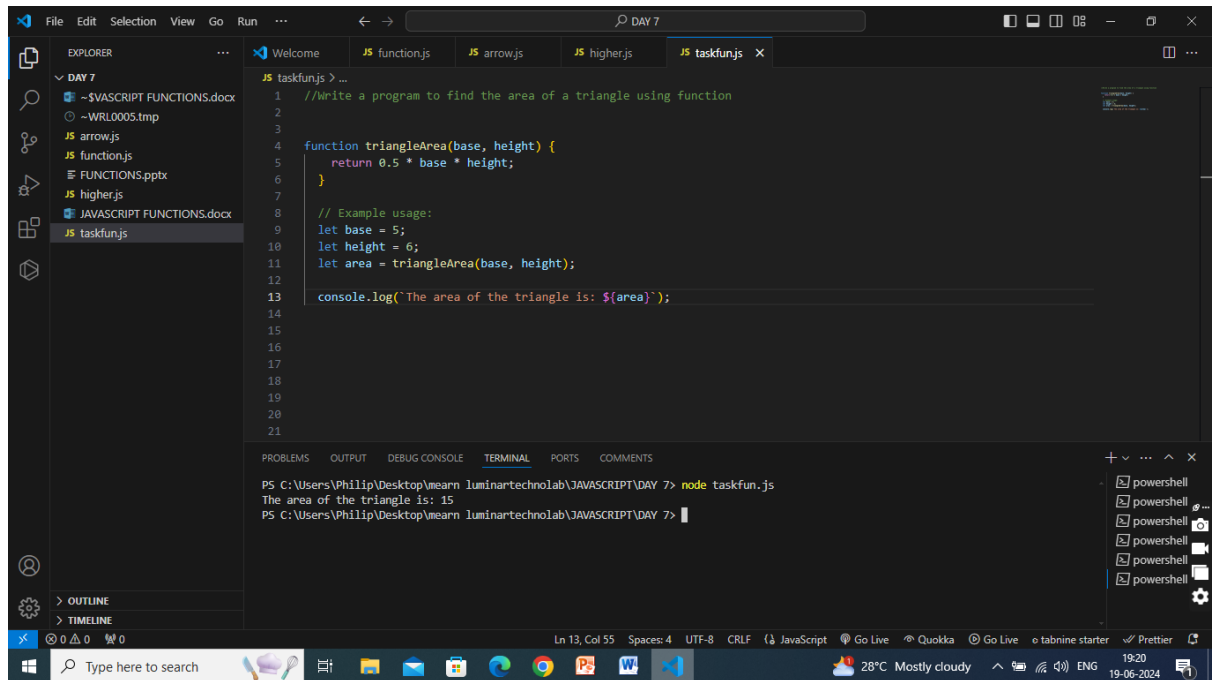


TASKS

1. Write a program to find the area of a triangle using function.

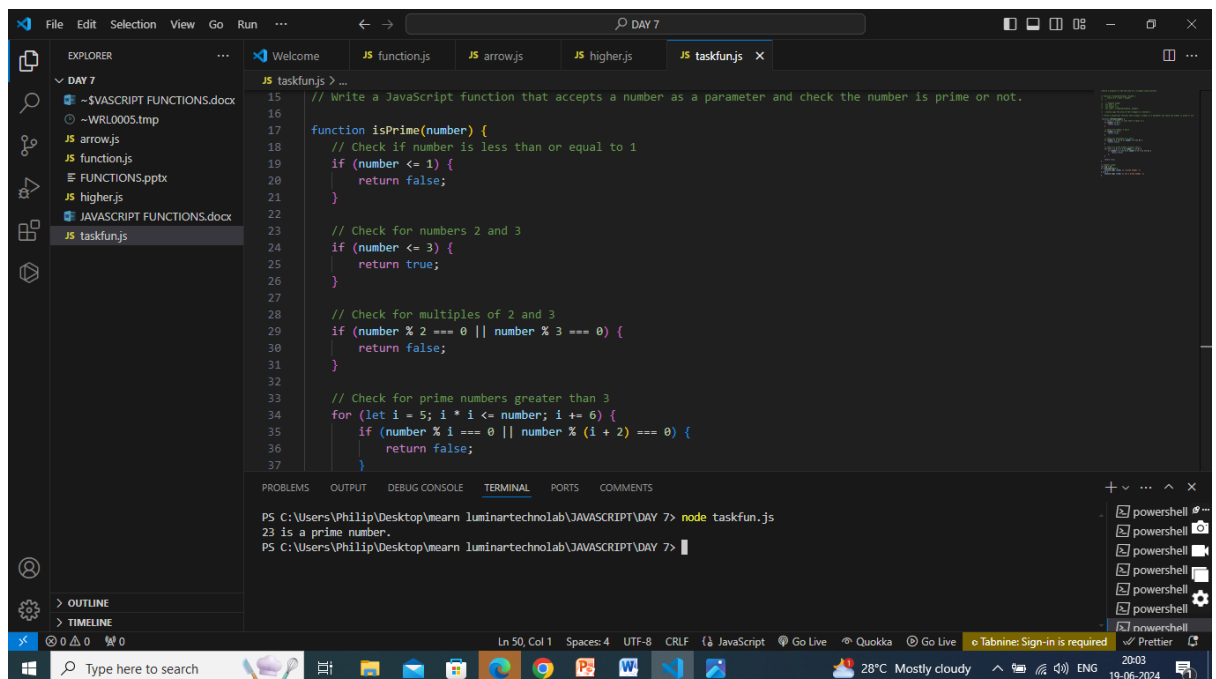


The screenshot shows the Visual Studio Code editor with a file named `taskfun.js` open. The code defines a function `triangleArea` that calculates the area of a triangle given its base and height. The function is called with base 5 and height 6, and the result is logged to the console.

```
1 //Write a program to find the area of a triangle using function
2
3
4 function triangleArea(base, height) {
5     return 0.5 * base * height;
6 }
7
8 // Example usage:
9 let base = 5;
10 let height = 6;
11 let area = triangleArea(base, height);
12
13 console.log('The area of the triangle is: ${area}');
```

The terminal output shows the command `node taskfun.js` being executed, resulting in the output: `The area of the triangle is: 15`.

2. Write a JavaScript function that accepts a number as a parameter and check the number is prime or not.

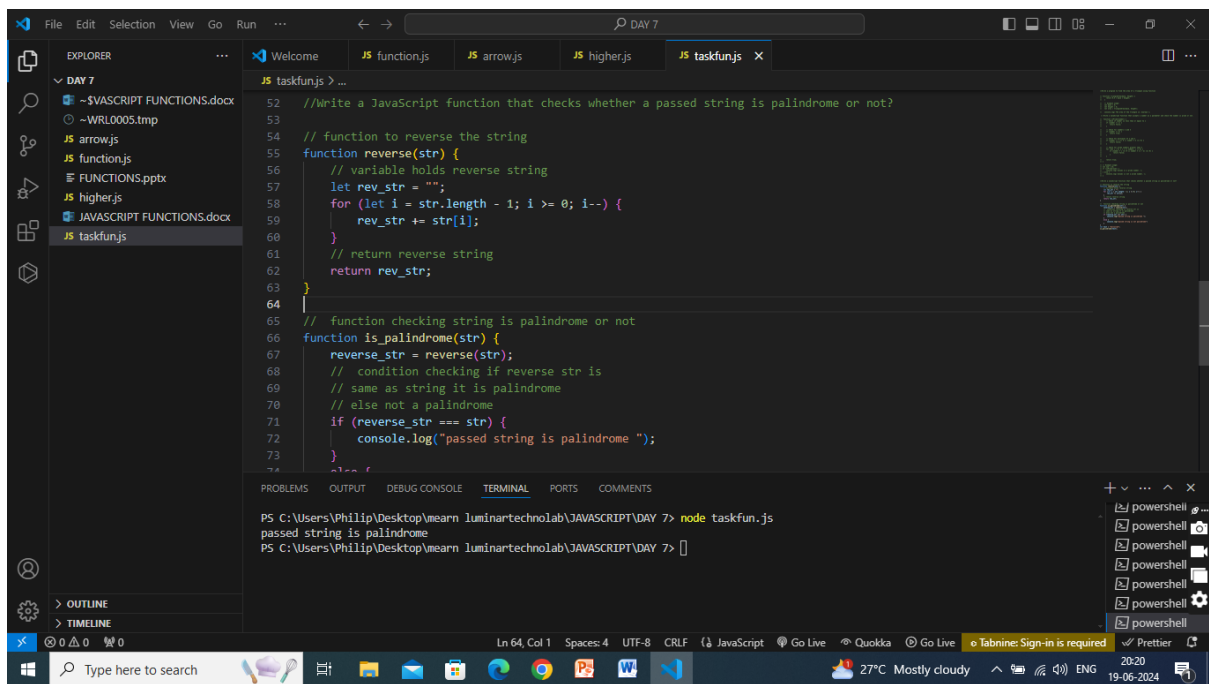


The screenshot shows the Visual Studio Code editor with a file named `taskfun.js` open. The code defines a function `isPrime` that checks if a number is prime or not. The function is called with the number 23, and the result is logged to the console.

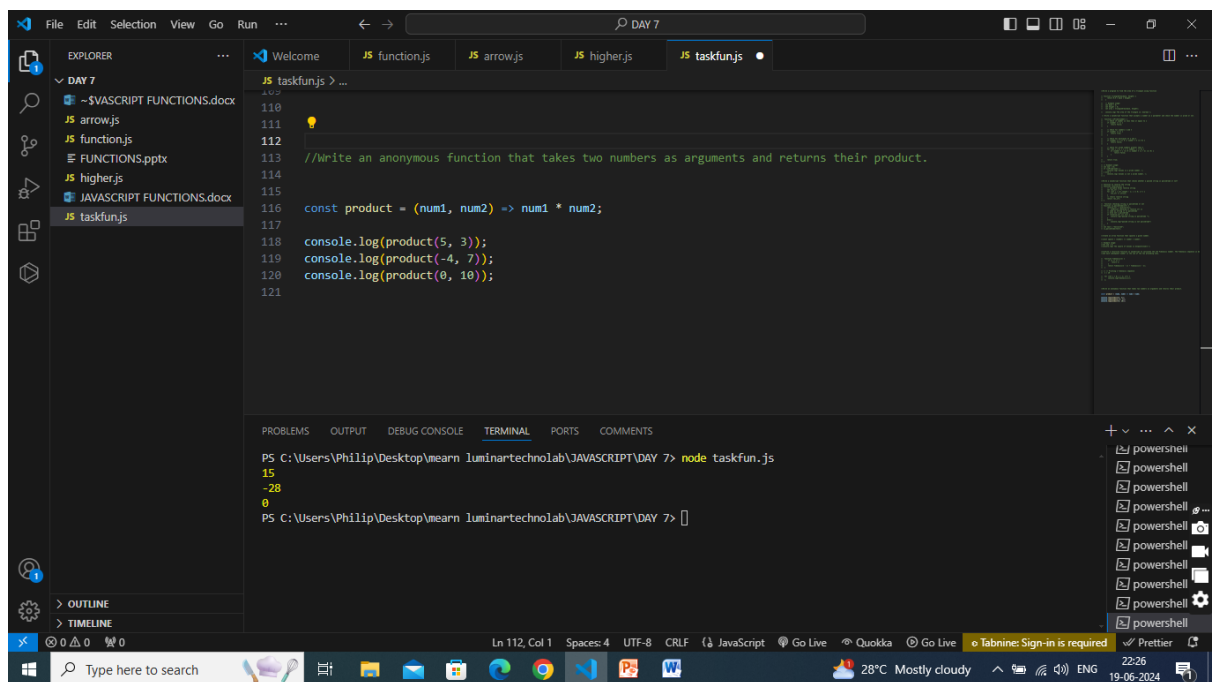
```
15 // Write a JavaScript function that accepts a number as a parameter and check the number is prime or not.
16
17 function isPrime(number) {
18     // Check if number is less than or equal to 1
19     if (number <= 1) {
20         return false;
21     }
22
23     // Check for numbers 2 and 3
24     if (number <= 3) {
25         return true;
26     }
27
28     // Check for multiples of 2 and 3
29     if (number % 2 === 0 || number % 3 === 0) {
30         return false;
31     }
32
33     // Check for prime numbers greater than 3
34     for (let i = 5; i * i <= number; i += 6) {
35         if (number % i === 0 || number % (i + 2) === 0) {
36             return false;
37         }
38     }
39
40     return true;
41 }
```

The terminal output shows the command `node taskfun.js` being executed, resulting in the output: `23 is a prime number.`

3. Write a JavaScript function that checks whether a passed string is palindrome or not?



- Write an anonymous function that takes two numbers as arguments and returns their product.



- Create an arrow function that squares a given number.

```
JS taskfunjs > [0] num
77 // }
78 // let test = "hellolleh";
79 // is_palindrome(test);
80
81
82
83 //Create an arrow function that squares a given number.
84
85 const square = (number) => number * number;
86
87 // Example usage:
88 let num = 6;
89 console.log(`The square of ${num} is ${square(num)}`);
90
91
```

```
PS C:\Users\Philip\Desktop\mearn_luminartechnolab\JAVASCRIPT\DAY 7> node taskfun.js
The square of 6 is 36
PS C:\Users\Philip\Desktop\mearn_luminartechnolab\JAVASCRIPT\DAY 7>
```

6. Write a recursive function in JavaScript to calculate the nth Fibonacci number. The Fibonacci sequence is defined as follows: the first two numbers are 0 and 1, and each subsequent number is the sum of the two preceding ones.

```
JS taskfunjs > ...
92 //6. Write a recursive function in JavaScript to calculate the nth Fibonacci number. The Fibonacci sequence is defined
93 //and each subsequent number is the sum of the two preceding ones.
94
95
96 function fibonacci(n) {
97     if (n <= 1) {
98         return n;
99     }
100     return fibonacci(n - 1) + fibonacci(n - 2);
101 }
102
103 // Printing n fibonacci sequence
104 n = 10
105
106 for (let i = 0; i < n; i++) {
107     console.log(fibonacci(i));
108 };
```

```
PS C:\Users\Philip\Desktop\mearn_luminartechnolab\JAVASCRIPT\DAY 7> node taskfun.js
0
1
1
2
3
5
8
13
21
34
PS C:\Users\Philip\Desktop\mearn_luminartechnolab\JAVASCRIPT\DAY 7>
```

7. What is the difference between function parameters and arguments?

Function Parameters:

- Variables listed within the parentheses of a function definition.

- To act as placeholders for the values that will be passed to the function.

Function Arguments:

- Values passed to a function when it is called.
- To provide values to the function parameters.

8. What is the purpose of the return statement in a function, and what does it do?

In JavaScript, the "return" statement is used to specify the value that a function should output or "return" back to the caller. When a return statement is encountered within a function, the function will immediately stop executing, and the specified value will be sent back to the code that called the function.

9. What are the differences between arrow functions and regular functions in JavaScript?

Arrow Function:

- Shorter and more concise syntax
- Does not have its own this binding
- Does not have its own arguments binding
- Cannot be used as constructors
- Inherits the lexical scope of the surrounding code

Regular Function:

- Longer and more verbose syntax
- Has its own this binding
- Has its own arguments binding
- Can be used as constructors
- Has its own lexical scope

10. What is the difference between local and global scope in JavaScript functions?

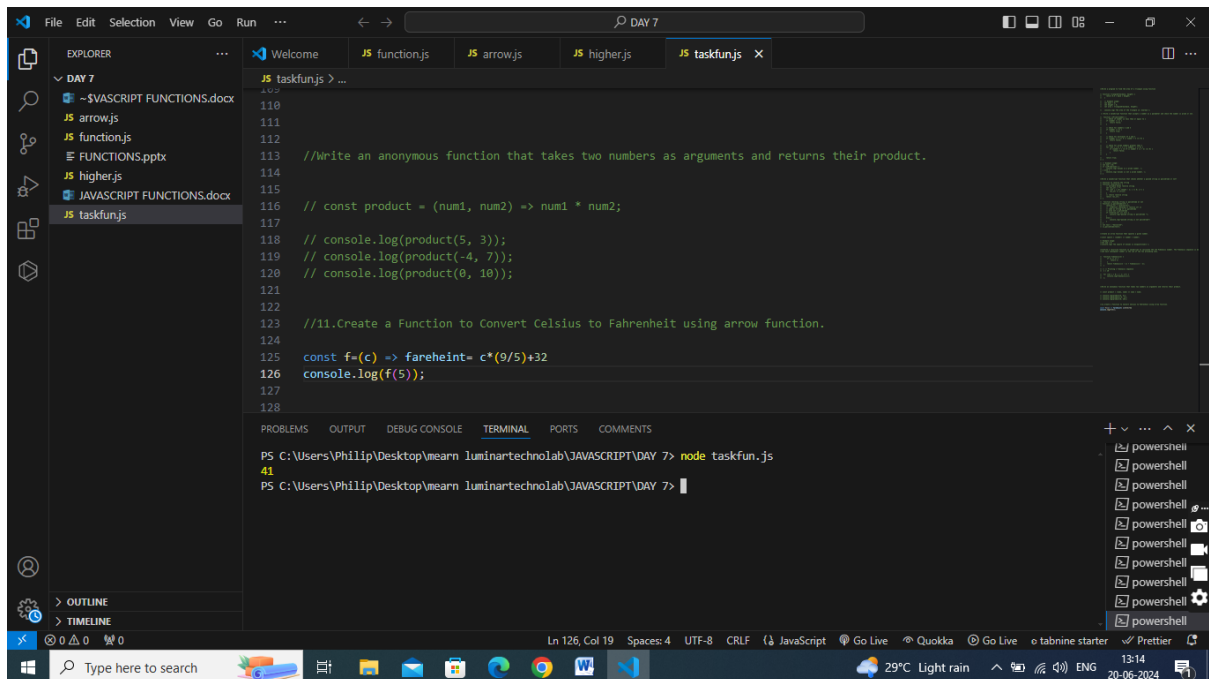
Global Scope:

Global scope refers to the variables, functions, and objects that are declared outside of any function and are accessible from anywhere in the program. Global variables are created when the program starts and are destroyed when the program ends.

Local Scope:

Local scope refers to the variables, functions, and objects that are declared inside of a function and are only accessible from within that function. Local variables are created when the function is called and are destroyed when the function returns.

11. Create a Function to Convert Celsius to Fahrenheit using arrow function.

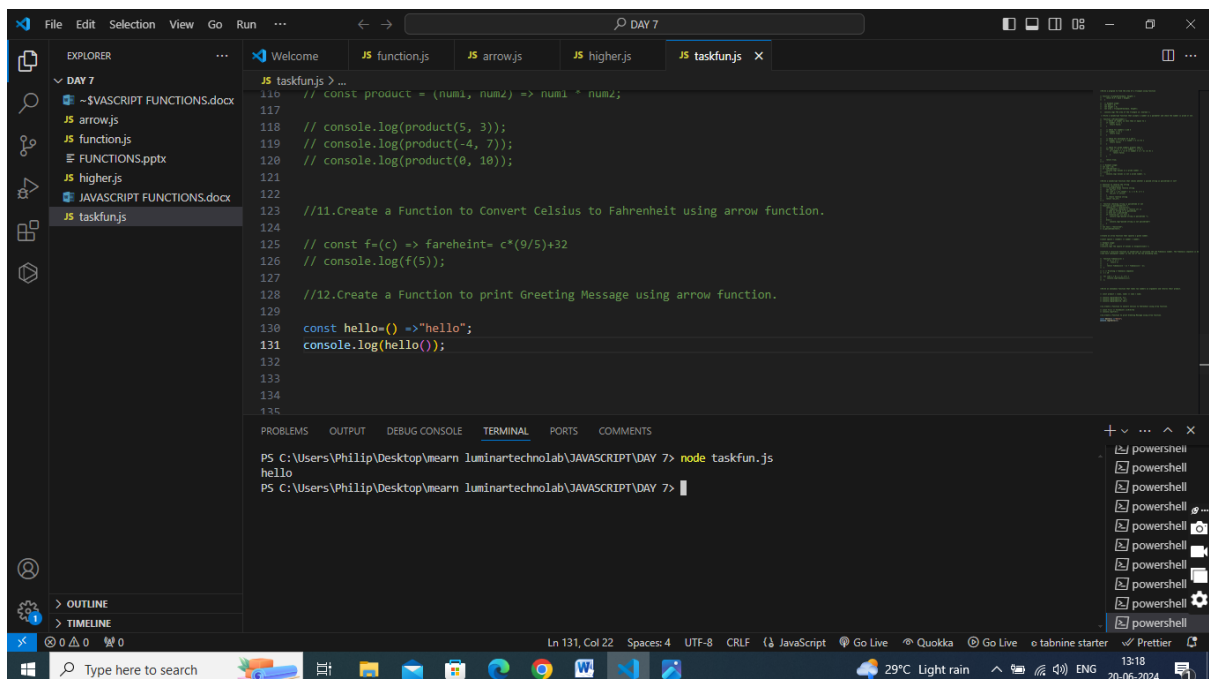


The screenshot shows the Visual Studio Code editor with the file explorer on the left displaying a project structure for 'DAY 7' including files like 'function.js', 'arrow.js', 'higher.js', and 'taskfun.js'. The main editor window is open to 'taskfun.js', which contains the following JavaScript code:

```
110
111
112
113 //Write an anonymous function that takes two numbers as arguments and returns their product.
114
115
116 // const product = (num1, num2) => num1 * num2;
117
118 // console.log(product(5, 3));
119 // console.log(product(-4, 7));
120 // console.log(product(0, 10));
121
122
123 //11.Create a Function to Convert Celsius to Fahrenheit using arrow function.
124
125 const f=(c) => fareheint= c*(9/5)+32
126 console.log(f(5));
127
128
```

The bottom panel shows the 'TERMINAL' tab with the command `node taskfun.js` executed, resulting in the output `41`.

12. Create a Function to print Greeting Message using arrow function.



The screenshot shows the Visual Studio Code editor with the file explorer on the left displaying a project structure for 'DAY 7' including files like 'function.js', 'arrow.js', 'higher.js', and 'taskfun.js'. The main editor window is open to 'taskfun.js', which contains the following JavaScript code:

```
110 // const product = (num1, num2) => num1 * num2;
111
112 // console.log(product(5, 3));
113 // console.log(product(-4, 7));
114 // console.log(product(0, 10));
115
116
117
118 //11.Create a Function to Convert Celsius to Fahrenheit using arrow function.
119
120 const f=(c) => fareheint= c*(9/5)+32
121 // console.log(f(5));
122
123
124 //12.Create a Function to print Greeting Message using arrow function.
125
126 const hello=() =>"hello";
127 console.log(hello());
128
129
130
131
132
133
134
135
```

The bottom panel shows the 'TERMINAL' tab with the command `node taskfun.js` executed, resulting in the output `hello`.