Here you go 👍 — **very long, clear, exam-oriented answers** for **all 10 JavaScript questions**, written exactly in the style expected for **LPU / MCA / Web Technologies** exams.

---

1. Explain JavaScript? Explain its application in HTML?

JavaScript is a **high-level, interpreted, object-based scripting language** used to make web pages **dynamic and interactive**. It was initially developed to run in web browsers but is now also used on servers and other platforms.

JavaScript works together with **HTML and CSS**:

- HTML defines the structure of a webpage
- CSS controls the presentation and styling
- JavaScript controls the behavior and interaction

Applications of JavaScript in HTML:

JavaScript can be embedded into HTML to:

- Validate form inputs (e.g., checking empty fields)
- Manipulate HTML elements dynamically
- Handle user events such as clicks, key presses, and mouse movements
- Change content without reloading the page
- Create animations and interactive UI components

Example:

<button onclick="alert('Hello JavaScript!')">Click Me</button>

JavaScript enhances user experience by making webpages responsive and interactive.

---

2. Which IDE can be used implementing JavaScript?

JavaScript can be written and executed using various **IDEs and code editors**.

Common IDEs for JavaScript:

- **Visual Studio Code** – Most popular, lightweight, powerful extensions
- **WebStorm** – Advanced IDE with debugging tools
- **Sublime Text** – Fast and minimal editor
- **Atom** – Open-source and customizable
- **Notepad++** – Simple editor for beginners
- **Brackets** – Designed for web development

Explanation:

Most IDEs provide features like:

- Syntax highlighting
- Auto-completion
- Debugging support
- Error detection

Visual Studio Code is widely preferred due to its simplicity and strong community support.

---

3. What is equality in JavaScript?

Equality in JavaScript is used to **compare two values** to check whether they are equal.

JavaScript provides two main equality operators:

1. Loose Equality (==)

- Compares values after **type conversion**

```
5 == "5"   // true
```

2. Strict Equality (===)

- Compares values **without type conversion**
- Checks both value and type

```
5 === "5"  // false
```

Conclusion:

Strict equality (===) is recommended because it avoids unexpected results.

---

4. What's the difference between Host objects and Native objects?

Native Objects:

Native objects are built into the JavaScript language itself.

Examples:

- Object
- Array
- String
- Number
- Boolean
- Function
- Date

They are available in all JavaScript environments.

Host Objects:

Host objects are provided by the **environment** in which JavaScript runs (usually the browser).

Examples:

- window
- document
- navigator
- console
- location

Key Difference:

- Native objects are part of JavaScript
- Host objects depend on the runtime environment

---

5. How to compare two objects in JavaScript?

In JavaScript, objects are compared by **reference**, not by value.

Example:

```
let obj1 = { a: 1 };

let obj2 = { a: 1 };


obj1 == obj2   // false
```

Even though both objects contain the same data, they are stored in different memory locations.

Correct way:

```
let obj3 = obj1;

obj1 === obj3   // true
```

To compare object values, you must manually compare properties or convert them to strings.

---

6. How JavaScript can be used in innerHTML?

innerHTML is a property used to **get or set HTML content** inside an element.

Example:

```
<p id="demo"></p>


<script>

document.getElementById("demo").innerHTML = "Hello World";
```

</script>

Uses:

- Dynamically update content
- Display results without page reload
- Create dynamic UI changes

innerHTML is powerful but should be used carefully to avoid security issues.

---

7. Explain Prototype Inheritance in JavaScript?

JavaScript uses **prototype-based inheritance**, not class-based inheritance like Java.

Every JavaScript object has a hidden property called **prototype**, which allows it to inherit properties and methods from another object.

Example:

```
function Person(name) {
  this.name = name;
}

Person.prototype.sayHello = function() {
  return "Hello " + this.name;
};

let p1 = new Person("John");
```

Here, p1 inherits sayHello() from Person.prototype.

Benefits:

- Memory efficient
- Dynamic inheritance
- Flexible object creation

---

8. What is the new keyword in JavaScript?

The new keyword is used to **create an instance of an object** from a constructor function.

Example:

```
function Car(model) {
  this.model = model;
```

}

let car1 = new Car("Tesla");

What new does:

1. Creates a new object
2. Sets prototype
3. Binds this
4. Returns the object

---

9. Is JavaScript a pass-by-reference or pass-by-value language?

JavaScript is **pass-by-value**, but the behavior differs for primitive and reference types.

Primitive types:

- Passed by value

let a = 10;

Objects:

- Object references are passed by value

let obj = {};

Explanation:

- Primitives create a copy
- Objects pass the reference address

Thus, JavaScript is technically pass-by-value.

---

10. Why Script Tag is Used in JavaScript?

The <script> tag is used to **embed JavaScript code in an HTML document** or link an external JavaScript file.

Example:

<script>

alert("Hello JavaScript");

</script>

External JS:

<script src="script.js"></script>

Uses:

- Executes JavaScript in browser
- Links external JS files
- Enables interactivity

The <script> tag acts as a bridge between HTML and JavaScript.
