

Agenda

- Functions
- EcmaScript 6
- DOM

Window Object

- It represents an open window in the browser. It is browser's object(not JS object) which is created automatically
- It is a global object with lot of properties and methods

DOM (Document Object Model)

- When a webpage is loaded the browser creates DOM of the page
- It is the data representation of the objects that comprise the structure and content of a document on the web.
- DOM represents an HTML document in memory
- The DOM represents the document as nodes and objects so that programming languages can interact with the page.
- In both cases, it is the same document but the Document Object Model (DOM) representation allows it to be manipulated.
- `console.dir(document)` will display all the properties and methods from the document
- It is a tree like structure (window-> document -> html -> and all its sub nodes)

DOM Manipulation

Selection

1. Selecting with id
 - `document.getElementById("myId")` (#)
2. Selecting with class
 - `document.getElementsByClassName("myClass")` (.)
 - returns HTML collection an array of objects
3. Selecting with tag
 - `document.getElementsByTagName("tagName")`
 - returns HTML collection an array of objects
4. Query Selector
 - used to select the id,name and class automatically
 - `document.querySelector("myId/myClass/tag")`
 - returns first element
 - `document.querySelectorAll("myId/myClass/tag")`
 - returns a NodeList

Properties

1. tagName
 - returns tag for element nodes

2. innerText

- returns text content of the element and all its children
- It represents only the text part

3. innerHTML

- returns the plain text or html contents in the elements
- It represents text as well as any element/tag inside it

4. textContent

- returns textual content even for hidden elements

Attribute

1. getAttribute("attr")

- to get the attribute value

2. setAttribute("attr",value)

- to set the attribute value

Style

node.style - It helps to style the elements i.e apply css on it

Insert elements

1. node.append(e)

- add at the end of the node (inside)

2. node.prepend(e)

- add at the start of the node (inside)

3. node.before(e)

- add before the node (outside)

4. node.after(e)

- add after the node (outside)

Delete elements

- node.delete(e)

- Used to delete the node