JavaScript (JS)

Two types

1. Client Side Web Scripting
2. Server Side Web Scripting

Alternatives

-Applets

-ActiveX

<script>

* Linked a.k.a external scripts
* execution of linked scripts can be deferred or run asynchronously

\*Problem -before rendering of page, it have to fetch the script.js

* Embedded a.k.a internal scripts
* top-level code (i.e. code outside functions) execute as the script is loaded
* code inside functions are executed only when the function is involved

-preferred way

-anywhere

Exclusives

-defer

-async

* Inline scripts
* scripts may be embedded anywhere in the document and any top-level code is executed where it is concentered
* provides alternative content when the browser does not support or disabled

<noscript>

* JavaScript Execution Environment

-accessible

-JavaScript was built on NetScape

* Core JavaScript (EcmaScript) language and API
* (Traditional) Browser Object Model (BOM) API (non-standardized)

- Window, Navigator, Screen, Location, History

* Document Object Model (DOM) API

- Node, Document, Element, Text, Attr, DocumentType, Comment, etc.

- Event, EventTarget, Event Listener, etc.

- CSSStyleSheet, CSSRule, etc.

* Miscellaneous JavaScript Web APIs
* Geolocatiom, IndexedDB, Local Storage, Push Notifications, Service Workers, Web Sockets, Web Workers, XML HttpRequest (AJAX), etc.
* Most of the APIs are accessed from scripts via the global object **Window**
* DOM (Document Object Model)
  + API that allows access to the HTML document from within scripts associated with the web page
  + a pursed HTML document is represented by a **DOM tree**, which contain **nodes** representing **elements**, **element attributes**, **textual content**, and other HTML document (e.g DOCTYPE, comments)
  + accessed in the Document object property of the global Window object

\*the way they look in real time

DOM 4

DOM 3

DOM 2

DOM 1

Others

Functionalities:

* retrieve references to nodes (or node collections) in the DOM
  + getElementById()
  + getElementByTagName, getElementByClassName(), getElementByName()
  + querySelector(), querySelectorAll()
* transverse the DOM tree (from a given node)
* parentNode, parentElement
* childNodes, children
* firstChild, lastChild, nextSibling, previousSibling
* firstElementChild, lastElementChild, nextElementSibling, previousElementSibling
* construct/copy nodes

elements - nodeValue is always null