Day-17

Agenda

- Dom and Dom events
- event handlers
- regular expressions
- DOM and Selectors
- Object

Dom

- The **Document Object Model** (**DOM**) is the data representation of the objects that comprise the structure and content of a document on the web.

- The Document Object Model (DOM) is a programming interface for HTML and XML documents.

- It represents the page so that programs can change the document structure, style, and content. The DOM represents the document as nodes and objects.

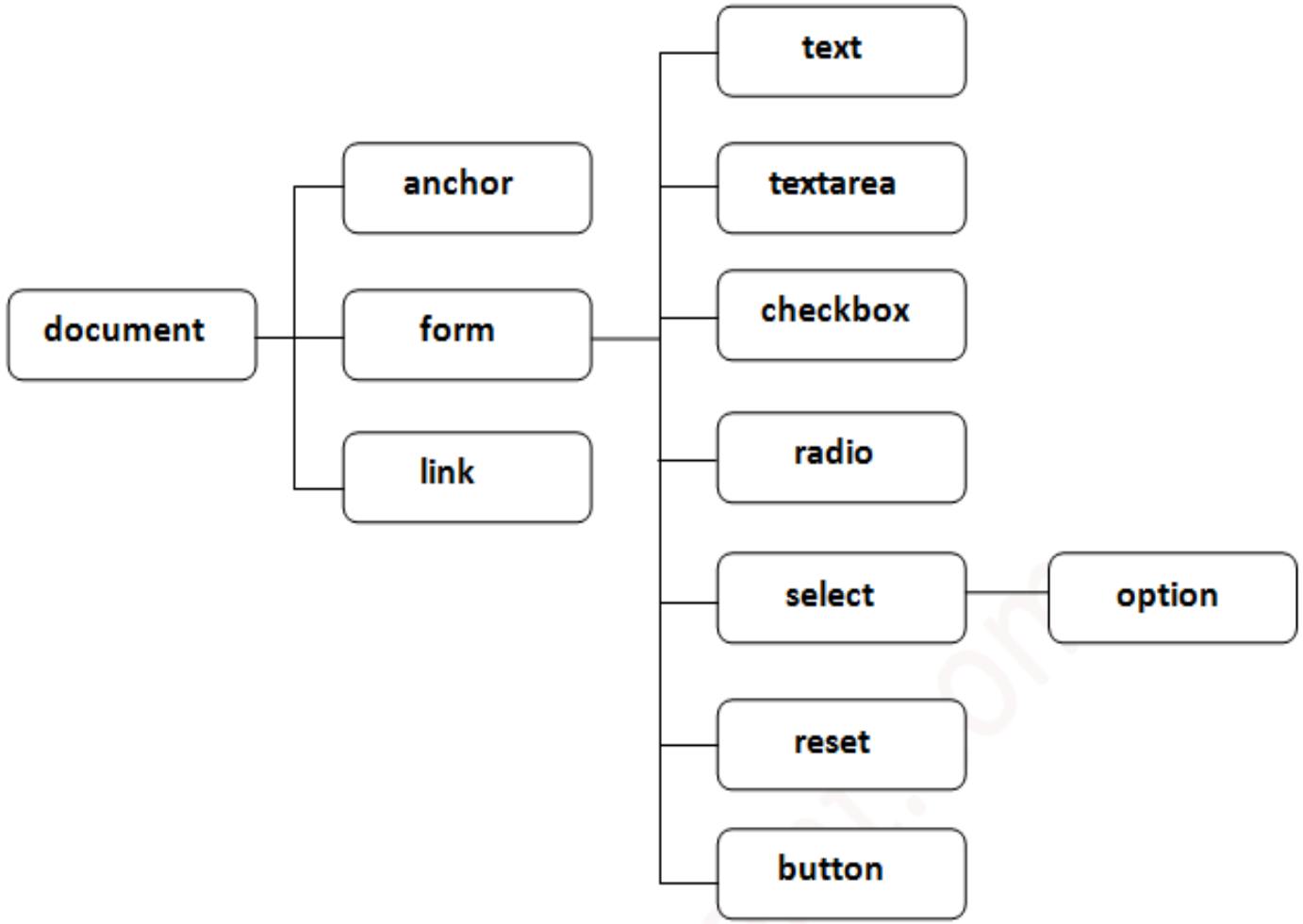
- A Web page is a document. This document can be either displayed in the browser window or as the HTML source. But it is the same document in both cases.

- The Document Object Model (DOM) represents that same document so it can be manipulated. The DOM is an object-oriented representation of the web page, which can be modified with a scripting language such as JavaScript.

- The modern DOM is built using multiple APIs that work together. The core DOM defines the objects that fundamentally describe a document and the objects within it. This is expanded upon as needed by other APIs that add new features and capabilities to the DOM.

- According to W3C - "The W3C Document Object Model (DOM) is a platform and language-neutral interface that allows programs and scripts to dynamically access and update the content, structure, and style of a document."

Properties of document object



Methods of document object

Method	Description
write("string")	writes the given string on the doucment.
writeln("string")	writes the given string on the doucment with newline character at the end.
getElementById()	returns the element having the given id value.
getElementsByName()	returns all the elements having the given name value.
getElementsByTagName()	returns all the elements having the given tag name.
getElementsByClassName()	returns all the elements having the given class name.

Accessing field value by document object

- Lets See in the code….

- The HTML DOM can be accessed with JavaScript (and with other programming languages).

- In the DOM, all HTML elements are defined as **objects**.

- The programming interface is the properties and methods of each object.

- A **property** is a value that you can get or set (like changing the content of an HTML element).

- A **method** is an action you can do (like add or deleting an HTML element).

Selecting elements

- getElementById() – select an element by id.
- getElementsByName() – select elements by name.
- getElementsByTagName() – select elements by a tag name.
- getElementsByClassName() – select elements by one or more class names.
- querySelector() – select elements by CSS selectors.

Traversing elements

- Get the parent element – get the parent node of an element.
- Get child elements – get children of an element.
- Get siblings of an element – get siblings of an element.

Manipulating elements

- createElement() – create a new element.
- appendChild() – append a node to a list of child nodes of a specified parent node.
- textContent – get and set the text content of a node.
- innerHTML – get and set the HTML content of an element.
- innerHTML vs. createElement – explain the differences beetween innerHTML and createElement when it comes to creating new elements.

- DocumentFragment – learn how to compose DOM nodes and insert them into the active DOM tree.

- insertBefore() – insert a new node before an existing node as a child node of a specified parent node.

- insertAfter() helper function – insert a new node after an existing node as a child node of a specified parent node.

- append() – insert a node after the last child node of a parent node.

- prepend() – insert a node before the first child node of a parent node.

- insertAdjacentHTML() – parse a text as HTML and insert the resulting nodes into the document at a specified position.

- replaceChild() – replace a child element by a new element.

- cloneNode() – clone an element and all of its descendants.

- removeChild() – remove child elements of a node.

Working with Attributes

- HTML Attributes & DOM Object's Properties – understand the relationship between HTML attributes & DOM object's properties.
- setAttribute() – set the value of a specified attribute on a element.
- getAttribute() – get the value of an attribute on an element.
- removeAttribute() – remove an attribute from a specified element.
- hasAttribute() – check if an element has a specified attribute or not.

Manipulating Element’s Styles

- style property – get or set inline styles of an element.
- getComputedStyle() – return the computed style of an element.
- className property – return a list of space-separated CSS classes.
- classList property – manipulate CSS classes of an element.
- Element’s width & height – get the width and height of an element.

Working with Events

- JavaScript events – introduce you to the JavaScript events, the event models, and how to handle events.
- Handling events – show you three ways to handle events in JavaScript.
- Page Load Events – learn about the page load and unload events.
- load event – walk you through the steps of handling the load event originated from the document, image, and script elements.
- DOMContentLoaded – learn how to use the DOMContentLoaded event correctly.
- beforeunload event – guide you on how to show a confirmation dialog before users leave the page.

- unload event – show you how to handle the unload event that fires when the page is completely unloaded.
- Mouse events – how to handle mouse events.
- Keyboard events – how to deal with keyboard events.
- Scroll events – how to handle scroll events effectively.
- scrollIntoView – learn how to scroll an element into view.
- Focus Events – cover the focus events.
- haschange event – learn how to handle the event when URL hash changes.

- Event Delegation – is a technique of levering event bubbling to handle events at a higher level in the DOM than the element on which the event originated.

- dispatch Event – learn how to generate an event from code and trigger it.

- Custom Events – define a custom JavaScript event and attach it to an element.

- Mutation Observer – monitor the DOM changes and invoke a callback when the changes occur.

Scripting Web Forms

- JavaScript Form – learn how to handle form submit event and perform a simple validation for a web form.
- Radio Button – show you how to write the JavaScript for radio buttons.
- Checkbox – guide you on how to manipulate checkbox in JavaScript.
- Select box – learn how to handle the select box and its option in JavaScript.
- Add / Remove Options – show you how to dynamically add options to and remove options from a select box.
- Handling change event – learn how to handle the change event of the input text, radio button, checkbox, and select elements.
- Handling input event – handle the input event when the value of the input element changes.

Thanks……