

Ans:-1

js:-high-level programming language

```
document.write("hello World?" + "<br>");
```

Ans:-2

3 type of variable

Ans:-3

primitive data type

1. number

2. string

3. boolean

4. undifind

5. null

6. symbol

7. bigInt

non-primitive

1. array

2. object

Ans:-4

```
let a = 3;
```

```
    let b = 4;
```

```
    let c = a * b;
```

```
    document.write(c + "<br>");
```

Ans:-5

undefined:-variable has been declared but not assigned

ex: let x;

```
document.write(x);
```

undeclared:-variables being used without being declared first using "var","const","let"

ex:- document.write(y);

Ans:-6

```
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." + "<br>");
```

```
console.log('I am quoting Mother Teresa: "If you judge people, you have no time to love them."' + "<br>");
```

Ans:-7

```
if (typeof '10' !== 'number') {  
    let num = parseInt('10');  
    console.log(num);  
} else {  
    console.log('The type of \'10\' is exactly equal to number.');
```

Ans:-8

```
let base = 200  
  
let height = 300  
  
let area = 0.5 * base * height;  
  
document.write(area + "<br>")
```

Ans:-9

```
function daysUntilChristmas() {  
    let today = new Date();  
  
    let currentYear = today.getFullYear();  
  
    let christmasDate = new Date(currentYear, 11, 25);  
  
    if (today.getMonth() === 11 && today.getDate() > 25) {  
        christmasDate.setFullYear(currentYear + 1);  
    }  
  
    let timeDiff = christmasDate.getTime() - today.getTime();  
  
    let daysLeft = Math.ceil(timeDiff / (1000 * 3600 * 24));
```

```
        return daysLeft;
    }

    let daysLeft = daysUntilChristmas();

    console.log("Days left until next Christmas:", daysLeft + "<br>");
```

Ans:-10

A condition statement, often referred to as a conditional statement or simply a condition, is a programming construct

used to execute different code blocks based on whether a certain condition evaluates to true or false. Condition

statements allow you to control the flow of your program's execution based on logical conditions.

Ans:-11

```
let A = 10;

let C = 4 * A;

document.write(C + "<br>");
```

Ans:-12

```
function yearsToDays(years) {
    return years * 365.25;
}

function daysToYears(days) {
    return days / 365.25;
}
```

```
let years = 5;
```

```
let days = 1826;
```

```
console.log(years + " years is equal to " + yearsToDays(years) + " days." + "<br>");
```

```
console.log(days + " days is equal to " + daysToYears(days) + " years." + "<br>");
```

Ans:-13

```

function fahrenheitToCelsius(fahrenheit) {
    return (fahrenheit - 32) / 1.8;
}

let temperatureFahrenheit = 32;

let temperatureCelsius = fahrenheitToCelsius(temperatureFahrenheit);

console.log(temperatureFahrenheit + " degrees Fahrenheit is equal to " +
temperatureCelsius.toFixed(2) + " degrees Celsius.");

if (temperatureCelsius > 0) {
    console.log("The temperature is above freezing.");
} else if (temperatureCelsius < 0) {
    console.log("The temperature is below freezing.");
} else {
    console.log("The temperature is at freezing point.");
}

```

Ans:-14

```
let parts = filename.split('.');
```

Ans:-15

true

Ans:-16

true

Ans:-17

False

Ans:-18

loop:-In JavaScript, a loop is a programming construct that allows you to repeatedly execute a block of code as long as

a specified condition is true. Loops are used to automate repetitive tasks, iterate over arrays or objects, and traverse

data structures.

switch:-A switch case is another programming construct used for decision-making. It's often used as an alternative to

long if...else if...else chains when you have multiple possible conditions to evaluate.

Ans:-19

The isNaN() function in JavaScript is used to determine whether a value is "Not-a-Number" (NaN). It returns true if the

value is NaN, and false if it's a valid number or can be converted into a valid number

Ans:-20

AND(&&):-The && operator returns true if both of its operands are true, otherwise, it returns false.

OR(||):-The || operator returns true if at least one of its operands is true, otherwise, it returns false.

Ans:-21

In JavaScript, void(0) is an expression that evaluates to undefined. It's commonly used in situations where you want to force the return value of an expression to be undefined

Ans:-22

Let A=3;

If(A>0){

Document.write("positive");

}

Else if(A<0){

Document.write("negative");

}

Else{

Document.write("zero");

Ans:23

function isVowel(character) {

character = character.toLowerCase();

return ['a', 'e', 'i', 'o', 'u'].includes(character);

}

// Example usage:

```
console.log(isVowel('a')); // Output: true
```

```
console.log(isVowel('b')); // Output: false
```

```
console.log(isVowel('E')); // Output: true (converted to lowercase)
```

Ans:-24

```
Let num= 3
```

```
If(num%2==0){
```

```
Document.write("even")
```

```
}
```

```
Else{
```

```
document.write("odd")
```

```
}
```

Ans:-25

```
Let num=4
```

```
(num%2==0)?document.write("even"):document.write("odd");
```

Ans:26

```
Let num1=23
```

```
Let num2=20
```

```
Let num3=30
```

```
(num1 > num2) ? ((num1 > num3) ? num1 : num3) : ((num2 > num3) ? num2 : num3);
```

Ans:-27

```
Let num1=40
```

```
Let num2=30
```

```
Let num3=70
```

```
(num1 < num2) ? ((num1 < num3) ? num1 : num3) : ((num2 < num3) ? num2 : num3);
```

Ans:-28

```
Let num1=2
```

```
Let num2=4
```

Let num3=8

(num1 > num2) ? ((num1 > num3) ? num1 : num3) : ((num2 > num3) ? num2 : num3);

Ans:-29(i)

```
let dayName;
```

```
switch (dayNumber) {
```

```
    case 1:
```

```
        dayName = "Monday";
```

```
        break;
```

```
    case 2:
```

```
        dayName = "Tuesday";
```

```
        break;
```

```
    case 3:
```

```
        dayName = "Wednesday";
```

```
        break;
```

```
    case 4:
```

```
        dayName = "Thursday";
```

```
        break;
```

```
    case 5:
```

```
        dayName = "Friday";
```

```
        break;
```

```
    case 6:
```

```
        dayName = "Saturday";
```

```
        break;
```

```
    case 7:
```

```
        dayName="Sunday";
```

```
        break;
```

```
    default:
```

```
        dayName = "Invalid day number";
```

Ans:-29(ii)

```
function checkVowelOrConsonant(character) {  
    character = character.toLowerCase();  
    if ((character >= 'a' && character <= 'z')) {  
        switch (character) {  
            case 'a':  
            case 'e':  
            case 'i':  
            case 'o':  
            case 'u':  
                return "Vowel";  
            default:  
                return "Consonant";  
        }  
    } else {  
        return "Not a letter";  
    }  
}
```

Ans:-30

In JavaScript, there are several looping structures that allow you to execute a block of code repeatedly. The most commonly used looping structures are:

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



```

let i = 0;
do {
    console.log(i);
    i++;
} while (i < 5);
const obj = { a: 1, b: 2, c: 3 };
for (const key in obj) {
    console.log(key + ': ' + obj[key]);
}
const arr = [1, 2, 3];
for (const element of arr) {
    console.log(element);
}

```

Ans:-31

```

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

```

Ans:-32

```

function factorial(num) {
    if (num === 0 || num === 1) {
        console.log(1);
    } else {
        let result = 1;
        for (let i = 2; i <= num; i++) {
            result *= i;
        }
        Document.write(result);
    }
}

```

Ans:-33

```
function fibonacciSeries(limit) {  
    let fibArray = [0, 1];  
  
    while (fibArray[fibArray.length - 1] + fibArray[fibArray.length - 2] <= limit) {  
        fibArray.push(fibArray[fibArray.length - 1] + fibArray[fibArray.length - 2]);  
    }  
  
    Document.write(fibArray);  
}
```

Ans:-34

```
function reverseNumber(number) {  
    const reversed = parseInt(number.toString().split('').reverse().join(''));  
    document.wrtie(reversed);  
}
```

Ans:35

```
function digitSum(number) {  
    let sum = 0;  
    while (number) {  
        sum += number % 10; // Add the last digit to sum  
        number = Math.floor(number / 10); // Remove the last digit  
    }  
    Document.write(sum);  
}
```

Ans:-36

```
function sumFirstAndLastDigit(number) {  
    const lastDigit = number % 10; // Extract the last digit  
    let firstDigit = number;  
    while (firstDigit >= 10) {
```

```

        firstDigit = Math.floor(firstDigit / 10);
    }

    Document.write(firstDigit + lastDigit;)
}

```

Ans:-37

```

for (let i = 1; i <= 5; i++) {
    let row = "";
    for (let j = 1; j <= 5; j++) {
        if (j === 1) {
            row += i + ' ';
        } else {
            row += Math.pow(i, j) + ' ';
        }
    }
    console.log(row);
}

```

Ans:-38

(i)

```

for (let i = 1; i <= 5; i++) {
    let row = "";
    for (let j = 1; j <= i; j++) {
        if (j % 2 === 0) {
            row += '0 ';
        } else {
            row += '1 ';
        }
    }
    console.log(row);
}

```

(ii)

```
let currentCharCode = 65;
```

```
for (let i = 1; i <= 5; i++) {  
    let row = "";  
    for (let j = 1; j <= i; j++) {  
        row += String.fromCharCode(currentCharCode) + ' '  
        currentCharCode++;    }  
    console.log(row);  
}
```

(iii)

```
let count = 1;
```

```
for (let i = 1; i <= 5; i++) {  
    let row = "";  
    for (let j = 1; j <= i; j++) {  
        row += count + ' '  
        count++;  
    }  
    console.log(row);  
}
```

(iv)

```
for (let i = 1; i <= 5; i++) {  
    let row = "";  
    for (let j = 1; j <= 5; j++) {  
        if (j <= i) {  
            row += '* '  
        } else {  
            row += ' '  
        }  
    }  
}
```

```
    }  
    console.log(row);  
  }  
}
```

Ans:-39

```
const readline = require('readline');
```

```
const rl = readline.createInterface({  
  input: process.stdin,  
  output: process.stdout  
});
```

```
function isPalindrome(number) {  
  const numString = number.toString();  
  const reversedNumString = numString.split('').reverse().join('');  
  return numString === reversedNumString;  
}
```

```
let count = 1;
```

```
const checkPalindrome = () => {  
  rl.question(`Enter number ${count}: `, (number) => {  
    if (isNaN(number)) {  
      console.log("Invalid input. Please enter a valid number.");  
      checkPalindrome();  
      return;  
    }  
    if (isPalindrome(number)) {  
      console.log(`${number} is a palindrome.`);  
    } else {
```

```

        console.log(`${number} is not a palindrome.`);
    }

    if (count < 3) {
        count++;
        checkPalindrome();
    } else {
        rl.close();
    }
});
}

```

```
checkPalindrome();
```

Ans:-40

```

function getCurrentDay() {
    const days = ['Sunday', 'Monday', 'Tuesday', 'Wednesday', 'Thursday', 'Friday', 'Saturday'];
    const currentDate = new Date();
    const dayIndex = currentDate.getDay();
    return days[dayIndex];
}

```

```

function getCurrentTime() {
    const currentDate = new Date();
    let hours = currentDate.getHours();
    const ampm = hours >= 12 ? 'PM' : 'AM';
    hours = hours % 12;
    hours = hours ? hours : 12; // Handle midnight (0 hours)
    const minutes = currentDate.getMinutes();
    const seconds = currentDate.getSeconds();
}

```

```
    return `${hours} ${ampm}: ${minutes} : ${seconds}`;  
}
```

```
const currentDay = getCurrentDay();
```

```
const currentTime = getCurrentTime();
```

```
console.log(`Today is ${currentDay}. Current Time is ${currentTime}`);
```

Ans:-41

```
function getCurrentDate() {
```

```
    const currentDate = new Date();
```

```
    const year = currentDate.getFullYear();
```

```
    let month = currentDate.getMonth() + 1; // Month starts from 0
```

```
    let day = currentDate.getDate();
```

```
    // Add leading zero if month or day is less than 10
```

```
    if (month < 10) {
```

```
        month = '0' + month;
```

```
    }
```

```
    if (day < 10) {
```

```
        day = '0' + day;
```

```
    }
```

```
    // Return date in the format YYYY-MM-DD
```

```
    return `${year}-${month}-${day}`;
```

```
}
```

```
const currentDate = getCurrentDate();
```

```
console.log("Current Date:", currentDate);
```

Ans:-42

```

function compareObjects(obj1, obj2) {
  const keys1 = Object.keys(obj1);
  const keys2 = Object.keys(obj2);
  if (keys1.length !== keys2.length) {
    return false;
  }
  for (let key of keys1) {
    if (obj1[key] !== obj2[key]) {
      return false;
    }
  }
  return true;
}

```

Ans:-43

```

function convertArrayToCSV(array) {
  const headers = Object.keys(array[0]);
  const headerRow = headers.join(',') + '\n';
  const csvRows = [];
  array.forEach(obj => {
    const rowValues = headers.map(key => obj[key]);
    const rowString = rowValues.join(',');
    csvRows.push(rowString);
  });
  const csvString = headerRow + csvRows.join('\n');
  return csvString;
}

```

Ans:-44

```

function capitalizeFirstLetter(str) {
  if (str.length === 0) {

```



```
        return str;
    }

    return str.charAt(0).toUpperCase() + str.slice(1);
}
```

Ans:-45

```
function isArray(variable){
    return Array.isArray(variable);
}
```

Ans:-46

```
function cloneArray(arr) {
    return arr.slice();
}
```

Ans:-47

Lack of encapsulation

Increased memory usage

Limited reusability

Difficulty in testing

Inflexibility

Ans:-48

```
const str = "Hello, World!";
console.log("Length of the string:", str.length);
```

Ans:-49

```
const str = "hello, world!";
const uppercaseStr = str.toUpperCase();
console.log(uppercaseStr);
```

Ans:-50

Lack of encapsulation

Increased memory usage

Limited reusability

Difficulty in testing

Inflexibility

Ans:-51

```
function getCurrentDate(format) {  
    const currentDate = new Date();  
    const day = String(currentDate.getDate()).padStart(2, '0');  
    const month = String(currentDate.getMonth() + 1).padStart(2, '0');  
    const year = currentDate.getFullYear();  
  
    // Determine the desired format  
    switch (format) {  
        case 'mm-dd-yyyy':  
            return `${month}-${day}-${year}`;  
        case 'mm/dd/yyyy':  
            return `${month}/${day}/${year}`;  
        case 'dd-mm-yyyy':  
            return `${day}-${month}-${year}`;  
        case 'dd/mm/yyyy':  
            return `${day}/${month}/${year}`;  
        default:  
            return 'Invalid format';  
    }  
}
```

Ans:52

```
const str = "30 Days Of JavaScript";  
const position = str.indexOf('a');  
console.log("Position of the first occurrence of 'a':", position);
```

Ans:53

```
const str = "30 Days Of JavaScript";
```

```
const position = str.lastIndexOf('a');  
console.log("Position of the last occurrence of 'a':", position);
```

Ans:-54

```
<html>  
</body>  
<form id="myForm" onsubmit="return validateForm()">  
  <label for="username">Username:</label>  
  <input type="text" id="username" name="username" required>  
  
  <label for="email">Email:</label>  
  <input type="email" id="email" name="email" required>  
  
  <button type="submit">Submit</button>  
</form>  
<Script>  
function validateForm() {  
  const username = document.getElementById('username').value;  
  const email = document.getElementById('email').value;  
  
  if (username.trim() === '') {  
    alert('Please enter a username');  
    return false;  
  }  
  
  if (email.trim() === '') {  
    alert('Please enter an email address');  
    return false;  
  }  
}
```

```
    return true;
}
```

```
</script>
```

```
</body>
```

```
</html>
```

Ans:-55

```
<html>
```

```
<body>
```

```
<form id="myForm" onsubmit="return validateForm()">
```

```
    <label for="email">Email:</label>
```

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

```
    <label for="phoneNumber">Phone Number:</label>
```

```
    <input type="tel" id="phoneNumber" name="phoneNumber" pattern="[0-9]{3}-[0-9]{3}-[0-9]{4}"
    required>
```

```
    <small>Format: XXX-XXX-XXXX</small>
```

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

```
    <input type="password" id="password" name="password" minlength="6" required>
```

```
    <button type="submit">Submit</button>
```

```
</form>
```

```
<script>
```

```
function validateForm() {
```

```
    const email = document.getElementById('email').value;
```

```
    const phoneNumber = document.getElementById('phoneNumber').value;
```

```
    const password = document.getElementById('password').value;
```

```
    if (!isValidEmail(email)) {
```

```
        alert('Please enter a valid email address');
```

```

        return false;
    }
    if (!isValidPhoneNumber(phoneNumber)) {
        alert('Please enter a valid phone number (Format: XXX-XXX-XXXX)');
        return false;
    }
    if (password.length < 6) {
        alert('Password must be at least 6 characters long');
        return false;
    }

    return true;
}

function isValidEmail(email) {
    const emailRegex = /^[^\s@]+@[^\s@]+\.[^\s@]+$/;
    return emailRegex.test(email);
}

function isValidPhoneNumber(phoneNumber) {
    const phoneRegex = /^\\d{3}-\\d{3}-\\d{4}$/;
    return phoneRegex.test(phoneNumber);
}
</script>
</body>
</html>

```

Ans:-56

```

<html>
<body>
<form id="myForm">
    <label for="email">Email:</label>

```

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

```
<span id="emailError" class="error"></span>
```

```
<label for="phoneNumber">Phone Number:</label>
```

```
<input type="tel" id="phoneNumber" name="phoneNumber">
```

```
<span id="phoneError" class="error"></span>
```

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

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

```
<span id="passwordError" class="error"></span>
```

```
<button type="submit">Submit</button>
```

```
</form>
```

```
<script>
```

```
document.addEventListener('DOMContentLoaded', function() {
```

```
    const emailInput = document.getElementById('email');
```

```
    const phoneInput = document.getElementById('phoneNumber');
```

```
    const passwordInput = document.getElementById('password');
```

```
    const form = document.getElementById('myForm');
```

```
    emailInput.addEventListener('input', validateEmail);
```

```
    phoneInput.addEventListener('input', validatePhoneNumber);
```

```
    passwordInput.addEventListener('input', validatePassword);
```

```
    form.addEventListener('submit', function(event) {
```

```
        event.preventDefault();
```

```
        if (validateEmail() && validatePhoneNumber() && validatePassword()) {
```

```
            // Submit the form if all validations pass
```

```
            form.submit();
```

```
    } else {  
        // Display error message  
        alert('Form submission failed. Please correct errors.');
```

```
    }  
});  
  
function validateEmail() {  
    const email = emailInput.value;  
    const emailError = document.getElementById('emailError');  
    if (!email || !isValidEmail(email)) {  
        emailError.textContent = 'Please enter a valid email address';  
        return false;  
    }  
    emailError.textContent = '';  
    return true;  
}
```

```
function validatePhoneNumber() {  
    const phoneNumber = phoneInput.value;  
    const phoneError = document.getElementById('phoneError');  
    if (phoneNumber && !isValidPhoneNumber(phoneNumber)) {  
        phoneError.textContent = 'Please enter a valid phone number (Format: XXX-XXX-XXXX)';  
        return false;  
    }  
    phoneError.textContent = '';  
    return true;  
}
```

```
function validatePassword() {
```

```

const password = passwordInput.value;
const passwordError = document.getElementById('passwordError');
if (password.length < 6) {
    passwordError.textContent = 'Password must be at least 6 characters long';
    return false;
}
passwordError.textContent = '';
return true;
}

// Email validation function
function isValidEmail(email) {
    // Regular expression for basic email validation
    const emailRegex = /^[^\s@]+@[^\s@]+\.[^\s@]+$/;
    return emailRegex.test(email);
}

// Phone number validation function
function isValidPhoneNumber(phoneNumber) {
    // Regular expression for phone number with format XXX-XXX-XXXX
    const phoneRegex = /^\d{3}-\d{3}-\d{4}$/;
    return phoneRegex.test(phoneNumber);
}
});
</script>
</body>
</html>

```

Ans:-57

Identify the type of event you want to handle (e.g., click, keydown, submit).

Select the DOM element you want to attach the event to.

Attach an event listener to the element using the `addEventListener()` method.

Provide a callback function that will be executed when the event occurs.

Ans:-59

DOM (Document Object Model): Represents the structure of an HTML document as a tree of objects, allowing JavaScript to access and manipulate the content, structure, and style of the document.

BOM (Browser Object Model): Represents everything else in the browser beyond the document, providing objects and interfaces for interacting with the browser window, controlling browser behavior, managing client-side storage, and more.

Ans:-60

Arrays are ordered collections of elements, typically used for storing lists of similar items, and they support methods for iteration and manipulation.

Objects are unordered collections of key-value pairs, used for storing data with named properties, and they provide efficient key-based lookup and manipulation.

Ans:-61

```
const str = "Hello, world!";  
const arr = str.split(',');  
console.log(arr); // Output: ["Hello", " world!"]
```

Ans:-62

```
const str = "JavaScript is a scripting language.";  
if (str.includes("Script")) {  
    console.log("The string contains the word 'Script'.");  
} else {  
    console.log("The string does not contain the word 'Script'.");  
}
```

Ans:-63

```
const str = "Hello, WORLD!";  
const lowercaseStr = str.toLowerCase();  
console.log(lowercaseStr);
```

Ans:-64

```
const str = '30 Days of JavaScript';  
const charAtIndex15 = str.charAt(15);  
console.log("Character at index 15:", charAtIndex15); // Output: "J"
```