**Basic JS**

**1. Variable Declarations: let, var, const**

**- var**

* **Scope**: Function-scoped or globally scoped.
* **Re-declarable**: Can be re-declared in the same scope.
* **Hoisting**: Variables are hoisted to the top of their scope.

**- let**

* **Scope**: Block-scoped.
* **Re-declarable**: Cannot be re-declared in the same scope.
* **Hoisting**: Hoisted, but not initialized until declared (Temporal Dead Zone).

- **const**

* **Scope**: Block-scoped.
* **Re-declarable**: Cannot be re-declared in the same scope.
* **Immutability**: Value cannot be reassigned (but object properties can be modified).

**2. Loops: for, while, do-while**

**for Loop**

Used for iterating over a range or collection.

for (let i = 0; i < 5; i++) {

console.log(i); // Output: 0 1 2 3 4

}

**while Loop**

Executes as long as the specified condition is true.

let i = 0;

while (i < 5) {

console.log(i); // Output: 0 1 2 3 4

i++;

}

**do-while Loop**

Executes at least once and then continues while the condition is true.

let j = 0;

do {

console.log(j); // Output: 0 1 2 3 4

j++;

} while (j < 5);

**3. Array Methods**

**Common Array Methods**

* **push()**: Adds an element to the end of an array.
* **pop()**: Removes the last element from an array.
* **shift()**: Removes the first element from an array.
* **unshift()**: Adds an element to the beginning of an array.
* **map()**: Creates a new array with the results of calling a function on every element.
* **filter()**: Creates a new array with elements that pass a test.
* **reduce()**: Executes a reducer function on each element, resulting in a single value.
* **forEach()**: Executes a provided function once for each array element.

**Example**

let arr = [1, 2, 3];

arr.push(4); // arr = [1, 2, 3, 4]

let doubled = arr.map(x => x \* 2); // doubled = [2, 4, 6, 8]

**4. Basic Functions**

**Function Declaration**

function add(a, b) {

return a + b;

}

**Function Expression**

const subtract = function(a, b) {

return a - b;

};

**Example of Function Usage**

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

console.log(subtract(5, 3)); // Output: 2

**5. Objects, Deep and Shallow Copy**

**Objects**

Objects are collections of key-value pairs.

const person = {

name: 'Rohanshu',

age: 25

};

**Shallow Copy**

A shallow copy creates a new object, but nested objects are still referenced.

const shallowCopy = { ...person }; // Using spread operator

**Deep Copy**

A deep copy creates a new object and recursively copies all nested objects.

const deepCopy = JSON.parse(JSON.stringify(person));

**Example**

person.name = 'rohan';

console.log(shallowCopy.name); // Output: rohan

console.log(deepCopy.name); // Output: Rohanshu

**6. String Methods**

**Common String Methods**

* **length**: Returns the length of the string.
* **toUpperCase()**: Converts the string to uppercase.
* **toLowerCase()**: Converts the string to lowercase.
* **trim()**: Removes whitespace from both ends.
* **slice(start, end)**: Extracts a section of the string.
* **split(separator)**: Splits the string into an array of substrings.

**Example**

let str = " Hello World ";

console.log(str.trim()); // Output: "Hello World"

console.log(str.toUpperCase()); // Output: " HELLO WORLD "

**7. DOM (Document Object Model)**

**What is DOM?**

The DOM is a programming interface for web documents. It represents the page so that programs can change the document structure, style, and content.

**Common DOM Manipulations**

* **document.getElementById()**: Selects an element by its ID.
* **document.getElementsByClassName()**: Selects elements by their class name.
* **document.querySelector()**: Selects the first matching element using CSS selectors.
* **document.createElement()**: Creates a new HTML element.
* **appendChild()**: Adds a new child node to an element.

**Example**

const newDiv = document.createElement('div');

newDiv.textContent = "Hello, World!";

document.body.appendChild(newDiv);

**8. Events in HTML using JavaScript**

**What are Events?**

Events are actions that occur in the browser, such as user interactions (clicks, key presses) or browser actions (loading a page).

**Adding Event Listeners**

You can respond to events using the addEventListener method.

document.getElementById("myButton").addEventListener("click", function() {

alert("Button clicked!");

});

**9. Various Event Names**

**Common Event Names**

* click: Fired when an element is clicked.
* mouseenter: Fired when the mouse enters an element.
* mouseleave: Fired when the mouse leaves an element.
* keydown: Fired when a key is pressed down.
* keyup: Fired when a key is released.
* submit: Fired when a form is submitted.
* load: Fired when the page has finished loading.

**Example**

window.addEventListener("load", function() {

console.log("Page loaded");

});

**10. Form Validation using JavaScript**

**Basic Form Validation**

JavaScript can be used to validate form inputs before they are submitted.

**11. Math Object**

**Common Math Methods**

* **Math.random()**: Returns a random number between 0 and 1.
* **Math.floor(x)**: Rounds down to the nearest integer.
* **Math.abs(x)**: Returns the absolute value.
* **Math.round(x)**: Rounds to the nearest integer.
* **Math.max(...numbers)**: Returns the largest of zero or more numbers.
* **Math.PI**: The ratio of the circumference of a circle to its diameter.
* **Math.log10(x)**: Returns the base 10 logarithm of a number.
* **Math.trunc(x)**: Returns the integer part of a number by removing any fractional digits.
* **Math.sqrt(x)**: Returns the square root of a number.
* **Math.cos(x)**: Returns the cosine of a number.

**12. Date Object**

**Creating Dates**

* You can create a date using the Date constructor with various parameters.