Answers to Question 1:

1. What do you understand by hoisting in JavaScript?

\*Hoisting\* is a JavaScript mechanism where variables and function declarations are moved to the top of their containing scope before code execution. This means you can use variables and functions before you declare them in your code.

2. Why is super used in JavaScript?

The \*super\* keyword is used to call the constructor of the parent class and to access the parent's properties and methods in a subclass. It is primarily used in class-based inheritance.

3. What is let and const? And how it differs from var?

\*let\*: It declares a block-scoped local variable, optionally initializing it to a value. It cannot be redeclared in the same scope.

\*const\*: It declares a block-scoped, read-only named constant. The value must be initialized during the declaration and cannot be reassigned.

\*var\*: It declares a function-scoped or globally-scoped variable, optionally initializing it to a value. It can be redeclared and updated in the same scope.

Differences:

1.Scope: `let` and `const` are block-scoped, while `var` is function-scoped or globally-scoped.

2.Hoisting: Variables declared with `var` are hoisted and initialized with `undefined`, while `let` and `const` are hoisted but not initialized.

3.Reassignment and Redeclaration: `var` can be redeclared and updated, `let` can be updated but not redeclared, and `const` cannot be updated or redeclared.

4. Discuss the Rest parameter in ES6. What is an Arrow function? What are all its uses? How it differs from a normal function?

\*Rest Parameter\*: The rest parameter syntax (`...args`) allows a function to accept an indefinite number of arguments as an array.

\*Arrow Function\*: Arrow functions provide a shorter syntax for writing functions and lexically bind the `this` value. They are useful for non-method functions and callbacks.

Differences from Normal Functions:

1.Syntax: Arrow functions have a concise syntax.

2.`this` Binding: Arrow functions do not have their own `this` context; they inherit `this` from the parent scope.

3.No `arguments` Object: Arrow functions do not have the `arguments` object.

4.Cannot be used as Constructors: Arrow functions cannot be used with the `new` keyword.

5. What is the difference between the readonly and disabled attributes for the <textarea> element?

\*readonly\*: The `readonly` attribute allows users to view and select the content of the textarea but prevents them from modifying it.

\*disabled\*: The `disabled` attribute prevents users from interacting with the textarea entirely. It cannot be focused, and its content is not submitted with the form.

6. How do you specify units in CSS? What are the different ways to do it?

Units in CSS can be specified using different types of measurements:

- \*\*Absolute Units\*\*:

- `px` (pixels)

- `pt` (points)

- `cm` (centimeters)

- `mm` (millimeters)

- `in` (inches)

- \*\*Relative Units\*\*:

- `%` (percentage relative to the parent element)

- `em` (relative to the font size of the element)

- `rem` (relative to the font size of the root element)

- `vw` (viewport width)

- `vh` (viewport height)

7. What property is used for changing the font face?

The `font-family` property is used to change the font face in CSS.

8. How to center align a div inside another div? [2 Ways]

- \*\*Using Flexbox\*\*

- \*\*Using CSS Grid\*\*

- \*\*Using Margin Auto (for horizontal centering)\*\*

Answers To Question 3 :

1. JavaScript is ……

The correct answer is:2. asynchronous, non-blocking, single-threaded language.

JavaScript operates as a single-threaded language but utilizes asynchronous programming to handle non-blocking operations like I/O tasks, allowing other code to run while waiting for these operations to complete.

2. is the concept of object-oriented programming used to hide the internal representation, or state, of an object from the outside

The correct answer is:2) Encapsulation

Encapsulation is a fundamental concept in object-oriented programming that restricts access to certain components of an object and prevents the internal state from being accessed directly from outside the object's methods.