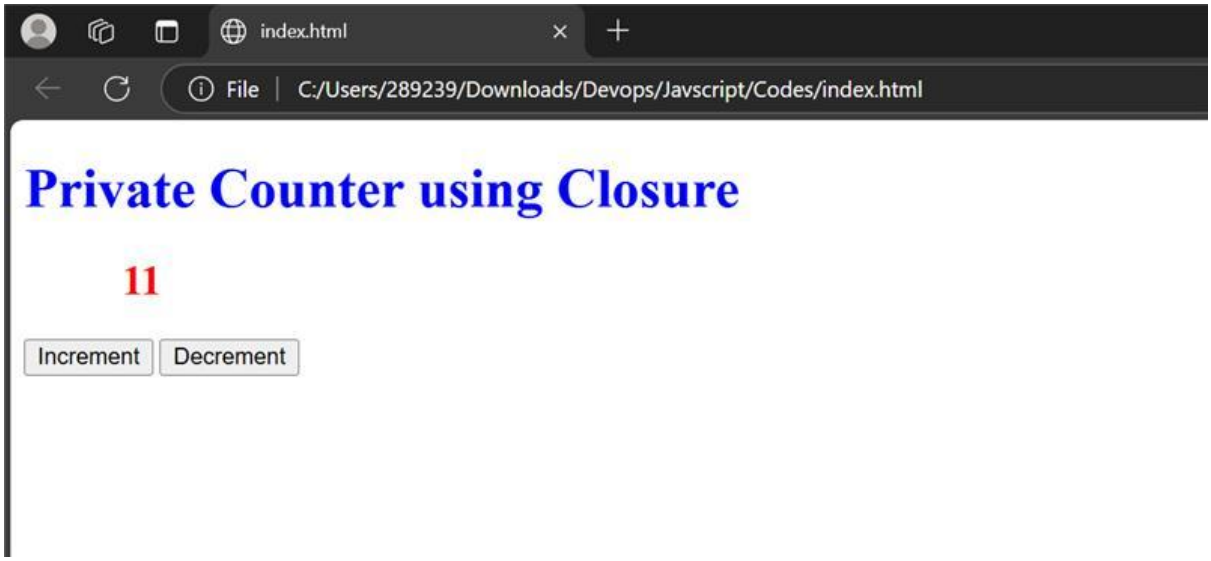
```js
function counter() {
    let count = 0;
    function increment() {
      count++;
    }
    function decrement() {
      count--;
    }
    function modify(val) {
      if (val === "1") increment();
      else if (val === "0") decrement();

      return count;
    }

    return modify;
  }

  const closure = counter();
  function counterHandler(objButton) {

    let count = closure(objButton.value);
    document.getElementById("counter_div")
      .innerHTML = "<h2>" + count + "</h2>";
  }
```

OUTLINE

TIMELINE

Ln 22, Col 43 Spaces: 4 UTF-8 CRLF {} JavaScript

32°C Mostly sunny Search ENG IN 16:48 25-03-2025



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

```
function vehicle(name, maker, engine) {
    this.name = name;
    this.maker = maker;
    this.engine = engine;
}
let car = new vehicle('GT', 'BMW', '1998cc');
console.log(car.name);
console.log(car.maker);
console.log(car['engine']);
```

Output:

```
GT
BMW
1998cc
```

Using object literals

```
let car = {
    name: 'GT',
    maker: 'BMW',
    engine: '1998cc'
};
console.log(car.name);
console.log(car['maker']);
```

Output:

```
GT
BMW
```

```
let car = {
    name: 'GT',
    maker: 'BMW',
    engine: '1998cc'
};
car.brakesType = 'All Disc';
console.log(car);
```

Output:

```
{ name: 'GT', maker: 'BMW', engine: '1998cc', brakesType: 'All Disc' }
```

```
let car = {
    name : 'GT',
    maker : 'BMW',
    engine : '1998cc',
    start : function(){
        console.log('Starting the engine...');
    }
};
car.start();
car.stop = function() {
    console.log('Applying Brake...');
}
car.stop();
```

Output:

```
Starting the engine...
Applying Brake...
```

Creating object with Object.create() Method

```
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 = 'Mukul';
me.isStudying = true;
me.printIntroduction();
```

Output:

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

```
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);
```

Output:

```
GT
```