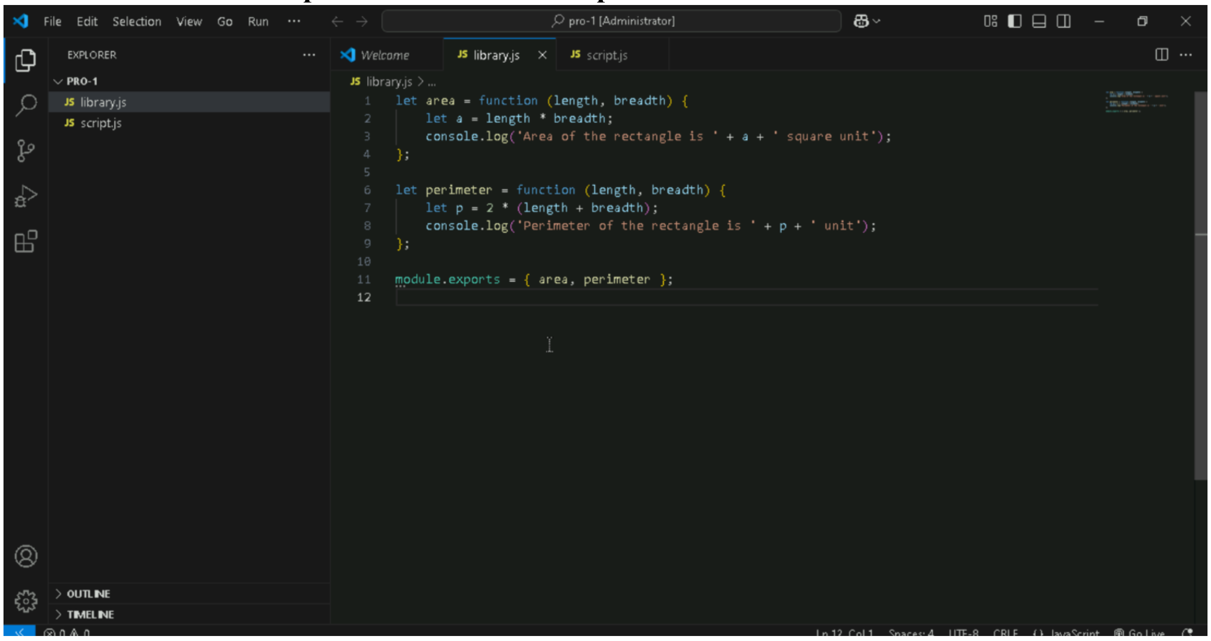
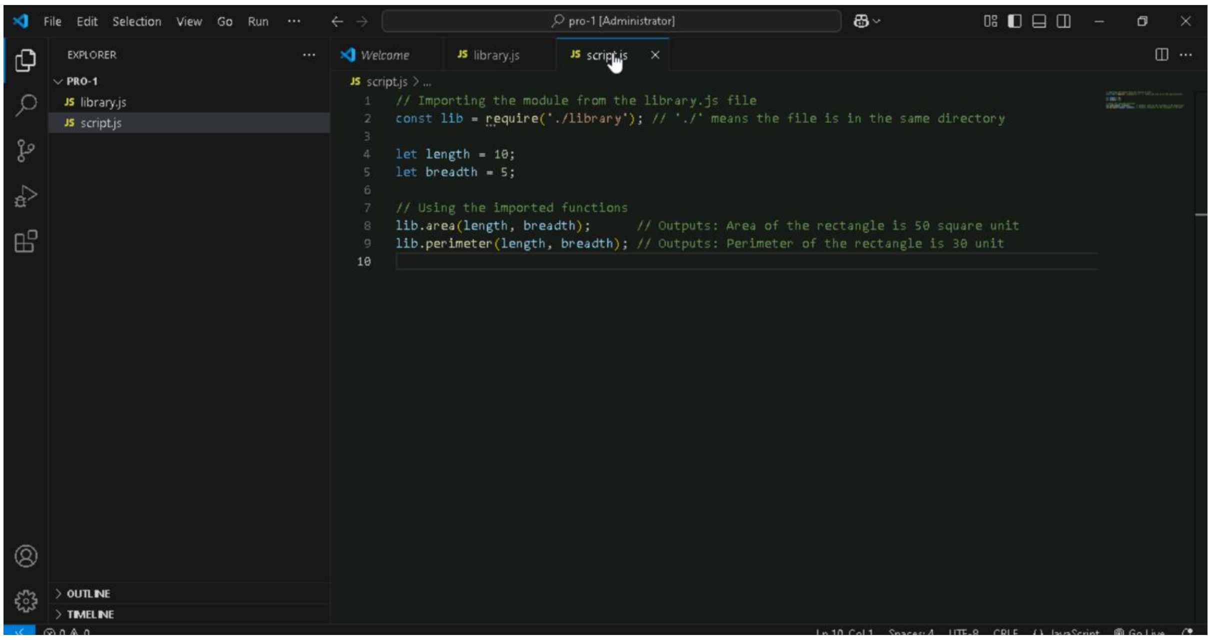
JAVASCRIPT PROJECT

Create a module that exports a function and import it into another file.



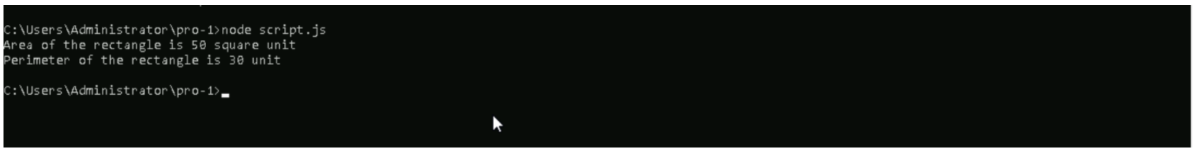
```
let area = function (length, breadth) {
    let a = length * breadth;
    console.log('Area of the rectangle is ' + a + ' square unit');
};

let perimeter = function (length, breadth) {
    let p = 2 * (length + breadth);
    console.log('Perimeter of the rectangle is ' + p + ' unit');
};

module.exports = { area, perimeter };
```

\



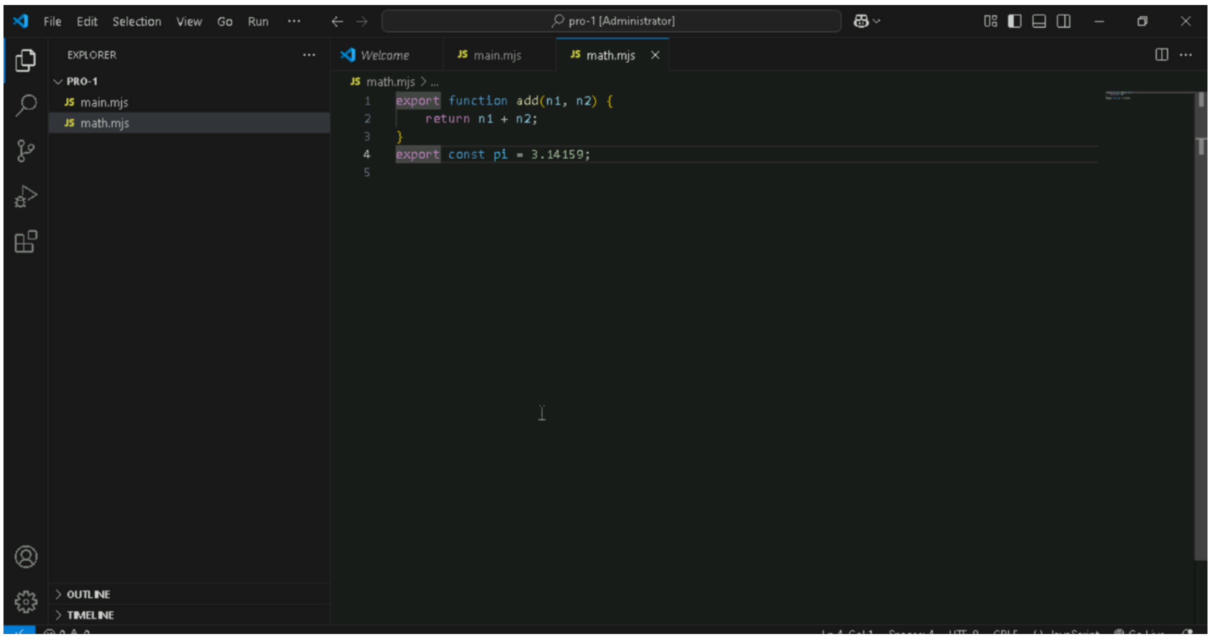
```
// Importing the module from the library.js file
const lib = require('./library'); // './' means the file is in the same directory

let length = 10;
let breadth = 5;

// Using the imported functions
lib.area(length, breadth);      // Outputs: Area of the rectangle is 50 square unit
lib.perimeter(length, breadth); // Outputs: Perimeter of the rectangle is 30 unit
```

OUTPUT

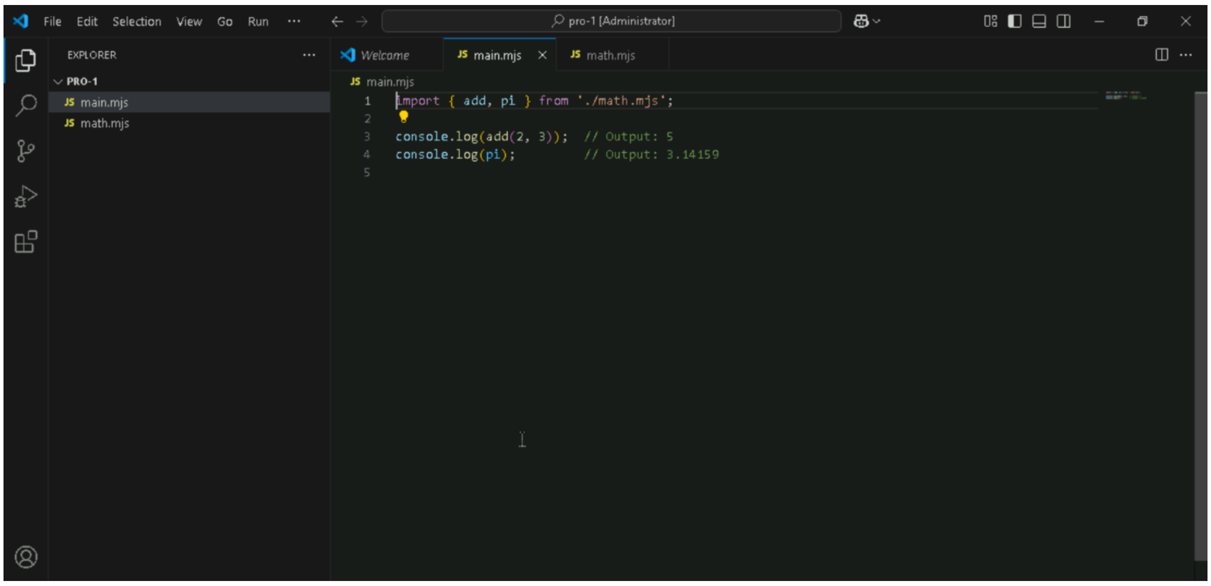


```
C:\Users\Administrator\pro-1>node script.js
Area of the rectangle is 50 square unit
Perimeter of the rectangle is 30 unit

C:\Users\Administrator\pro-1>_
```

Convert a JavaScript object to JSON and back using JSON.stringify() and JSON.parse().

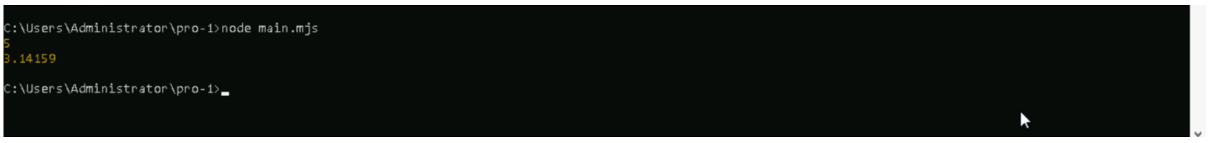


```
export function add(n1, n2) {
    return n1 + n2;
}
export const pi = 3.14159;
```



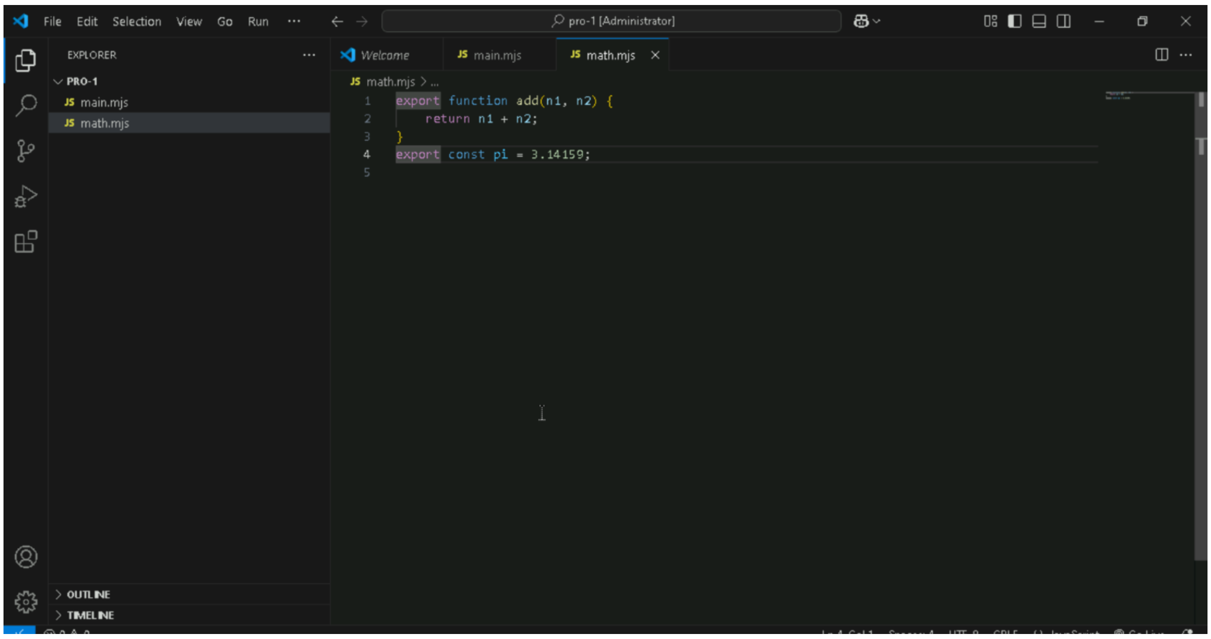
```
import { add, pi } from './math.mjs';

console.log(add(2, 3));  // Output: 5
console.log(pi);         // Output: 3.14159
```

OUTPUT

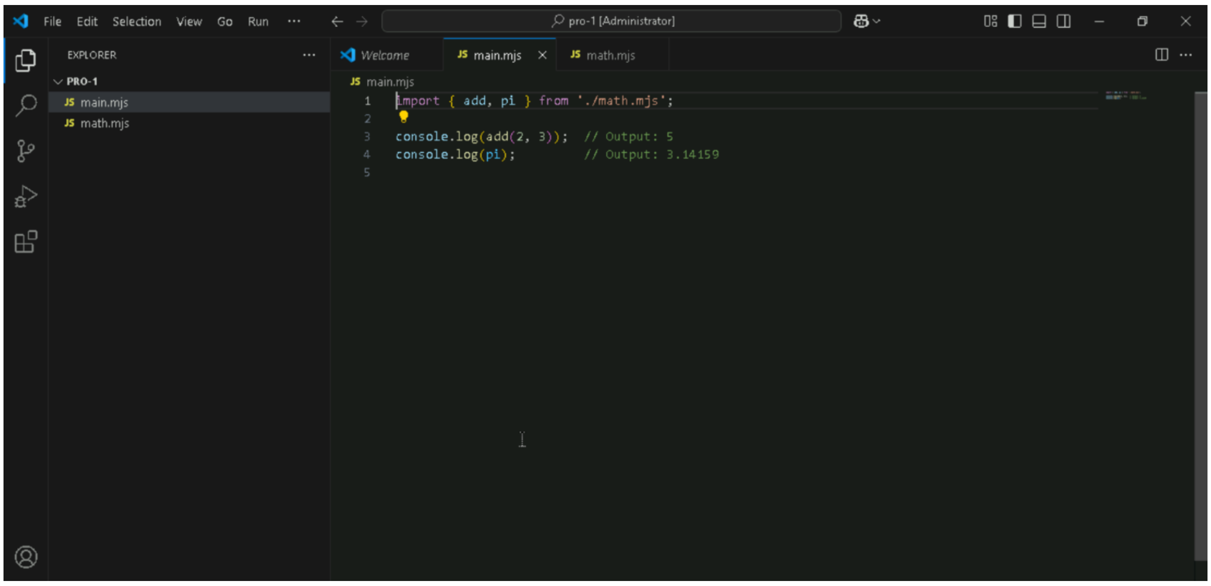


```
C:\Users\Administrator\pro-1>node main.mjs
5
3.14159

C:\Users\Administrator\pro-1>
```

Using require() Method (CommonJS)

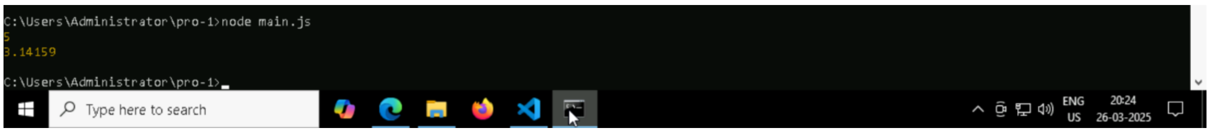


```
export function add(n1, n2) {
    return n1 + n2;
}
export const pi = 3.14159;
```



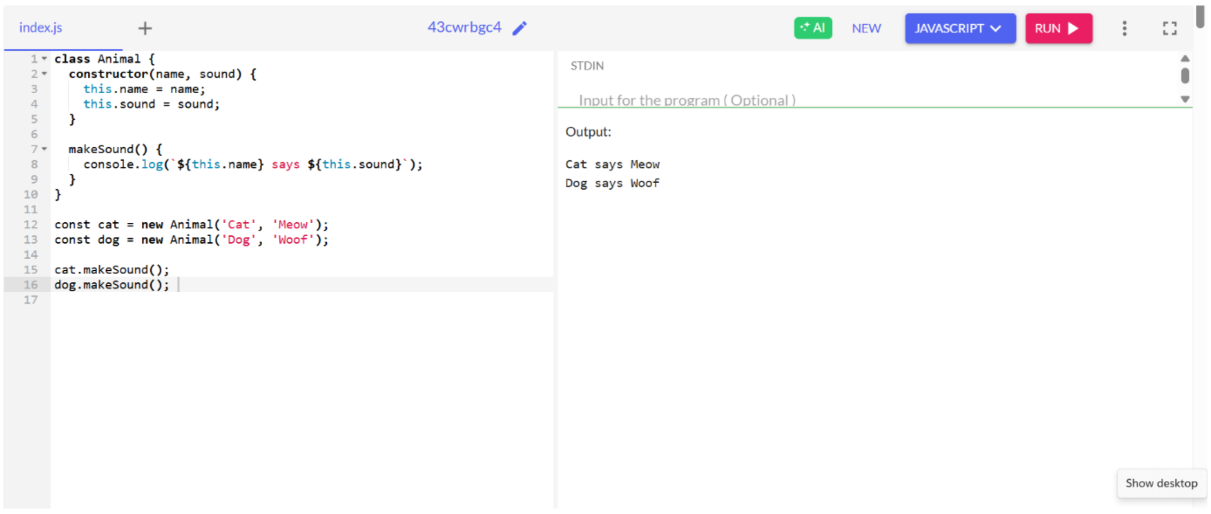
```
import { add, pi } from './math.mjs';

console.log(add(2, 3));  // Output: 5
console.log(pi);         // Output: 3.14159
```

OUTPUT



```
C:\Users\Administrator\pro-1>node main.js
5
3.14159

C:\Users\Administrator\pro-1>_
```

Define a class Animal with properties and a method, and create instances

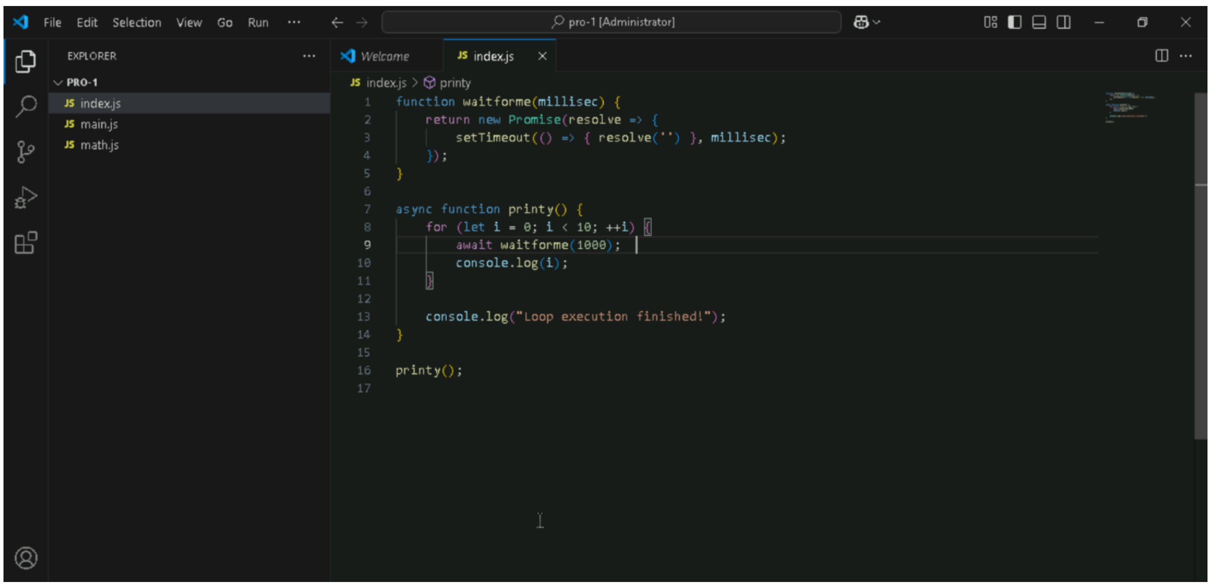


```
class Animal {
  constructor(name, sound) {
    this.name = name;
    this.sound = sound;
  }

  makeSound() {
    console.log(`${this.name} says ${this.sound}`);
  }
}

const cat = new Animal('Cat', 'Meow');
const dog = new Animal('Dog', 'Woof');

cat.makeSound();
dog.makeSound();
```

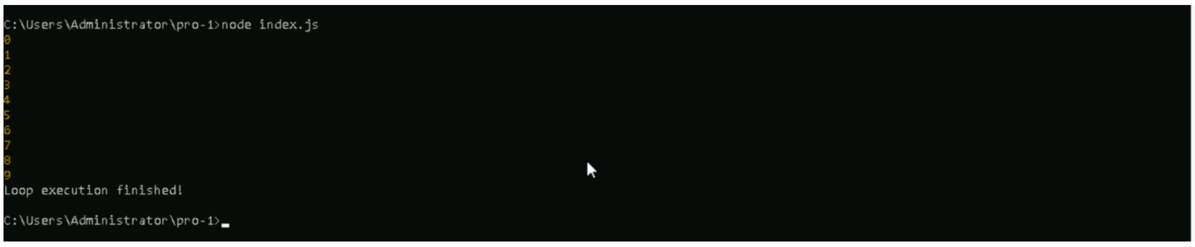
Output:

```
Cat says Meow
Dog says Woof
```

Write a promise that resolves after a delay and convert it into an async function using async/await.



```
function waitforme(millisec) {
    return new Promise(resolve => {
        setTimeout(() => { resolve('') }, millisec);
    });
}

async function printy() {
    for (let i = 0; i < 10; ++i) {
        await waitforme(1000);
        console.log(i);
    }

    console.log("Loop execution finished!");
}

printy();
```



```
C:\Users\Administrator\pro-1>node index.js
0
1
2
3
4
5
6
7
8
9
Loop execution finished!

C:\Users\Administrator\pro-1>
```