INDEX

Basics of JavaScript :

JavaScript Introduction :

JavaScript is a programming language that makes websites and web applications interactive. It allows you to add cool features like animations, interactive forms, and dynamic content to web pages.

JavaScript is executed by your web browser, so you don't need any special software to use it. You can write JavaScript code directly in your HTML document or include it from external files. It's a versatile and essential tool for modern web development.

JavaScript is indeed an object-oriented language. It allows you to work with objects and manipulate them to create interactive and dynamic web applications.

JavaScript is considered lightweight because it's designed to be easy for developers to learn and use. It doesn't have a steep learning curve compared to some other languages, which makes it accessible for beginners.

JavaScript is cross-platform, meaning it can run on various operating systems and web browsers. This cross-compatibility is one of the reasons why it's so widely used for web development. Developers can write JavaScript code that works consistently across different devices and browsers.

Use of JavaScript :

JavaScript has a wide range of uses in web development

Web Development: JavaScript is primarily used to enhance the interactivity and functionality of websites. It can be used to create dynamic web pages, handle user input, and update content. This includes things like form validation, image sliders, and real-time updates.

Server side : While JavaScript is primarily known for client-side web development, it can also be used on the server-side by using Node.js for building server applications.

User Interface (UI) : JavaScript can be used to improve the user experience by creating responsive and interactive user interfaces. You can add animations, transitions, and custom widgets to make web applications more engaging.

Security: It's also used to enhance security on websites by implementing features like user authentication and access control.

Way to include javascript :

1. Embedding Code: You can include JavaScript directly within your HTML file using the <script> tag.

Here's an example:

```
<script>
   // JavaScript code goes here
   alert("Hello, World!");
</script>
```

This method is quick and easy but **not recommended for larger scripts** or when you want to reuse code across multiple pages.

2. Event Handlers: You can also attach JavaScript code to specific HTML events, such as button clicks or page load events, using event handlers like onclick or onload

```
<button onclick="myFunction()">Click me</button>
<script>
   function myFunction() {
      // JavaScript code goes here
      alert("Button clicked!");
   }
</script>
```

3. External JavaScript File: To keep your code organized and reusable, it's common to place your JavaScript code in separate files with a .js extension. Then, you can link to this external file in your HTML using the <script> tag's src attribute:

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

This method is **preferred for larger scripts** and for separating your HTML and JavaScript for **better maintainability**.

Syntax of JavaScript :

```
// How to create variables:

var x;

let y;

// How to use variables:

x = 5;

y = 6;

let z = x + y;
```

The JavaScript syntax defines two types of values:

Fixed values :

1. Numbers are written with or without decimals:
2. Strings are text, written within double or single quotes:

Variable values :

JavaScript uses the keywords var, let and const to declare variables.

A JavaScript name must begin with:

- A letter (A-Z or a-z)
- A dollar sign ($)
- Or an underscore (_)

Subsequent characters may be letters, digits, underscores, or dollar signs.

Basic event of JavaScript

In JavaScript, events are actions or occurrences that happen in the browser, typically as a result of user interactions or other activities. You can use JavaScript to listen for these events and respond to them with specific functions or code.

Syntax
<element event='some JavaScript'> or <element event="some JavaScript">

Mouse events:

Event Performed	Event Handler	Description
click	onclick	When mouse click on an element
mouseover	onmouseover	When the cursor of the mouse comes over the element
mouseout	onmouseout	When the cursor of the mouse leaves an element
mousedown	onmousedown	When the mouse button is pressed over the element
mouseup	onmouseup	When the mouse button is released over the element
mousemove	onmousemove	When the mouse movement takes place.

Keyboard events:

Event Performed	Event Handler	Description
Keydown & Keyup	onkeydown & onkeyup	When the user press and then release the key

Form events:

Event Performed	Event Handler	Description
focus	onfocus	When the user focuses on an element
submit	onsubmit	When the user submits the form
blur	onblur	When the focus is away from a form element
change	onchange	When the user modifies or changes the value of a form element

Window/Document events:

Event Performed	Event Handler	Description
load	onload	When the browser finishes the loading of the page
unload	onunload	When the visitor leaves the current webpage, the browser unloads it
resize	onresize	When the visitor resizes the window of the browser

Basic Validation with JavaScript :

JavaScript form validation is a way to ensure that the data entered by the user is accurate, complete, and properly formatted before it is submitted to the server.

JavaScript validation is performed on the client-side, before the data is sent to the server.

However, JavaScript validation is not as secure as backend sanitization, which is the process of validating the data on the server-side.

In general, it is a good idea to use both JavaScript validation and backend sanitization to ensure the safety of your forms.

Here are some of the benefits of using JavaScript form validation:

- It can help to prevent errors in the data submitted by the user.
- It can help to prevent malicious attacks, such as cross-site scripting (XSS) attacks.
- It can improve the user experience by preventing the user from submitting invalid data.

Here are some of the limitations of using JavaScript form validation:

- It can be bypassed by users who have JavaScript disabled in their browser.
- It is not as secure as backend sanitization.