

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

Ans. JavaScript is a text-based programming language used both on the client-side and server-side that it helps to make web pages interactive.

### **Q.2 How many types of Variable in JavaScript?**

Ans. A JavaScript variable define storage location. There are two types of variables in JavaScript:

local variable For example, let const

global variable. For example, var

### **Q.3 Define a Data Types in js?**

Ans. Data types in JavaScript define the type of data that a variable can store.

JavaScript includes primitive and non-primitive data types.

The primitive data types in JavaScript include string, number, boolean, undefined, null, and symbol.

The non-primitive data type includes the object.

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

Ans. The mul () method is a useful tool for working with CSS values. It allows you to easily multiply CSS values by numbers or other CSS values.

### **Q.5 What the deference between undefined and undeclare in JavaScript?**

**Ans.** 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:**

**Ans.** `console.log("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.");`

### **Q.7 Check if type of '10' is exactly equal to 10. If not make it exactly equal?**

**Ans.** To check if `typeof '10'` is exactly equal to 10, we can use the following code in JavaScript:

```
const x = '10';  
console.log(typeof x === 10); // false
```

This will return false, because the type of '10' is a string, and the type of 10 is a number. To make the type of '10' exactly equal to 10, you can use the following code:

```
const x = parseInt('10');  
console.log(typeof x === 10); // true
```

This will parse the string '10' into a number, and then assign it to the variable x. The type of x will now be a number, and it will be exactly equal to 10.

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

**Ans.**

```
function findArea(base, height) {  
    let area = base * height / 2;  
    return area;  
}  
  
let base = prompt("Enter the base of the triangle:");  
let height = prompt("Enter the height of the triangle:");  
let area = findArea(base, height);  
console.log("The area of the triangle is:", area);
```

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

**Ans.** Here is a JavaScript program to calculate days left until next Christmas:

```
const today = new Date();  
const christmas = new Date("December 25, 2024");  
const msPerDay = 24 * 60 * 60 * 1000;  
const timeLeft = (christmas.getTime() - today.getTime())  
/ milisecPerDay;  
console.log(Math.ceil(timeLeft));
```

### **Q.10 What is Condition Statement?**

**Ans.** A conditional statement is a statement that connects two ideas with the words "if" and "else".

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

**Ans.**

```
function circumferenceOfRectangle(a) {  
    return 4 * a;  
}  
console.log(circumferenceOfRectangle(5)); // 20
```

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

**Ans.** a JavaScript program to convert years into days and days into years:

```
function convertYearsToDays(years) {  
    return years * 365;  
}  
function convertDaysToYears(days) {  
    return days / 365;  
}  
const years = 10;  
const days = 3650;  
console.log(convertYearsToDays(years)); // 3650  
console.log(convertDaysToYears(days)); // 10
```

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

**Ans.** a simple JavaScript program to convert temperature from Fahrenheit to Celsius:

```
function convertFahrenheitToCelsius(fahrenheit) {  
    var celsius = (fahrenheit - 32) * 5 / 9;  
    return celsius;  
}  
  
var fahrenheit = prompt("Enter a temperature in Fahrenheit: ");  
var celsius = convertFahrenheitToCelsius(fahrenheit);  
alert("The temperature in Celsius is " + celsius);
```

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

**Ans.**

```
function getExtension(filename) {  
    const parts = filename.split(".");  
    return parts[parts.length - 1];  
}  
  
const filename = "myfile.txt";  
const extension = getExtension(filename)  
console.log(extension);
```

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

**Ans.** The result of the expression ((5>3)&&(2<4))

The logical AND operator && returns (true) if both operands are (true), and (false) otherwise

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

**Ans.** The result of the expression (true && 1 && "hello") is "hello".

The && operator is the logical AND operator. It returns true if both of its operands are true, and false otherwise.

In this case, both true and 1 are truthy values, so the expression evaluates to true. The && operator then returns the last truthy operand, which is "hello".

Therefore, the result of the expression is "hello".

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

**Ans.**        true && false || false && true // ans is false;

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

**Ans.**        Loops: are programming constructs that allow us to execute a block of code repeatedly. JavaScript has three types of loops: for, while, and do...while.

Switch case: is a programming construct that allows us to execute different blocks of code based on the value of a variable

### **Q.19 What is the use of is NaN function?**

**Ans.**        We can use the isNaN() function to validate input from users or to check the results of operations. For example, you could use it to make sure that a user enters a valid number before submitting a form.

We can also use the isNaN() function to check the results of operations. For example, you could use it to make sure that the result of a division operation is not NaN before using it in another calculation.

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

**Ans.**        If the expression on the left of && is falsy, it will immediately return false without checking the expression on the right. If the expression on the left of || is truthy, it will immediately return true without checking the expression on the right.

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

**Ans.**        When a link is clicked, we don't want the browser to load a new page or refresh the same page . Instead it will just perform the JavaScript attached to that link.

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

```
const num = -5;
```

```
if (num > 0) {  
    console.log("The number is positive");  
}  
else (num < 0) {  
    console.log("The number is negative");  
}
```

### **Q.23 Find the Character Is Vowel or Not ?**

**Ans.**

```
function isVowel(character) {  
    // Create an array of vowels.  
    const vowels = ["a", "e", "i", "o", "u"];  
    // Check if the character is in the array of vowels.  
    return vowels.includes(character);  
}  
  
const input = "A";  
const isVowel = isVowel(input);  
if (isVowel) {  
    console.log("The character is a vowel.");  
} else {  
    console.log("The character is not a vowel.");  
}
```

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

**Ans.** const num = -5;

```
if (num > 0) {  
    console.log("The number is positive");  
} else if (num < 0) {  
    console.log("The number is negative");  
} else {  
    console.log("The number is zero");  
}
```

```
}
```

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

**Ans.**

```
const number = 11;  
const isEven = number % 2 === 0 ? "Even" : "Odd";  
console.log(isEven); // "Odd"
```

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

**Ans.** function findMax(a, b, c) {

```
  // Ternary operator
```

```
  return (a > b) ? (a > c ? a : c) : (b > c ? b : c);
```

```
}
```

```
const max = findMax(10, 20, 30);
```

```
console.log(max); // 30
```

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

**Ans.** function findMin(a, b, c) {

```
  // Use the ternary operator to compare a and b and store the smaller  
  number in temp.
```

```
  let temp = a < b ? a : b;
```

```
  // Use the ternary operator to compare temp and c and return the  
  smaller number.
```

```
  return temp < c ? temp : c;
```

```
}
```

```
// Example usage:
```

```
const min = findMin(1, 2, 3);
```

```
console.log(min); // 1
```

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

**Ans.** let largestNum;

```
if (num1 > num2 && num1 > num3) {  
    largestNum = num1;  
}  
else if (num2 > num1 && num2 > num3) {  
    largestNum = num2;  
}  
else {  
    largestNum = num3;  
}  
console.log(largestNum);
```

### **Q.29 Write to show**

**i. Monday to Sunday using switch case in JS?**

**ii. Vowel or Consonant using switch case in JS?**

**(Conditional looping logic Question)**

**Ans.**

// Show Monday to Sunday using switch case in JS

```
const day = new Date().getDay();
```

```
switch (day) {
```

```
    case 0:
```

```
        console.log("Today is Sunday");
```

```
        break;
```

```
    case 1:
```

```
        console.log("Today is Monday");
```

```
        break;
```

```
    case 2:
```

```
        console.log("Today is Tuesday");
```

```
        break;
```



```
case 3:
    console.log("Today is Wednesday");
    break;
case 4:
    console.log("Today is Thursday");
    break;
case 5:
    console.log("Today is Friday");
    break;
case 6:
    console.log("Today is Saturday");
    break;
default:
    console.log("Invalid day");
}

// Show Vowel or Consonant using switch case in JS
const letter = "a";

switch (letter) {
    case "a":
        console.log("Vowel");
        break;
    case "e":
        console.log("Vowel");
        break;
    case "i":
        console.log("Vowel");
        break;
```

```

case "o":
  console.log("Vowel");
  break;
case "u":
  console.log("Vowel");
  break;
default:
  console.log("Consonant");
}

```

### **Q.30 What are the looping structures in JavaScript? Any one Example?**

Kinds of loops:

for - loops through a block of code a number of times.

for/in - loops through the properties of an object.

for/of - loops through the values of an iterable object.

while - loops through a block of code while a specified condition is true.

Example.

```

for (let i = 0; i < 5; i++) {
  console.log(i);
}

```

### **Q.31 Write a print 972 to 897 using for loop in JS?**

**Ans.** for (let i = 972; i >= 897; i--) {  
 console.log(i);  
}

### **Q.32 Write to print factorial of given number?**

**Ans.**function factorial(n) {  
 if (n === 0 || n === 1) {  
 return 1;  
 } else {

```

    return n * factorial(n - 1);
}
}

```

### **Q.33 Write to print Fibonacci series up to given numbers?**

**Ans.** if (n < 0) {

```

    throw new Error("n must be a non-negative integer");

```

```

}

```

```

if (n === 0 || n === 1) {

```

```

    return n;

```

```

} else {

```

```

    return fibonacci(n - 1) + fibonacci(n - 2);

```

```

}

```

```

}

```

// This function prints the Fibonacci series up to the given number

```

function printFibonacciSeries(n) {

```

```

    for (let i = 0; i < n; i++) {

```

```

        console.log(fibonacci(i));

```

```

    }

```

```

}

```

// Example usage:

```

printFibonacciSeries(10);

```

### **Q.34 Write to print number in reverse order e.g.: number = 64728 ---> reverse = 82746 in JS?**

**Ans.** let result = num1.toString().split('').reverse().join('');

### **Q.35 Write a program make a summation of given number (E.g., 1523 Ans:- 11) in JS?**

**Ans.**

```

function sumOfDigits(number) {

```

```
// Convert the number to a string.
const numberString = number.toString();

// Initialize the sum to 0.
let sum = 0;

// Iterate through the string, adding each digit to the sum.
for (let i = 0; i < numberString.length; i++) {
    sum += parseInt(numberString[i]);
}

// Return the sum.
return sum;
}

// Find the sum of the digits of 1523.
const sum = sumOfDigits(1523);

// Print the sum to the console.
console.log(sum);
```

**Q.36 Write a program you have to make a summation of first and last Digit. (E.g., 1234 Ans: 5) in JS?**

**Ans.**

```
function sumFirstAndLastDigit(number) {
    // Convert the number to a string.
    const numberString = number.toString();

    // Get the first and last digits.
    const firstDigit = numberString[0];
```

```

const lastDigit = numberString[numberString.length - 1];

// Sum the first and last digits.
const sum = parseInt(firstDigit) + parseInt(lastDigit);

// Return the sum.
return sum;
}

// Example usage:
const sum = sumFirstAndLastDigit(1234);
console.log(sum); // Output: 5

```

**Q.37 Use console.log() and escape characters to print the following pattern in JS?**

```

1 1 1 1 1
2 1 2 4 8
3 1 3 9 27
4 1 4 16 64
5 1 5 25 125

```

**Ans.**

```

console.log("1 1 1 1 1\n2 1 2 4 8\n3 1 3 9 27\n4 1 4 16 64\n5 1 5 25 125");

```

**Q.38 Use pattern in console.log in JS?**

```

1)
1
1 0
1 0 1

```

**1 0 1 0**

**1 0 1 0 1**

**2)**

**A**

**B C**

**D E F**

**G H I J**

**K L M N O**

**Ans.**

```
let n = 5;
let string = "";
// External loop
for (let i = 1; i <= n; i++) {
  // printing characters
  for (let j = 0; j < i; j++) {
    string += String.fromCharCode((i - 1) + 65);
  }
  string += "\n";
}
console.log(string);
```

**3)**

**1**

**2 3**

**4 5 6**

**7 8 9 10**

**11 12 13 14 15**

**Ans.**

```

let n = 5; // height of pattern
let string = "";
let count = 1;
// External loop
for (let i = 1; i <= n; i++) {
  // Internal loop
  for (let j = 1; j <= i; j++) {
    string += count;
    count++;
  }
  string += "\n";
}
console.log(string);

```

**4)**

```

*
* *
* * *
* * * *
* * * * *

```

**Ans.**

```

let n = 5;
let string = "";
for (let i = 1; i <= n; i++) {
  for (let j = 0; j < i; j++) {
    string += "*";
  }
  string += "\n";
}
console.log(string);

```

### **Q.39 Accept 3 numbers from user using while loop and check each numbers palindrome?**

**Ans.**

```
function checkPalindrome(number) {  
    // Convert the number to a string.  
    const numberString = number.toString();  
    // Reverse the string.  
    const reversedString = numberString.split("").reverse().join("");  
    // Check if the original string and the reversed string are equal.  
    return numberString === reversedString;  
}  
  
// Accept three numbers from the user.  
let numbers = [];  
while (numbers.length < 3) {  
    const number = prompt("Enter a number:");  
    numbers.push(number);  
}  
  
// Check each number to see if it is a palindrome.  
for (const number of numbers) {  
    const isPalindrome = checkPalindrome(number);  
    if (isPalindrome) {  
        console.log(` ${number} is a palindrome.`);  
    } else {  
        console.log(` ${number} is not a palindrome.`);  
    }  
}
```

### **Q.40 Write a JavaScript Program to display the current day and time in the following format.**



**Sample Output: Today is Friday. Current Time is 12 PM:  
12 : 22 2 ?**

**Ans.**

```
// Create a new Date object
var today = new Date();
// Get the current day
var day = today.getDay();
// Get the current time
var hours = today.getHours();
var minutes = today.getMinutes();
var seconds = today.getSeconds();
// Convert the hours to 12-hour format
hours = hours % 12 || 12;
// Get the AM/PM indicator
var ampm = hours >= 12 ? "PM" : "AM";
// Display the current day and time
console.log("Today is " + day + ".");
console.log("Current Time is " + hours + ":" + minutes + ":" +
seconds + " " + ampm);
```

**Q.41 Write a JavaScript program to get the current date?**

**Ans.**

```
// Create a new Date object
const today = new Date();
// Get the date as a string
const dateString = today.toDateString();
// Display the date
console.log("Date:", dateString);
```

## Q.42 Write a JavaScript program to compare two objects?

**Ans.**

```
function compareObjects(obj1, obj2) {  
    // Check if the objects are the same reference.  
    if (obj1 === obj2) {  
        return true;  
    }  
    // Check if the objects have the same number of properties.  
    if (Object.keys(obj1).length !== Object.keys(obj2).length) {  
        return false;  
    }  
    // Check if the objects have the same properties.  
    for (const key in obj1) {  
        if (!obj2.hasOwnProperty(key)) {  
            return false;  
        }  
    }  
    // Check if the values of the properties are the same.  
    if (obj1[key] !== obj2[key]) {  
        return false;  
    }  
}  
  
// The objects are equal.  
return true;  
}  
  
// Example usage:  
const obj1 = {  
    name: "Alice",  
    age: 30,
```

```
};
```

```
const obj2 = {  
  name: "Alice",  
  age: 30,  
};
```

```
const areEqual = compareObjects(obj1, obj2);
```

```
console.log(areEqual); // true
```

### **Q.43 Write a JavaScript program to convert an array of objects into CSV string?**

**Ans.**

```
function compareObjects(obj1, obj2) {  
  // Check if the objects are the same reference.  
  if (obj1 === obj2) {  
    return true;  
  }  
  
  // Check if the objects have the same number of properties.  
  if (Object.keys(obj1).length !== Object.keys(obj2).length) {  
    return false;  
  }  
  
  // Check if the objects have the same properties.  
  for (const key in obj1) {  
    if (!obj2.hasOwnProperty(key)) {  
      return false;  
    }  
  }  
}
```

```

    // Check if the values of the properties are the same.
    if (obj1[key] !== obj2[key]) {
        return false;
    }
}

// The objects are equal.
return true;
}

```

// Example usage:

```

const obj1 = {
  name: "Alice",
  age: 30,
};

```

```

const obj2 = {
  name: "Alice",
  age: 30,
};

```

```

const areEqual = compareObjects(obj1, obj2);

```

```

console.log(areEqual); // true

```

#### **Q.44 Write a JavaScript program to capitalize first letter of a string?**

**Ans.**

```

function capitalizeFirstLetter(string) {
  return string.charAt(0).toUpperCase() + string.slice(1);
}

```

```
}  
const str = "hello world";  
const capitalizedStr = capitalizeFirstLetter(str);  
console.log(capitalizedStr); // "Hello world"
```

**Q. 45 Write a JavaScript program to determine if a variable is array?**

**Ans.** let str = 'This is a string';  
let ans = Array.isArray(str);  
console.log("Output for String: " + ans);

**Q.46 Write a JavaScript program to clone an array?**

**Ans.**  
const originalArray = [1, 2, 3];  
const clonedArray = originalArray.slice(0, originalArray.length);  
  
console.log(clonedArray); // [1, 2, 3]

**Q.47 What is the drawback of declaring methods directly in JavaScript objects?**

**Ans. This question is repeted**

**Q.48 Print the length of the string on the browser console using console.log()?**

**Ans.** console.log ("Hello World". length);

**Q.49 Change all the string characters to capital letters using toUpperCase() method?**

**Ans.**  
const string = "hello world";  
const uppercaseString = string.toUpperCase();  
console.log(uppercaseString);

**Q.50 What is the drawback of declaring methods directly in JavaScript objects?**

**Ans.** In this method, calling a function is slightly longer and if we have bunch of extra nested functions, it would be more longer  
Second method requires some extra processing by JS engine

**Q.51 Write a JavaScript program to get the current date. Expected Output : mm-dd-yyyy.**

**Ans.**

```
// Get the current date
var today = new Date();
// Get the month (adding 1 because months are zero-based)
var mm = today.getMonth() + 1;
// Get the day of the month
var dd = today.getDate();
// Get the year
var yyyy = today.getFullYear();
// Add leading zero if the day is less than 10
if (dd < 10) {
    dd = '0' + dd;
}
// Add leading zero if the month is less than 10
if (mm < 10) {
    mm = '0' + mm;
}
// Format the date as mm-dd-yyyy
today = mm + '-' + dd + '-' + yyyy;

// Log the date to the console
console.log(today);
```

**Q.52 Use indexOf to determine the position of the first occurrence of a in 30 Days Of JavaScript?**

**Ans.**

```
const str = "30 Days Of JavaScript";  
const index = str.indexOf("a");  
console.log(index); // 4
```

### **Q,53 Use lastIndexOf to determine the position of the last occurrence of a in 30 Days Of JavaScript?**

**Ans.**

```
const str = "30 Days Of JavaScript";  
const lastIndex = str.lastIndexOf("a");  
console.log(lastIndex); // 12
```

### **Q.54 Form Validtion in JS?**

**Ans.** It is important to validate the form submitted by the user because it can have inappropriate values. So, validation is must to authenticate user.

JavaScript provides facility to validate the form on the client-side so data processing will be faster than server-side validation. Most of the web developers prefer JavaScript form validation.

Through JavaScript, we can validate name, password, email, date, mobile numbers and more fields.

### **Q.55 Form in Email, number, Password, Validation?**

**Ans.**

```
<!DOCTYPE html>  
<html>  
  <head>  
    <title>Form Validation</title>  
  </head>  
  <body>  
    <h1>Form Validation</h1>  
  
    <form action="/action_page.php" method="post">  
      <label for="email">Email: </label>
```

```
<input type="email" id="email" name="email">
```

```
<label for="password">Password: </label>
```

```
<input type="password" id="password" name="password">
```

```
<input type="submit" value="Submit">
```

```
</form>
```

```
<script>
```

```
    // Email validation
```

```
    function validateEmail(email) {
```

```
        var re =
```

```
/^((([^\<>()\\[\]\/\.,;:\s@"]+(\.[^\<>()\\[\]\/\.,;:\s@"]+)*)|(".+"))@((\[[0-9]{1,3}\. [0-9]{1,3}\. [0-9]{1,3}\. [0-9]{1,3}\]|([a-zA-Z\0-9]+\.)+[a-zA-Z]{2,})))$/;
```

```
        return re.test(email);
```

```
    }
```

```
    // Password validation
```

```
    function validatePassword(password) {
```

```
        var re = /^(?=.*\d)(?=.*[a-z])(?=.*[A-Z]).{8,}$/;
```

```
        return re.test(password);
```

```
    }
```

```
    // Form validation
```

```
    function validateForm() {
```

```
        var email = document.getElementById("email").value;
```

```
        var password = document.getElementById("password").value;
```

```
        if (!validateEmail(email)) {
```

```
            alert("Invalid email address");
```

```
            return false;
```

```
        }
```



```

        if (!validatePassword(password)) {
            alert("Invalid password");
            return false;
        }

        return true;
    }

    // Add event listener to the form submit button
    document.getElementById("myForm").addEventListener("submit",
    validateForm);

</script>
</body>
</html>

```

## Q.56 Dynamic Form Validation in JS?

**Ans.** Dynamic form validation in JavaScript is a technique that allows you to validate forms that are created dynamically, meaning that the form fields and their validation rules can change at runtime. This can be useful for a variety of scenarios, such as creating forms that allow users to add or remove fields, or forms that are populated with data from a database.

## Q.57 how many type of JS Event? How to use it ?

**Ans.**

Mouse events: These events are triggered when the user interacts with the mouse, such as clicking, hovering, or scrolling.

Keyboard events: These events are triggered when the user presses or releases a key on the keyboard.

Form events: These events are triggered when the user interacts with a form element, such as submitting a form or changing the value of a field.

Browser events: These events are triggered when the user interacts with the browser window, such as resizing or scrolling the window.

## Q.59 What is Bom vs Dom in JS?

**Ans.** DOM: Objects that have to do with the currently loaded page (the page is also called the document)

BOM: Objects that deal with everything outside the page (the browser window and the desktop screen)

### **Q.60 Array vs object defences in JS?**

**Ans.** Arrays and objects are both data structures in JavaScript that are used to store data.

Arrays are ordered collections of data. This means that each element in an array has a specific index, and the elements are stored in the order in which they were added to the array. Arrays can store any type of data, including primitive data types and objects.

Objects are unordered collections of data. This means that the elements in an object do not have a specific index, and they can be stored in any order. Objects can store any type of data, primitive data types, arrays, and other objects.

### **Q.61 Split the string into an array using split() Method?**

**Ans.**

```
const string = "This is a string.";
const array = string.split(" ");

console.log(array); // ["This", "is", "a", "string."]
```

### **Q.62 Check if the string contains a word Script using includes() method?**

**Ans.**

```
const str = "This is an example string.";
const word = "Script";

if (str.includes(word)) {
  console.log("The string contains the word 'Script'.");
} else {
  console.log("The string does not contain the word 'Script'.");
}
```

### **Q.63 Change all the string characters to lowercase letters using toLowerCase() Method.**

**Ans.**

```
const myString = "HeLLo WoRLD!";  
const lowerCaseString = myString.toLowerCase();  
console.log(lowerCaseString);
```

**Q.64 What is Character at index 15 in '30 Days of JavaScript' string? Use charAt() method.**

**Ans.**

```
let text = "30 Days of JavaScript";  
let letter = text.charAt(15);  
ans is "s".
```

**Q.65 copy to one string to another string in JS?**

**Ans.**

```
const originalString = "Hello, world!";  
const newString = originalString.slice(0, 5);  
console.log(newString); // "Hello"
```

**Q.66 Find the length of a string without using libraryFunction?**

- **What is JavaScript?**

**Ans.** JavaScript is a text-based programming language used both on the client-side and server-side that make web pages interactive.

- **What is the use of is NaN function?**

**Ans.** The isNaN() function is used to check if a value is Not a Number (NaN).

- **What is negative Infinity?**

**Ans.** Negative infinity in JavaScript is a constant value that represents the lowest possible value.

- **Which company developed JavaScript?**

**Ans.** JavaScript was developed by Netscape Communications Corporation. It was initially released in 1995 as LiveScript, but was renamed JavaScript in 1996 to take advantage of the popularity of Java.

- **What are undeclared and undefined variables?**

**Ans.** In JavaScript, a variable is undeclared if it hasn't been declared with a keyword like var, let, or const. An undefined variable is a variable that has been declared but not assigned a value.

- **Write the code for adding new elements dynamically?**

**Ans.** // Create a new element

```
const newElement = document.createElement("div");
```

```
// Set the element's text content
```

```
newElement.textContent = "This is a new element.";
```

```
// Append the new element to the body of the document
```

```
document.body.appendChild(newElement);
```

- **What is the difference between ViewState and SessionState?**

**Ans.** Session state is saved on the server, ViewState is saved in the page.

Session state is usually cleared after a period of inactivity from the user

The view state is posted on subsequent post back in a hidden field.

- **What is === operator?**

**Ans.** The strict equality operator (===) in JavaScript is used to check for strict equality between two values.

- **How can the style/class of an element be changed?**

**Ans.** To change any old class name with a new class the syntax is  
`document.getElementById('myElement').className = "myclass";`

- **How to read and write a file using JavaScript?**

**Ans.** The fs. readFile() and rs. writeFile() methods are used to read and write of a file using javascript.

- **What are all the looping structures in JavaScript?**

**Ans.**

```
for (let i = 0; i < 5; i++) {  
    // code block to be executed
```

}

- **How can you convert the string of any base to an integer in JavaScript?**

**Ans.**

```
var x = parseInt(string name);
```

- **What is the function of the delete operator?**

**Ans.** The delete operator in JavaScript removes a property from an object.

- **What are all the types of Pop up boxes available in JavaScript?**

**Ans.** JavaScript has three kind of popup boxes: Alert box, Confirm box, and Prompt box.

- **What is the use of Void (0)?**

**Ans.** By running void(0) in the URL JavaScript code, nothing is evaluated or returned.

- **How can a page be forced to load another page in JavaScript?**

**Ans.** window.location.href = "new\_url"; is a syntax to load another page

- **What are the disadvantages of using innerHTML in JavaScript?**

**Ans.** It is very slow because as inner HTML already parses the content even we have to parse the content again so that's why it takes time.