Q1:->What are anonymous functions in JavaScript?

Ans:-> The definition of the term "anonymous" is "unknown or without identification." An anonymous function in JavaScript is a function that has no name, or more precisely, one that lacks a name. An anonymous function has no identification when it is created.

Q2:-> Explain strict comparison and Abstract comparison in javascript?

Ans:-> In JavaScript, **strict comparison (`===` and `!==`)** checks both the value and type of operands without any type conversion. For example, `5 === '5'` returns `false` because one is a number and the other is a string. On the other hand, **abstract comparison (`==` and `!=`)** performs type conversion if the types differ before comparing the values. For instance, `5 == '5'` returns `true` because the string `'5'` is converted to a number. Strict comparison is generally preferred to avoid unexpected results caused by type coercion.

Q3:-> Difference b/w arrow functions and regular functions?

Ans:-> Arrow functions and regular functions in JavaScript differ in syntax, behavior, and how they handle certain features like `this` binding.

1. **Syntax**: Arrow functions have a concise syntax: `const add = (a, b) => a + b;`, while regular functions use the `function` keyword: `function add(a, b) { return a + b; }`.
2. **`this` Binding**: Arrow functions do not bind their own `this` but inherit it from the surrounding context. Regular functions have their own `this`, determined by how they are called.
3. **Arguments Object**: Arrow functions do not have their own `arguments` object, whereas regular functions do.
4. **Usage**: Arrow functions are ideal for concise callbacks and functional programming, while regular functions are better for methods or scenarios requiring dynamic `this`.

Q4:-> What is Hoisting in JavaScript?

Ans:-> Hoisting in JavaScript is a built-in behavior that moves variable, function, and class declarations to the top of their scope before code execution. This allows you to use these declarations before they are officially declared in your code.

Q5:-> JavaScript is a garbage collected programming language, explain how?

Ans:-> JavaScript is a garbage-collected programming language because it automatically allocates memory when objects are created and frees it when they are no longer in use

Q6:-> Explain Shallow copy vs Deep copy in Javascript?
Ans:-> In JavaScript, a shallow copy creates a new object or array that references the original object's elements, while a deep copy creates a completely independent copy of the original object

Q7:-> What is Object.freeze?
Ans:-> In JavaScript, `Object.freeze()` is a static method that freezes an object, making it read-only and preventing any changes:

- **No new properties**: New properties cannot be added
- **No removed properties**: Existing properties cannot be removed
- **No changed values**: The values of existing properties cannot be changed