Questions (JavaScript)

1. **What is a variable in JavaScript?**

   A variable in JavaScript is a named container that holds a value, which can be data like numbers, strings, objects, etc. It allows you to store and manipulate data in your programs.

2. **How do you declare a variable in JavaScript?**

   We can declare a variable in JavaScript using the var, let, or const keywords.

   For example:

   ```
   var name = "John";
   let age = 30;
   const isStudent = true;
   ```

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

   var: *Function-scoped or globally scoped, can be redeclared, and is hoisted.*

   let: *Block-scoped, cannot be redeclared within the same scope, and is not hoisted in the same way as var.*

   const: *Block-scoped, cannot be reassigned after initialization, and like let, is not hoisted.*

4. **Explain variable hoisting in JavaScript?**

   Variable hoisting in JavaScript means that variable declarations (not initializations) are moved to the top of their containing scope during the compile phase. This means you can use variables before they are declared in the code, though their value will be undefined until the actual declaration is reached.

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

   var: *Function-scoped. It is accessible anywhere within the function it is declared.*

   let and const*: Block-scoped. They are accessible only within the block (delimited by {}) they are declared.*

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

   Template literals are enclosed by backticks (`) and allow embedded expressions using ${expression} syntax.

For example:

```
let name = "Alice";
let greeting = `Hello, ${name}!`;
console.log(greeting); // Output: Hello, Alice!
```

7. List the primitive data types in JavaScript.

- *String*
- *Number*
- *Boolean*
- *Null*
- *Undefined*
- *Symbol*
- *BigInt*

8. What is the difference between null and undefined?

***null**: Represents the intentional absence of any object value. It is an assignment value.*

***undefined**: Represents a variable that has been declared but not yet assigned a value.*

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

You can use the typeof operator to check the type of a variable.

For example:

```
let age = 25;
console.log(typeof age); // Output: "number"
```

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

***Primitive data types:** Store simple values and are immutable (e.g., number, string, boolean).*

***Reference data types:** Store references to objects (e.g., object, array, function). They are mutable and can have properties and methods.*

11. How does type coercion work in JavaScript?

Type coercion is the automatic or implicit conversion of values from one data type to another.

Example: 5 + '5' results in '55' because the number 5 is coerced into a string.

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

typeof: *Used to determine the type of a variable.*

For example, typeof 42 returns "number".

instanceof: *Used to determine whether an object is an instance of a specific class or constructor.*

For example, obj instanceof Array checks if obj is an array.

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

can use the Number() function, parseInt(), or parseFloat().

For example:

```
let str = "123";
let num = Number(str); // 123
let int = parseInt(str); // 123
let float = parseFloat("123.45"); // 123.45
```

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

You can use the String() function or the toString() method.

For example:

```
let num = 123;
let str = String(num); // "123"
let str2 = num.toString(); // "123"
```

15. What is implicit type conversion?

Implicit type conversion, or type coercion, is when JavaScript automatically converts one data type to another. For example, in console.log('5' + 5), JavaScript converts the number 5 to a string, resulting in the output '55'.

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

Number(): *Converts a string to a number.*

Example: Number("123") returns 123.

parseInt(): *Converts a string to an integer.*

Example: parseInt("123") returns 123.

parseFloat(): *Converts a string to a floating-point number.*

Example: parseFloat("123.45") returns 123.45.

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

To ensure correct type conversion, explicitly convert the string to a number before adding.

For example:

```
let num = 5;
let str = "10";
let sum = num + Number(str); // 15
```

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

parseInt(): *Parses a string and returns an integer. It stops parsing when it encounters a non-digit character.*

Example: parseInt("123abc") returns 123.

parseFloat(): *Parses a string and returns a floating-point number. It stops parsing when it encounters a non-digit character (except for the first decimal point).*

Example: parseFloat("123.45abc") returns 123.45.

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

Arrays are list-like objects used to store multiple values in a single variable. You declare an array using square brackets [ ].

Example:

```
let fruits = ["apple", "banana", "cherry"];
```

20. What is an object in JavaScript?

An object is a collection of properties, where each property is a key-value pair. Objects can store data and functions (methods). You can declare an object using curly braces {}.

Example:

```
let person = {
  name: "John",
  age: 30,
  greet: function() {
    console.log("Hello, " + this.name);
  }
};
```