

1.What is a variable in JavaScript?

- **Variable is used to store value or storing information.**

2.How do you declare a variable in JavaScript?

- **var**
- **Let**
- **Const**

3.What are the differences between var, let, and const?

- **Var: Defines a variable that can be accessed globally, meaning it can be accessed outside of functions.**
- **Let: Defines a variable that is local to the block (typically a function), restricting its access to only within that function.**

- **Const:** Defines a variable that cannot be changed after it is initially assigned, providing immutable values.

#### 4. Explain variable hoisting in JavaScript.

- **Variable hoisting in JavaScript** refers to the behavior where variable declarations are moved to the top of their containing scope during the compilation phase, before the code is executed.

```
console.log(myVar); // Outputs: undefined  
  
var myVar = 10;
```

5. What are the scoping rules for var, let, and const?

### Var

- Variables declared with var are function-scoped. They are visible throughout the function in which they are declared, regardless of block boundaries .

### Const:

- Variables declared with are also block-scoped like let. initialized with a value when declared and cannot be reassigned.

### **Let:**

- **Variables declared with `let` are block-scoped. They are only visible within the nearest enclosing block , including loops, conditions, and functions.**

## **6.How can you use template literals in JavaScript?**

- **In JavaScript, template literals provide an easy and more readable way to create strings that include embedded expressions or variables.**

```
let name = 'Alice';
```

```
let greeting = `Hello, ${name}!`;
```

```
console.log(greeting) // Output: Hello, Alice!
```

## **7.List the primitive data types in JavaScript.**

- **String, Number, BigInt, Boolean, Null**

8. What is the difference between null and undefined?

- **Null** means there is an empty value in variable
- **Undefined** means not any value is stored yet in the variable

9. How do you check the type of a variable in JavaScript?

- Using `typeof` method

10. Explain the difference between primitive and reference data types.

- **Primitive Data Types:** Represent simple, single values like numbers (1, 3.14), strings ('Hello', "World"),

**booleans (true, false), undefined, null, and symbols**

- **Reference Data Types: Represent objects stored and accessed by reference, including { }, arrays ( [ ] ), functions, dates, and custom objects. They are copied by reference and allow for mutable properties.**

## **11. How does type coercion work in JavaScript?**

- **Type coercion in JavaScript refers to the automatic conversion of values from one data type to another, typically during operations like comparisons or arithmetic involving different types.**

12. What are the `typeof` operator and the `instanceof` operator used for?

- **`typeof` operator:** Used to determine the data type of a variable or an expression. It returns a string indicating the type of the operand.

**Ex:**

`typeof 42`

returns "number"

- **`instanceof` operator:** Used to check if an object is an instance of a specific class or constructor function's prototype chain.

13. How do you convert a string to a number in JavaScript?

**Using parseInt**

**Let x="10"**

**y=parseInt(x)**

14. How do you convert a number to a string in JavaScript?

**let number = 42;**

**let str = number.toString();**

15. What is implicit type conversion?

- **Implicit type conversion, also known as type coercion, is when JavaScript automatically converts a value from one data type to another without the programmer explicitly specifying it.**



16. What are the different methods to convert a string to a number? Explain with examples.

```
let str = '123';
```

```
let num = Number(str); /
```

```
let str = '42';
```

```
let num = parseInt(str);
```

17. How can you handle type conversion when adding a number and a string?

- To handle type conversion when adding a number and a string in JavaScript, explicitly convert the string to a number using functions like `Number ( )` or `parseFloat ( )` before performing the addition operation. This ensures predictable behavior and

**prevents unintended concatenation of strings with numbers.**

- 

**18. Explain how parseInt() and parseFloat() functions work.**

**parseInt():**

**Converts a string into an integer (whole number) based on the specified radix (base).**

```
let str = '42 apples';
```

```
let num = parseInt(str);
```

**parseFloat():**

**Converts a string into a floating-point number (decimal number).**

```
let str = '3.14 meters';
```

```
let num = parseFloat(str);
```

19. What are arrays and how do you declare them?

- **Arrays in JavaScript are data structures that store multiple values sequentially under a single variable name.**

```
let fruit=['apple', 'banana', 'cherry']
```

20. What is an object in JavaScript?

- **In JavaScript, an object is a collection of key-value pairs where each key is a unique string (or symbol) and each value can be of any data type, including other objects or functions. Objects allow for structured data storage and manipulation within the language.**