

Module 6) JAVASCRIPT BASIC & DOM (Basic logic Question)

Q.1 What is JavaScript? How to use it?

- Javascript is a scripting language. It was designed to add interactivity to HTML Pages.
- Javascript is an interpreted language that means the script executes without preliminary compilation.
- It's code directly embedded to the HTML Page. It works for all major browsers.
- Javascript is a client side, interpreted, Object Oriented, high level scripting language which cannot run stand alone.
- It was created By Netscape in 1995, evolved from Netscape's Live Script.
- Javascript works with HTML and CSS (Cascading Style Sheets).
- You can attach javascript file to html file using two methods:
Internal and external file.

- **Internal javascript**

- **It can be written in head tag or in body tag both ways.**

```
<html>
<head>
<title></title>
<script src="path">
– write your code here –
</script>
</head>
<body></body>
</html>
```

```
<html>
<head>
<title></title>
</head>
<body>
<script src="path">
– write your code here –
</script>
</body>
</html>
```

- **External file**

- **Make an external js file and attach it in html.**

```
<html>
<head>
<title></title>
<script src="script.js"></script>
```

```
</head>
<body></body>
</html>
```

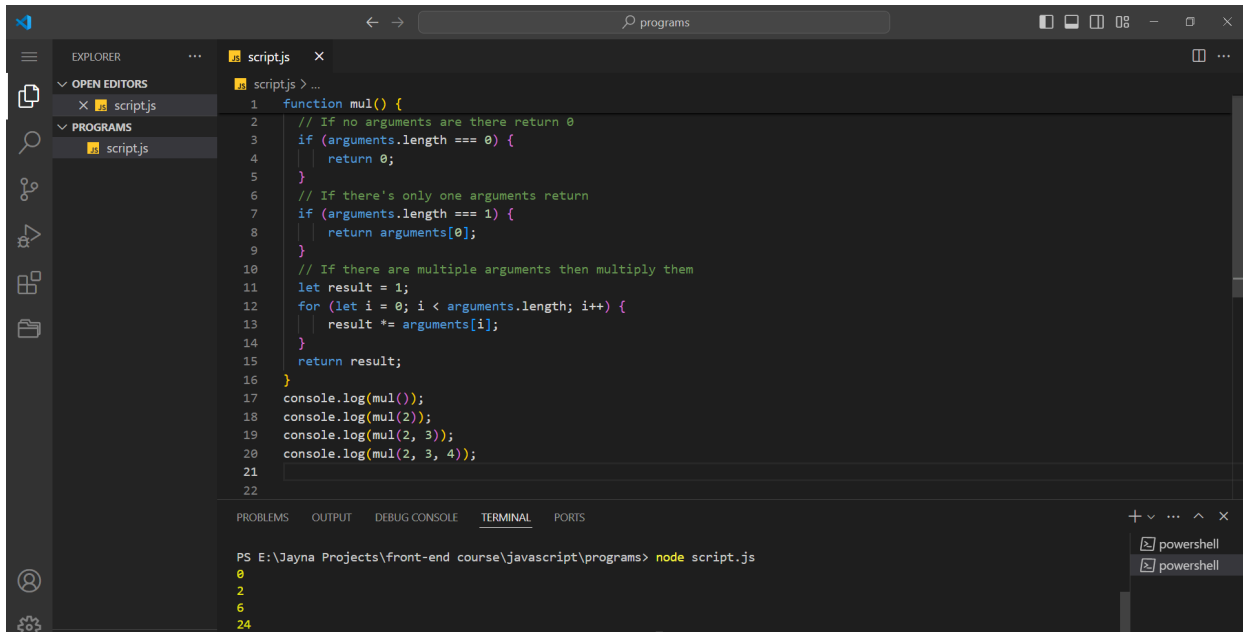
Q.2 How many types of Variables in JavaScript?

- Variables are containers for storing data.
- Javascript variables can be define in 4 ways:
Using **const** = use const if the value should not be changed.
Using **var** = Only use var if you MUST support old browsers.
Using **let** = Only use let if you can't use const.
- The var keyword was used in all JavaScript code from 1995 to 2015.
- The let and const keywords were added to JavaScript in 2015.
- The var keyword should only be used in code written for older browsers.

Q.3 Define a Data Types in js?

- In programming, data types are an important concept.
- To be able to operate on variables, it is important to know something about the type.
- JavaScript has 8 Data Types:
- **String,Number,Bigint,Boolean,Undefined,Null,Symbol,Object.**

Q.4 Write a mul Function Which will Work Properly When invoked With Following Syntax.



```
1 function mul() {
2   // If no arguments are there return 0
3   if (arguments.length === 0) {
4     return 0;
5   }
6   // If there's only one arguments return
7   if (arguments.length === 1) {
8     return arguments[0];
9   }
10  // If there are multiple arguments then multiply them
11  let result = 1;
12  for (let i = 0; i < arguments.length; i++) {
13    result *= arguments[i];
14  }
15  return result;
16 }
17 console.log(mul());
18 console.log(mul(2));
19 console.log(mul(2, 3));
20 console.log(mul(2, 3, 4));
21
22
```

Terminal Output:

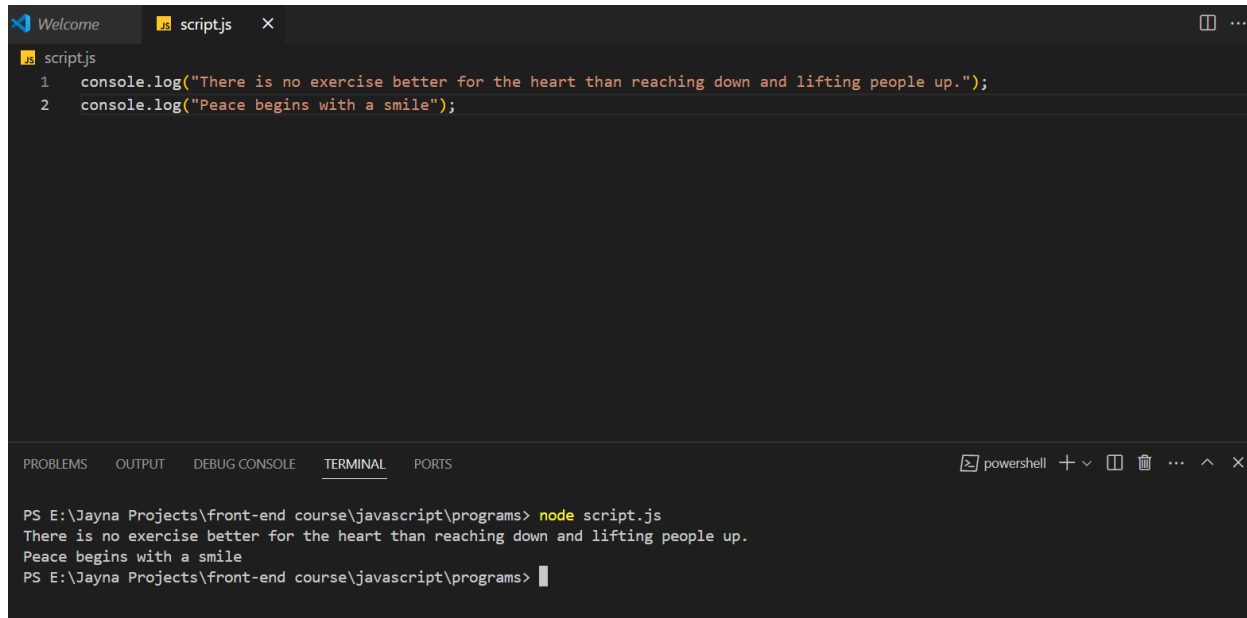
```
PS E:\Jayna Projects\front-end course\javascript\programs> node script.js
0
2
6
24
```

Q.5 What is the difference between undefined and undeclared in JavaScript?

- **Undefined:** It occurs when a variable has been declared but has not been assigned any value. Undefined is not a keyword.

- **Undeclared:** It occurs when we try to access any variable that is not initialized or declared earlier using the var or const keyword.

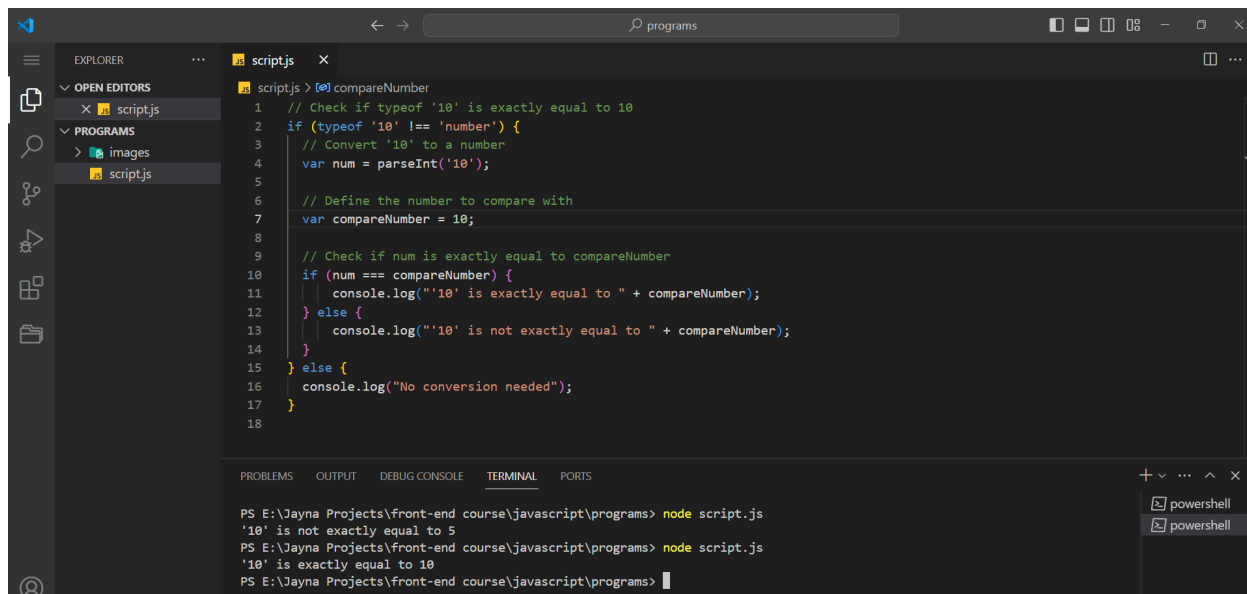
Q.6 Using console.log() print out the following statement: The quote 'There is no exercise better for the heart than reaching down and lifting people up.' by John Holmes teaches us to help one another. Using console.log() print out the following quote by Mother Teresa:



```
script.js
1 console.log("There is no exercise better for the heart than reaching down and lifting people up.");
2 console.log("Peace begins with a smile");
```

```
PS E:\Jayna Projects\front-end course\javascript\programs> node script.js
There is no exercise better for the heart than reaching down and lifting people up.
Peace begins with a smile
PS E:\Jayna Projects\front-end course\javascript\programs>
```

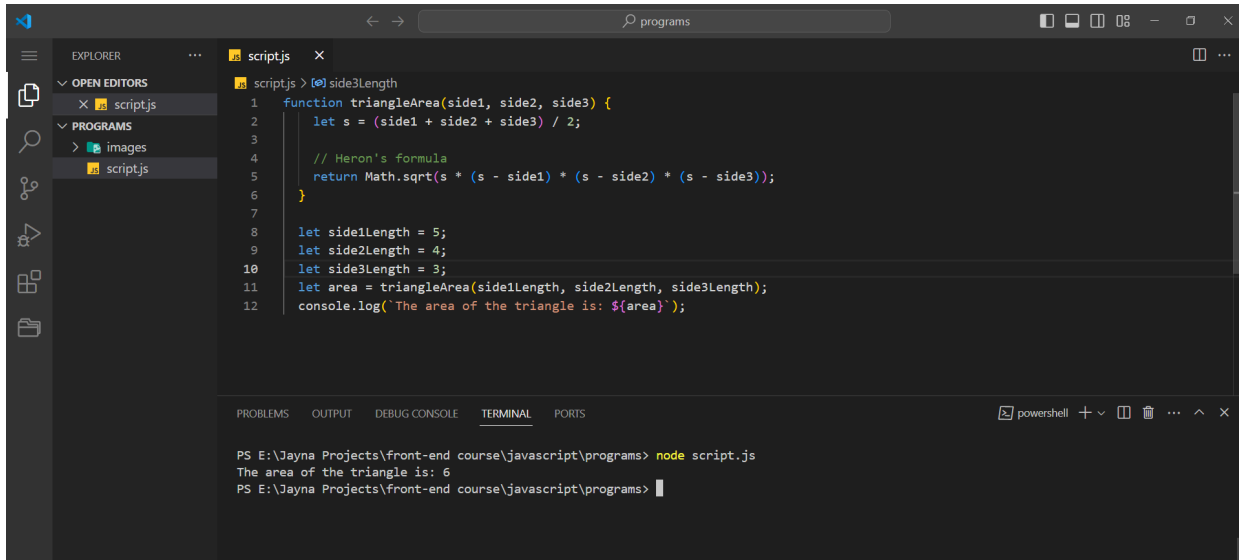
Q.7 Check if typeof '10' is exactly equal to 10. If not make it exactly equal?



```
script.js
1 // Check if typeof '10' is exactly equal to 10
2 if (typeof '10' !== 'number') {
3   // Convert '10' to a number
4   var num = parseInt('10');
5
6   // Define the number to compare with
7   var compareNumber = 10;
8
9   // Check if num is exactly equal to compareNumber
10  if (num === compareNumber) {
11    console.log("'10' is exactly equal to " + compareNumber);
12  } else {
13    console.log("'10' is not exactly equal to " + compareNumber);
14  }
15 } else {
16   console.log("No conversion needed");
17 }
18
```

```
PS E:\Jayna Projects\front-end course\javascript\programs> node script.js
'10' is not exactly equal to 5
PS E:\Jayna Projects\front-end course\javascript\programs> node script.js
'10' is exactly equal to 10
PS E:\Jayna Projects\front-end course\javascript\programs>
```

Q.8 Write a JavaScript Program to find the area of a triangle?

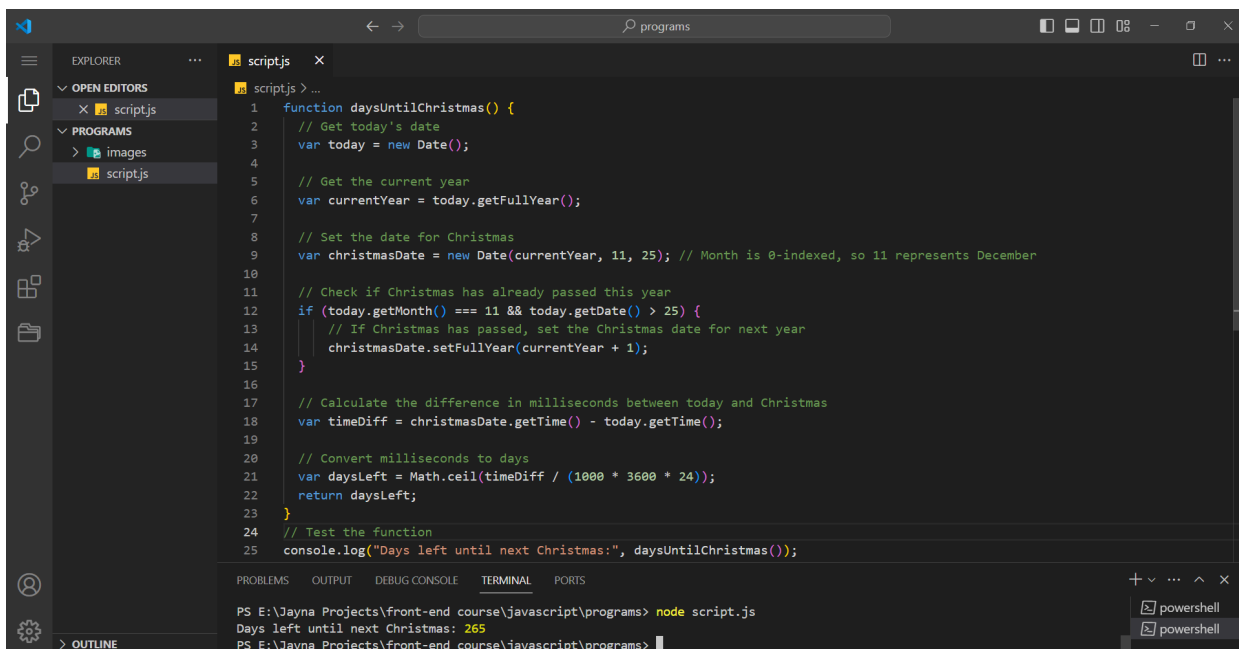


The screenshot shows a Visual Studio Code editor with a file named `script.js` open. The code defines a function `triangleArea` that takes three side lengths as arguments, calculates the semi-perimeter `s`, and then uses Heron's formula to find the area. The sides are set to 5, 4, and 3, and the result is logged to the console. The terminal at the bottom shows the command `node script.js` being executed, resulting in the output: "The area of the triangle is: 6".

```
1 function triangleArea(side1, side2, side3) {  
2   let s = (side1 + side2 + side3) / 2;  
3  
4   // Heron's formula  
5   return Math.sqrt(s * (s - side1) * (s - side2) * (s - side3));  
6 }  
7  
8 let side1Length = 5;  
9 let side2Length = 4;  
10 let side3Length = 3;  
11 let area = triangleArea(side1Length, side2Length, side3Length);  
12 console.log('The area of the triangle is: ${area}');
```

```
PS E:\Jayna Projects\front-end course\javascript\programs> node script.js  
The area of the triangle is: 6  
PS E:\Jayna Projects\front-end course\javascript\programs>
```

Q.9 Write a JavaScript program to calculate days left until next Christmas?



The screenshot shows a Visual Studio Code editor with a file named `script.js` open. The code defines a function `daysUntilChristmas` that calculates the number of days left until the next Christmas. It gets the current date and year, sets the Christmas date for the current year (December 25th), and if it's already past, it sets it for the next year. It then calculates the time difference in milliseconds and converts it to days. The result is logged to the console. The terminal at the bottom shows the command `node script.js` being executed, resulting in the output: "Days left until next Christmas: 265".

```
1 function daysUntilChristmas() {  
2   // Get today's date  
3   var today = new Date();  
4  
5   // Get the current year  
6   var currentYear = today.getFullYear();  
7  
8   // Set the date for Christmas  
9   var christmasDate = new Date(currentYear, 11, 25); // Month is 0-indexed, so 11 represents December  
10  
11   // Check if Christmas has already passed this year  
12   if (today.getMonth() === 11 && today.getDate() > 25) {  
13     // If Christmas has passed, set the Christmas date for next year  
14     christmasDate.setFullYear(currentYear + 1);  
15   }  
16  
17   // Calculate the difference in milliseconds between today and Christmas  
18   var timeDiff = christmasDate.getTime() - today.getTime();  
19  
20   // Convert milliseconds to days  
21   var daysLeft = Math.ceil(timeDiff / (1000 * 3600 * 24));  
22   return daysLeft;  
23 }  
24 // Test the function  
25 console.log("Days left until next Christmas:", daysUntilChristmas());
```

```
PS E:\Jayna Projects\front-end course\javascript\programs> node script.js  
Days left until next Christmas: 265  
PS E:\Jayna Projects\front-end course\javascript\programs>
```

Q.10 What is a Condition Statement?

- Conditional statement is very often used when you write code and you want to perform different actions for different decisions.
- In JavaScript these are the following conditional statements:
 - Use **if** to specify a block of code to be executed, if a specified condition is true

- Use **else** to specify a block of code to be executed, if the same condition is false
- Use **else if** to specify a new condition to test, if the first condition is false
- Use **switch** to specify many alternative blocks of code to be executed.

Q.11 Find circumference of Rectangle formula : $C = 4 * a$?

```

1 //function to calculate circumference of rectangle
2 function calculateRectangleCircumference(length, width) {
3     return 2 * (length + width);
4 }
5
6 const length = 10;
7 const width = 5;
8 const circumference = calculateRectangleCircumference(length, width);
9 console.log("Circumference of the rectangle:", circumference);
10

```

PS E:\Jayna Projects\front-end course\javascript\programs> node script.js
Circumference of the rectangle: 30
PS E:\Jayna Projects\front-end course\javascript\programs>

Q.12 WAP to convert years into days and days into years?

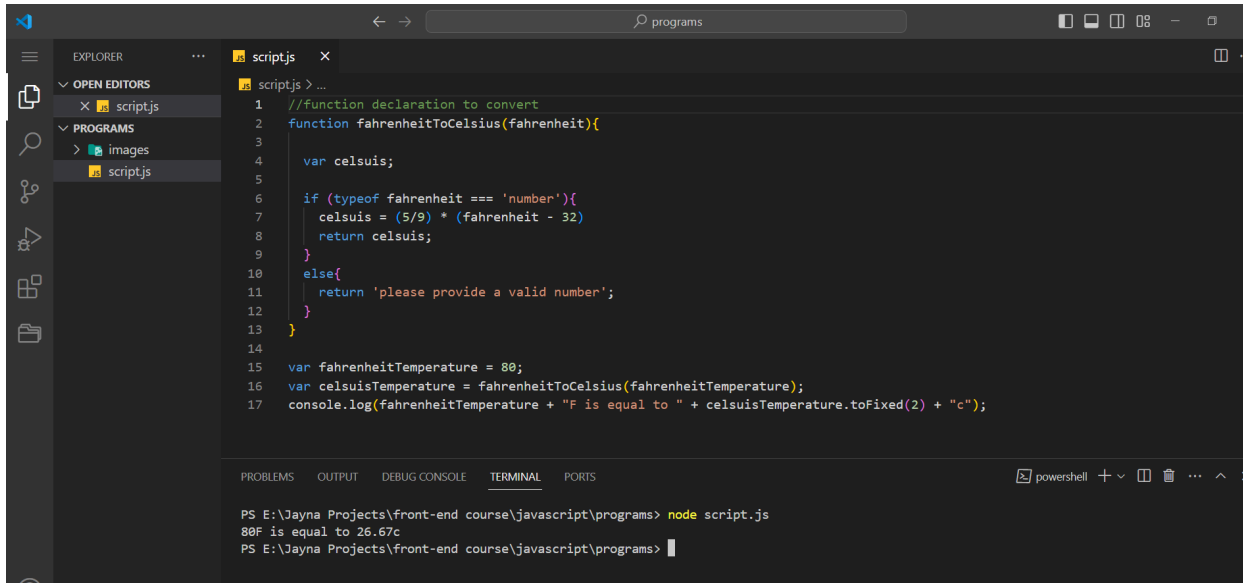
```

1 // Function to convert years to days
2 function yearsToDays(years) {
3     return years * 365.25;
4 }
5
6 // Function to convert days to years
7 function daysToYears(days) {
8     return days / 365.25;
9 }
10
11 var years = 6;
12 var days = yearsToDays(years);
13 console.log(years + " years is equal to approximately " + days + " days.");
14
15 var daysInput = 450;
16 var yearsOutput = daysToYears(daysInput);
17 console.log(daysInput + " days is equal to approximately " + yearsOutput.toFixed(2) + " years.");
18
19

```

PS E:\Jayna Projects\front-end course\javascript\programs> node script.js
6 years is equal to approximately 2191.5 days.
450 days is equal to approximately 1.23 years.
PS E:\Jayna Projects\front-end course\javascript\programs>

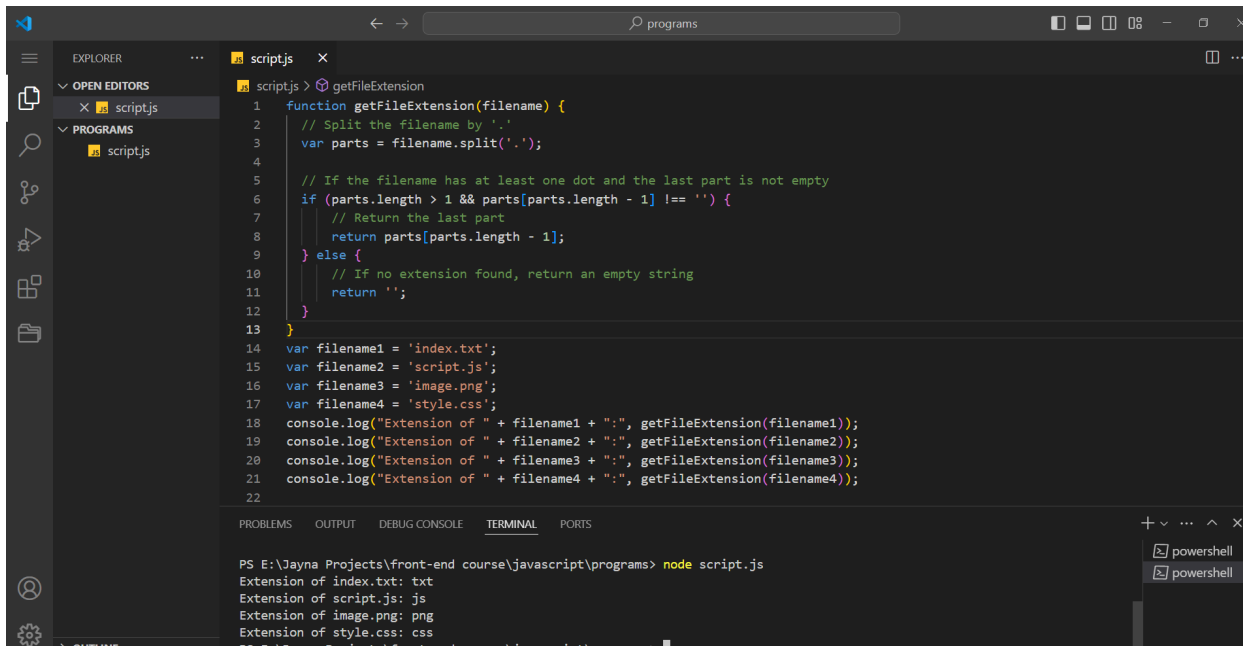
Q.13 Convert temperature Fahrenheit to Celsius? (Conditional logic Question)



```
1 //function declaration to convert
2 function fahrenheitToCelsius(fahrenheit){
3
4     var celsius;
5
6     if (typeof fahrenheit === 'number'){
7         celsius = (5/9) * (fahrenheit - 32)
8         return celsius;
9     }
10    else{
11        return 'please provide a valid number';
12    }
13 }
14
15 var fahrenheitTemperature = 80;
16 var celsiusTemperature = fahrenheitToCelsius(fahrenheitTemperature);
17 console.log(fahrenheitTemperature + "F is equal to " + celsiusTemperature.toFixed(2) + "C");
```

PS E:\Jayna Projects\front-end course\javascript\programs> node script.js
80F is equal to 26.67C
PS E:\Jayna Projects\front-end course\javascript\programs>

Q.14 Write a JavaScript exercise to get the extension of a filename.?



```
1 function getFileExtension(filename) {
2     // Split the filename by '.'
3     var parts = filename.split('.');
4
5     // If the filename has at least one dot and the last part is not empty
6     if (parts.length > 1 && parts[parts.length - 1] !== '') {
7         // Return the last part
8         return parts[parts.length - 1];
9     } else {
10        // If no extension found, return an empty string
11        return '';
12    }
13 }
14 var filename1 = 'index.txt';
15 var filename2 = 'script.js';
16 var filename3 = 'image.png';
17 var filename4 = 'style.css';
18 console.log("Extension of " + filename1 + ":", getFileExtension(filename1));
19 console.log("Extension of " + filename2 + ":", getFileExtension(filename2));
20 console.log("Extension of " + filename3 + ":", getFileExtension(filename3));
21 console.log("Extension of " + filename4 + ":", getFileExtension(filename4));
22
```

PS E:\Jayna Projects\front-end course\javascript\programs> node script.js
Extension of index.txt: txt
Extension of script.js: js
Extension of image.png: png
Extension of style.css: css
PS E:\Jayna Projects\front-end course\javascript\programs>

Q.15 What is the result of the expression (5 > 3 && 2 < 4)?

- The expression (5>3 && 2<4) is **true**. Because the condition 5>3 is true and 2<4 is true.
- So, the AND operator returns true because both the conditions are true.

Q.16 What is the result of the expression (true && 1 && "hello")?

- The result of the expression is **"hello"**.

- Javascript evaluates && expression from left to right. If all operands are truthy, it returns the value of the last operand.
- So in this case:
 - true is truthy.
 - 1 is truthy.
 - "hello" is also truthy.

Q.17 What is the result of the expression `true && false || false && true`?

- The result of the expression is **false**.
- Javascript evaluates logical AND (&&) expression before logical OR (||) expression.
- Evaluation step by step:
 - True && false: This evaluates false because one of the operands is false.
 - False && true: This evaluates false for the same reason.
 - False || false: This evaluates false because both operands are false.
- So, the final result is false.

Q.18 What is a Loop and Switch Case in JavaScript define that ?

- In JavaScript, a loop is a programming construct that allows you to execute a block of code repeatedly until a certain condition is met.
- There are several types of loops in JavaScript, including the for loop, while loop, and do-while loop.
- **for loop:** The for loop is used when you know the number of times you want to execute a block of code. It consists of three parts: initialization, condition, and iteration. The loop continues to execute as long as the condition evaluates to true.

Syntax:

```
for (initialization; condition; iteration) {
  // Code to be executed
}
```

- **while loop:** The while loop is used when you want to execute a block of code as long as a condition is true. The loop evaluates the condition before executing the block of code.

Syntax:

```
while (condition) {
  // Code to be executed
}
```

- **do-while loop:** The do-while loop is similar to the while loop, but it always executes the block of code at least once before checking the condition. The loop continues to execute as long as the condition evaluates to true.

Syntax:

```
do {
  // Code to be executed
} while (condition);
```

- **Switch case** is another control flow statement in JavaScript used to perform different actions based on different conditions. It provides a cleaner and more organized alternative to using multiple if...else statements.

Syntax:

```
switch (expression) {
  case value1:
    value1
    break;
  case value2:
    value2
    break;
  default:
  }
```

Q.19 What is the use of is Nan function?

- In javascript NAN stands for "Not a number".
- It is a special value that represents the result of an operation that cannot produce a meaningful numeric result.
- The NAN value is returned when:
 - Performing arithmetic operations involving non-numeric values or invalid calculations.
 - Converting non-numeric strings to numbers using parseInt() or parseFloat() functions when the string cannot be parsed as a valid number.

Q.20 What is the difference between && and || in JavaScript?

- In javascript && and || are both logical operators used for boolean operations, but they behave differently:

&&(Logical AND):

- The && operator returns true if both operands are true, otherwise it returns false.
- It short-circuits: if the left operand is false, the right operand is not evaluated because the result will always be false.
- Example: true && true returns true, true && false returns false, false && true returns false, false && false returns false.

|| (Logical OR):

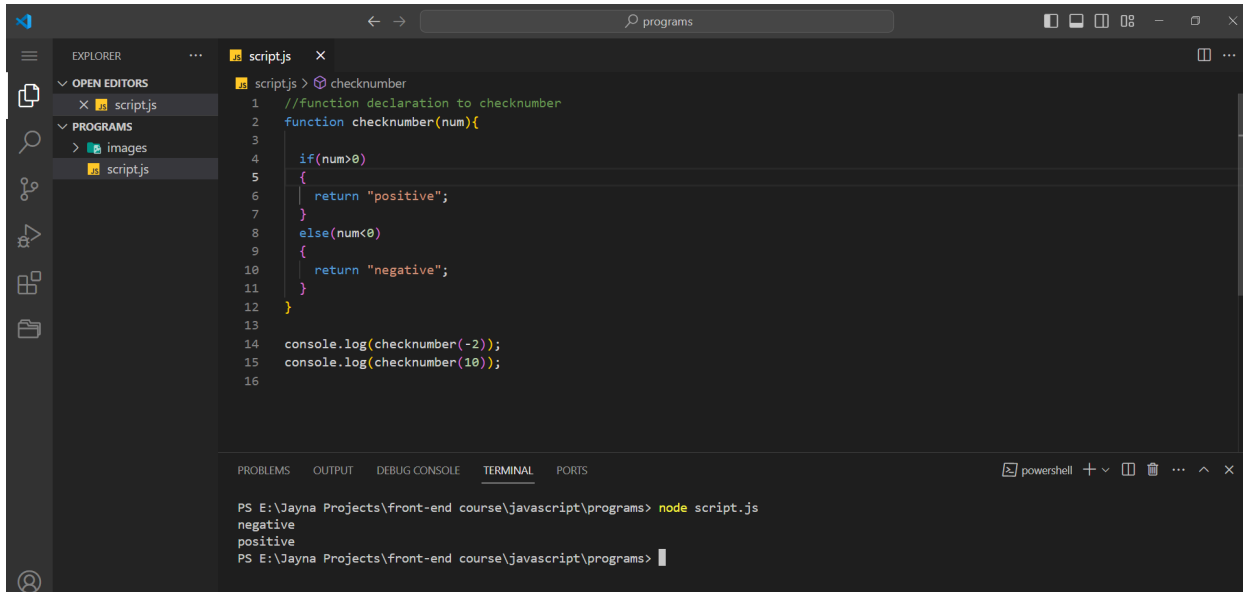
- The || operator returns true if at least one of the operands is true, otherwise it returns false.
- It short-circuits: if the left operand is true, the right operand is not evaluated because the result will always be true.
- Example: true || true returns true, true || false returns true, false || true returns true, false || false returns false.

Q.21 What is the use of Void (0)?

- In javascript void(0) is an expression that evaluates to undefined.
- It's often used to return undefined from an expression context.

- The primary use case of `void(0)` is to create an expression that returns undefined without explicitly using the undefined keyword.
- It can be useful in scenarios where you want to ensure that a function or event handler doesn't inadvertently return a value.

Q.22 Check Number Is Positive or Negative in JavaScript?



The screenshot shows a Visual Studio Code editor with a file named `checknumber.js` open. The code defines a function `checknumber(num)` that checks if a number is positive or negative. It uses `if` and `else` statements to return "positive" or "negative". The function is then called with `-2` and `10` using `console.log`.

```

1 //function declaration to checknumber
2 function checknumber(num){
3
4     if(num>0)
5     {
6         return "positive";
7     }
8     else(num<0)
9     {
10        return "negative";
11    }
12 }
13
14 console.log(checknumber(-2));
15 console.log(checknumber(10));
16

```

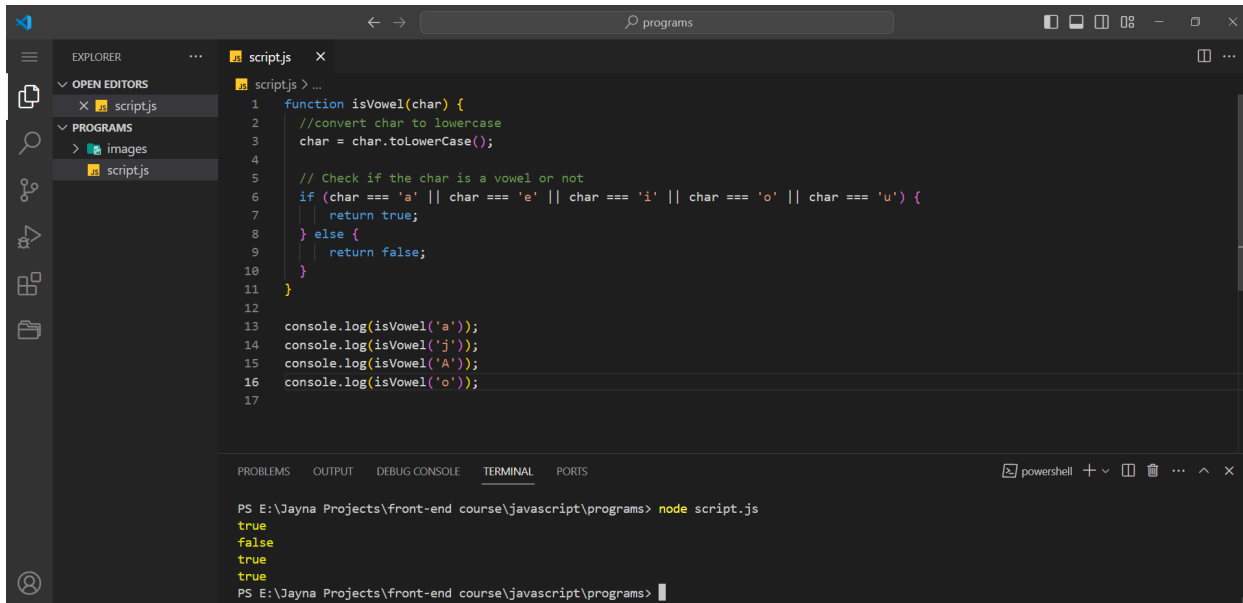
The terminal at the bottom shows the command `node script.js` being executed, resulting in the output:

```

PS E:\Jayna Projects\front-end course\javascript\programs> node script.js
negative
positive
PS E:\Jayna Projects\front-end course\javascript\programs>

```

Q.23 Find the Character Is Vowel or Not ?



The screenshot shows a Visual Studio Code editor with a file named `isVowel.js` open. The code defines a function `isVowel(char)` that checks if a character is a vowel. It converts the character to lowercase and then checks if it is 'a', 'e', 'i', 'o', or 'u'. The function returns `true` if it is a vowel and `false` otherwise. The function is then called with 'a', 'j', 'A', and 'o' using `console.log`.

```

1 function isVowel(char) {
2     //convert char to lowercase
3     char = char.toLowerCase();
4
5     // Check if the char is a vowel or not
6     if (char === 'a' || char === 'e' || char === 'i' || char === 'o' || char === 'u') {
7         return true;
8     } else {
9         return false;
10    }
11 }
12
13 console.log(isVowel('a'));
14 console.log(isVowel('j'));
15 console.log(isVowel('A'));
16 console.log(isVowel('o'));
17

```

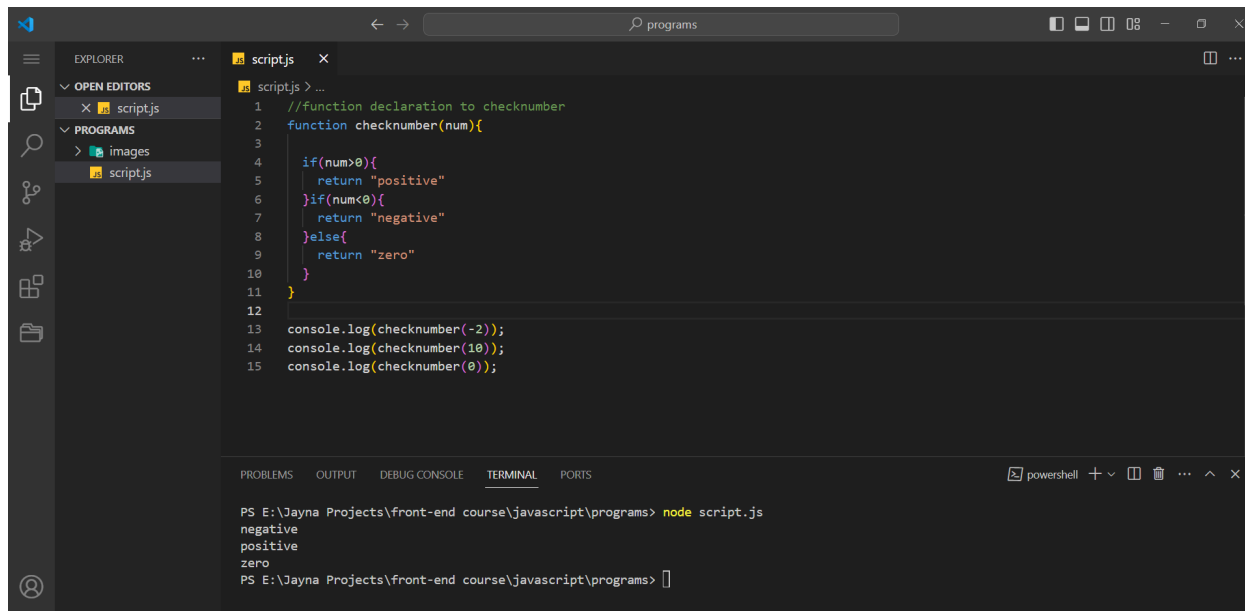
The terminal at the bottom shows the command `node script.js` being executed, resulting in the output:

```

PS E:\Jayna Projects\front-end course\javascript\programs> node script.js
true
false
true
true
PS E:\Jayna Projects\front-end course\javascript\programs>

```

Q.24 Write to check whether a number is negative, positive or zero?

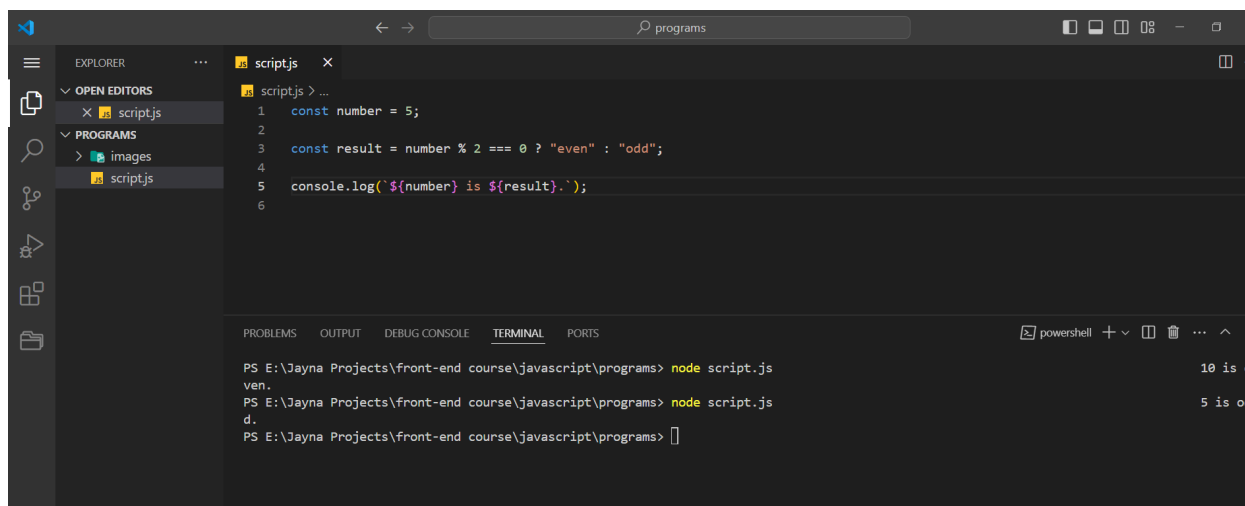


```
1 //function declaration to checknumber
2 function checknumber(num){
3
4     if(num>0){
5         return "positive"
6     }if(num<0){
7         return "negative"
8     }else{
9         return "zero"
10    }
11 }
12
13 console.log(checknumber(-2));
14 console.log(checknumber(10));
15 console.log(checknumber(0));
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS E:\Jayna Projects\front-end course\javascript\programs> node script.js
negative
positive
zero
PS E:\Jayna Projects\front-end course\javascript\programs>
```

Q.25 Write to find number is even or odd using ternary operator in JS?

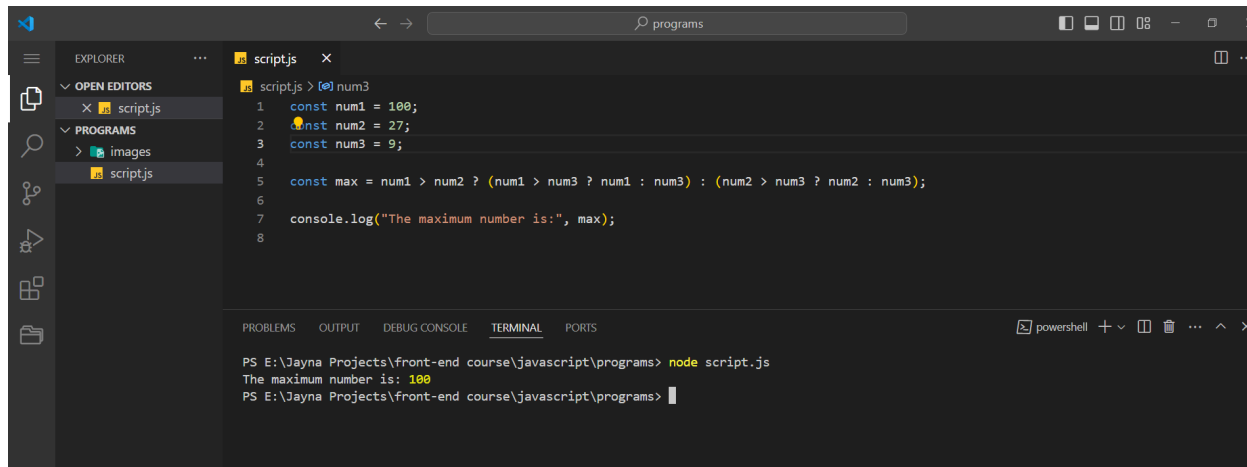


```
1 const number = 5;
2
3 const result = number % 2 === 0 ? "even" : "odd";
4
5 console.log(`${number} is ${result}.`);
6
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS E:\Jayna Projects\front-end course\javascript\programs> node script.js
10 is even.
PS E:\Jayna Projects\front-end course\javascript\programs> node script.js
5 is odd.
PS E:\Jayna Projects\front-end course\javascript\programs>
```

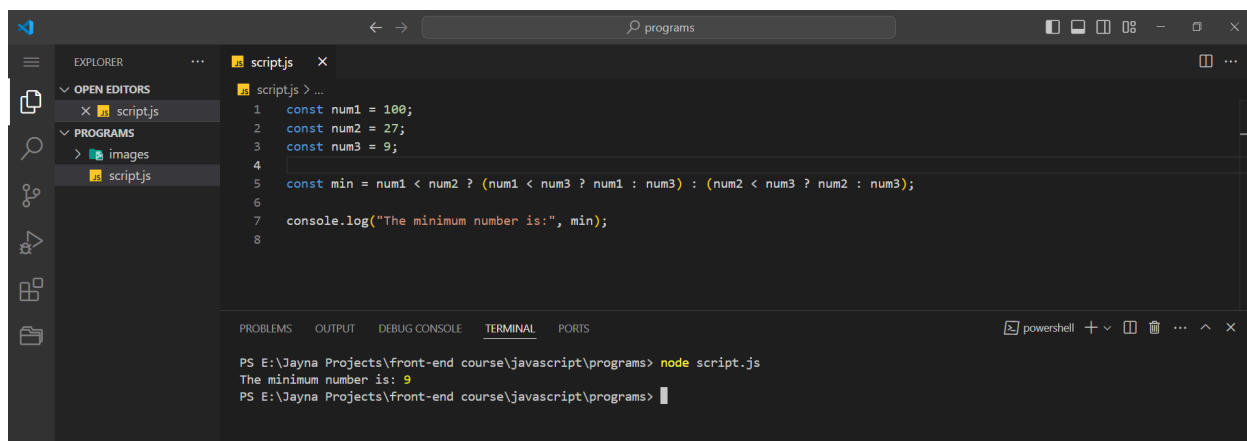
Q.26 Write find maximum number among 3 numbers using ternary operator in JS?



```
scriptjs > num3
1  const num1 = 100;
2  const num2 = 27;
3  const num3 = 9;
4
5  const max = num1 > num2 ? (num1 > num3 ? num1 : num3) : (num2 > num3 ? num2 : num3);
6
7  console.log("The maximum number is:", max);
8
```

```
PS E:\Jayna Projects\front-end course\javascript\programs> node script.js
The maximum number is: 100
PS E:\Jayna Projects\front-end course\javascript\programs>
```

Q.27 Write to find minimum number among 3 numbers using ternary operator in JS?



```
scriptjs > ...
1  const num1 = 100;
2  const num2 = 27;
3  const num3 = 9;
4
5  const min = num1 < num2 ? (num1 < num3 ? num1 : num3) : (num2 < num3 ? num2 : num3);
6
7  console.log("The minimum number is:", min);
8
```

```
PS E:\Jayna Projects\front-end course\javascript\programs> node script.js
The minimum number is: 9
PS E:\Jayna Projects\front-end course\javascript\programs>
```

Q.28 Write to find the largest of three numbers in JS?

The screenshot shows the Visual Studio Code editor with a file named `script.js` open. The code defines a function `findLargest` that takes three arguments (`num1`, `num2`, `num3`) and returns the largest value. It uses `let` for a local variable `largest` and `const` for the input numbers. The terminal at the bottom shows the command `node script.js` being executed, resulting in the output: `The largest number is: 201`.

```
1 function findLargest(num1, num2, num3) {
2   let largest = num1;
3
4   if (num2 > largest) {
5     largest = num2;
6   }
7
8   if (num3 > largest) {
9     largest = num3;
10  }
11
12  return largest;
13 }
14
15 const num1 = 101;
16 const num2 = 201;
17 const num3 = 150;
18 const largestNumber = findLargest(num1, num2, num3);
19 console.log("The largest number is:", largestNumber);
20
```

Terminal Output:

```
PS E:\Jayna Projects\front-end course\javascript\programs> node script.js
The largest number is: 201
PS E:\Jayna Projects\front-end course\javascript\programs>
```

Q.29 Write a program to show :

i. Monday to Sunday using switch case in JS?

The screenshot shows the Visual Studio Code editor with a file named `script.js` open. The code defines a function `getDayName` that takes a `dayNumber` as an argument and returns the corresponding day name using a `switch` statement. The cases are numbered 1 through 7 for Monday through Sunday, and a default case for invalid numbers. The terminal at the bottom shows the command `node script.js` being executed, resulting in the output: `Monday`.

```
1 function getDayName(dayNumber) {
2   let dayName;
3
4   switch(dayNumber) {
5     case 1:
6       dayName = "Monday";
7       break;
8     case 2:
9       dayName = "Tuesday";
10      break;
11     case 3:
12       dayName = "Wednesday";
13       break;
14     case 4:
15       dayName = "Thursday";
16       break;
17     case 5:
18       dayName = "Friday";
19       break;
20     case 6:
21       dayName = "Saturday";
22       break;
23     case 7:
24       dayName = "Sunday";
25       break;
26     default:
27       dayName = "Invalid day number";
28   }
29   return dayName;
30 }
```

Terminal Output:

```
PS E:\Jayna Projects\front-end course\javascript\programs> node script.js
Monday
PS E:\Jayna Projects\front-end course\javascript\programs>
```

The screenshot shows the Visual Studio Code interface. The Explorer sidebar on the left shows the file structure with 'scriptjs' under 'PROGRAMS'. The main editor window displays a JavaScript file named 'scriptjs' with the following code:

```
31  
32 for (let i = 1; i <= 7; i++) {  
33   console.log(`Day ${i}: ${getDayName(i)}`);  
34 }  
35
```

Below the editor, the TERMINAL panel is active, showing the output of running 'node script.js' in a PowerShell window:

```
PS E:\Jayna Projects\front-end course\javascript\programs> node script.js  
Day 1: Monday  
Day 2: Tuesday  
Day 3: Wednesday  
Day 4: Thursday  
Day 5: Friday  
Day 6: Saturday  
Day 7: Sunday  
PS E:\Jayna Projects\front-end course\javascript\programs>
```

ii. Vowel or Consonant using switch case in JS?

The screenshot shows the Visual Studio Code interface. The Explorer sidebar on the left shows the file structure with 'scriptjs' under 'PROGRAMS'. The main editor window displays a JavaScript file named 'scriptjs' with the following code:

```
1 function checkVowelOrConsonant(character) {  
2   switch(character.toLowerCase()) {  
3     case 'a':  
4     case 'e':  
5     case 'i':  
6     case 'o':  
7     case 'u':  
8       return "Vowel";  
9     default:  
10      return "Consonant";  
11   }  
12 }  
13  
14 const testCharacter = 'j';  
15 console.log(`${testCharacter} is a ${checkVowelOrConsonant(testCharacter)}.`);  
16
```

Below the editor, the TERMINAL panel is active, showing the output of running 'node script.js' in a PowerShell window:

```
PS E:\Jayna Projects\front-end course\javascript\programs> node script.js  
o is a Vowel.  
PS E:\Jayna Projects\front-end course\javascript\programs> node script.js  
A is a Vowel.  
PS E:\Jayna Projects\front-end course\javascript\programs> node script.js  
j is a Consonant.  
PS E:\Jayna Projects\front-end course\javascript\programs>
```