November 5th

Question 1: Explain what JavaScript is and its role in web development.

JavaScript is a high-level, dynamic programming language primarily used for adding interactivity and functionality to websites. While HTML defines the structure and CSS handles styling, JavaScript is responsible for making websites interactive, allowing them to respond to user actions such as clicks, form submissions, or animations.

Role in web development:

- Enhances user experience by dynamically updating content.
- Controls multimedia (e.g., videos, audio) and animates graphics.
- Handles form validations and user input.
- Communicates with servers using AJAX for updating data without refreshing the page.

Question 2: Explain the key differences between JavaScript and HTML. Provide examples of situations where you would use each.

- **HTML (Hyper-Text Markup Language)** is a markup language used to structure web pages by defining elements such as headings, paragraphs, links, and images.

  Example: Use HTML to define the layout of a Webpage.

  ```
  <h1>Welcome to My Website</h1>
  <p>This is a paragraph.</p>
  ```

- **JavaScript**, on the other hand, is a programming language used to make the webpage dynamic and interactive by manipulating HTML elements.

  Example: Use JavaScript to change the text of a paragraph when a button is clicked.

  ```
  document.getElementById('myButton').onclick = function() {
      document.getElementById('myText').innerText = "You clicked the
  ```

```
button!";
};
```

In summary, HTML structures content, while JavaScript adds behavior and interaction to that content.

Question 3: List and describe the five primitive data types in JavaScript.

1. **Number**: Represents numeric values (both integers and floating-point numbers).
   - Example: `let age = 25;`
2. **String**: Represents a sequence of characters or text enclosed in single or double quotes.
   - Example: `let name = "John Doe";`
3. **Boolean**: Represents a logical entity with two values: `true` or `false`.
   - Example: `let isLoggedIn = true;`
4. **Undefined**: Indicates a variable has been declared but not yet assigned a value.
   - Example: `let x; console.log(x); // undefined`
5. **Null**: Represents the intentional absence of any object or value.
   - Example: `let selectedItem = null;`

Question 4: What is the purpose of declaring variables in JavaScript, and how do you declare them using the 'let' keyword?

The purpose of declaring variables is to store data in memory so that it can be referenced and manipulated later in the program. Variables hold data like numbers, strings, or objects, making your code dynamic.

Declaration using `let`:

- `let` allows you to declare a block-scoped variable, which means its value can be updated but only within the block or function it is declared.

```
let age = 25;
age = 26; // You can update the value of 'age'
```

Variables declared with `let` are useful for scenarios where you expect the value to change over time.

Question 5: Explain the importance of comments in JavaScript and provide examples of single-line and multi-line comments.

Comments in JavaScript are used to explain code and make it easier for developers (and others) to understand what the code is doing. They are ignored by the JavaScript engine and do not affect the program execution.

- **Single-line comments** are useful for brief explanations:

```
// This is a single-line comment
let x = 10; // Set x to 10
```

- **Multi-line comments** are useful for more detailed descriptions or when disabling blocks of code:

```
/*
 This is a multi-line comment.
 It spans across multiple lines.
*/
let y = 20;
```

Question 6: Explain the importance of choosing meaningful and descriptive variable names in JavaScript. Provide an example where using a clear identifier improves code readability.

Choosing **meaningful and descriptive variable names** improves code readability, making it easier to understand the purpose of each variable. Well-named variables allow others (or even yourself, when revisiting code) to quickly grasp what the variable represents, reducing errors and improving maintainability.

Example:

- Less descriptive:

```
let x = 300;
```

```
let y = 400;
```

It's unclear what `x` and `y` represent.

- More descriptive:

```
let screenWidth = 300;
let screenHeight = 400;
```

Now it's clear that these variables represent the screen dimensions, which improves readability and understanding.