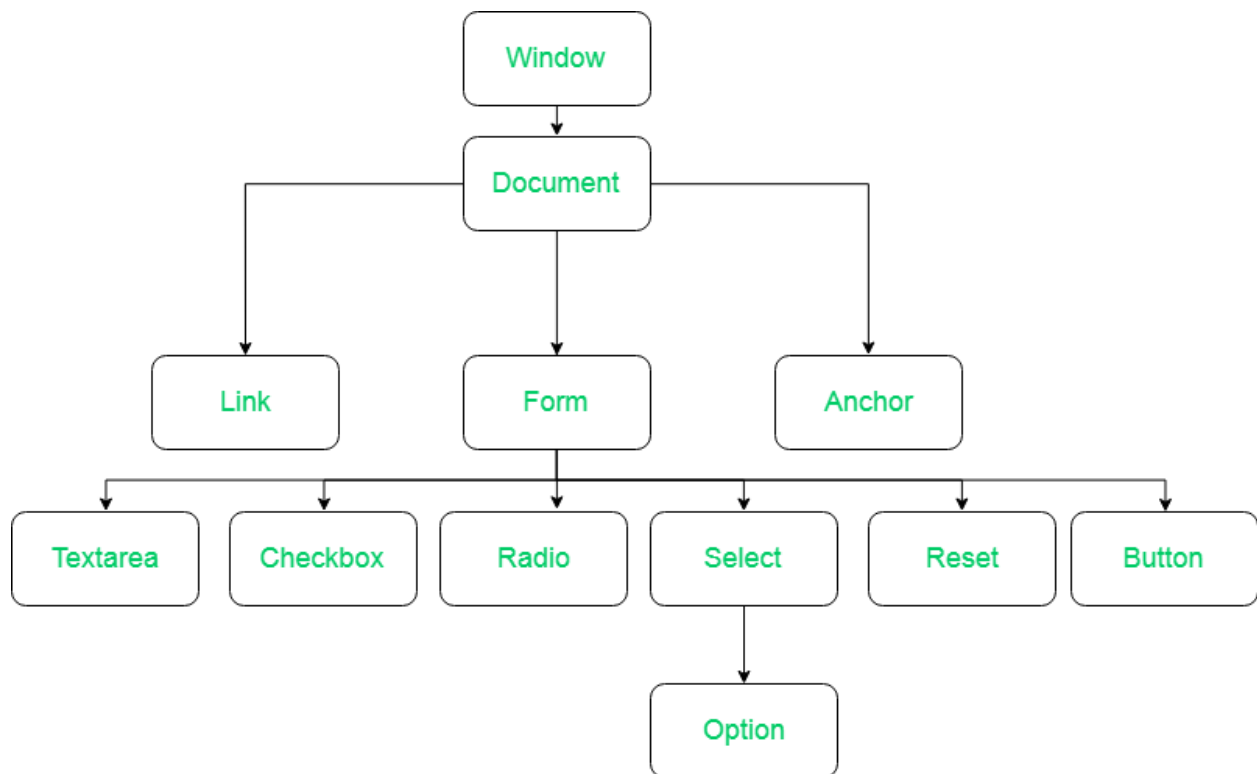


DOM (Document Object Model)

- The Document Object Model (DOM) is a programming interface for HTML and XML(Extensible markup language) documents. It defines the logical structure of documents and the way a document is accessed and manipulated.
- DOM is a way to represent the webpage in the structured hierarchical way so that it will become easier for programmers and users to glide through the document. With DOM, we can easily access and manipulate tags, IDs, classes, Attributes or Elements using commands or methods provided by Document object.



1. Window Object: Window Object is at always at top of hierarchy.
2. Document object: When HTML document is loaded into a window, it becomes a document object.
3. Form Object: It is represented by form tags.
4. Link Objects: It is represented by link tags.
5. Anchor Objects: It is represented by a href tags.
6. Form Control Elements:: Form can have many control elements such as text fields, buttons, radio buttons, and checkboxes, etc.

JavaScript BOM | Window Screen

The Browser Object Model (BOM) is a browser-specific convention referring to all the objects exposed by the web browser. The BOM allows JavaScript to “interact with” the browser.

The object of window represents a browser window and all its corresponding features. A window object is created automatically by the browser itself.

Java Script’s window.screen object contains information about the user’s screen. It can also be written without the window prefix.

Properties:

- screen.width
- screen.height
- screen.availWidth
- screen.availHeight
- screen.colorDepth
- screen.pixelDepth

Client-Side JavaScript

Client-side JavaScript is the most common form of the language. The script should be included in or referenced by an HTML document for the code to be interpreted by the browser.

It means that a web page need not be a static HTML, but can include programs that interact with the user, control the browser, and dynamically create HTML content.

The JavaScript client-side mechanism provides many advantages over traditional CGI server-side scripts. For example, you might use JavaScript to check if the user has entered a valid e-mail address in a form field.

The JavaScript code is executed when the user submits the form, and only if all the entries are valid, they would be submitted to the Web Server.

JavaScript can be used to trap user-initiated events such as button clicks, link navigation, and other actions that the user initiates explicitly or implicitly.

Create external JavaScript file with the extension .js. After creating, add it to the HTML file in the script tag. The src attribute is used to include that external JavaScript file.

If you have more than one external JavaScript file, then add it in the same web page to increase performance of the page.

Let's say the following new.js is our external JavaScript file:

```
function display() {  
    alert("Hello World!");  
}
```

Now add the external JavaScript file to the HTML web page:

```
<html>  
  <body>  
    <form>  
      <input type="button" value="Result" onclick="display()"/>  
    </form>  
    <script src="new.js">  
  </script>  
</body>  
</html>
```