Learning DOM

1. Window object is a way the browser allows us to work with the object. It is the tab we are currently operating. It contains methods and variables that help us interact with the browser. If a variable or function is called upon, if it is not declared, JS looks into the window to find such variable or function
2. The window is for a current tab of a browser while document (which is a property in the window object) refers to the current html document we are working with
3. DOM is called Document Object Model which is a property of the window object refers to all the methods, functions and variables of a particular html file.
4. Selecting element
   1. getElementByID(‘elementID’)

select the element id and manipulate it anyhow you want

* 1. getElementByTagName(‘tagName’)

this returns an array like kind of structure, should be converted to normal JS arrays in order to use real arrays functions. Conversion is like this […returnedElements]

* 1. getELementByClassName(‘className’)

works the same with getElementByTagName

* 1. querySelector(‘cssQuerySelector’) – this selects the first item

pass any css selector such as class, id

querySelectorAll(‘cssQuerySelector’) selects all the elements with the defined selector

1. traversing the DOM
   1. children selection – selects elements without white spaces
   2. childNode – selects elements with white spaces
   3. parentNode – selects the parent of a particular element up the tree
   4. previousSibling
   5. nextSibling
   6. nextElementSibling
   7. previousElementSibling
   8. textContent – returns the string property of an element
   9. getAttribute() returns the attributes of an element (e.g class)
   10. setAttribute()
2. classNames
   1. className returns the name of the class of an element
   2. classList is a list that contains a list of all classes of an element
   3. classList is a list data structure and supports the different methods of a typical list
3. createElement
   1. Firstly, we select the container where we want to place the new element
   2. Secondly, we create an empty element container
   3. Thirdly, we add contents into the new created element
   4. Then we append to the body
   5. insertBefore(element, location)
      1. element denotes the kind of element we want to insert the new element before
      2. location denotes the position we wanna insert this new element
   6. replaceChild(newElement, oldElement)
   7. prepend(element) – adds element in the beginning of a container
   8. remove() – this removes this element from the dom
   9. removeChild(childElement) – removes childElement from this parent node
4. innerHTML: refers to a complete html structure
5. textContent: refers to the textual content of the element
6. **eventHandling:** 
   1. steps in adding eventlistener
      1. select the element to be listened to
      2. add proper event listener
      3. add the call-back function – this defines what happens when the event is triggered
   2. addEventListener(event, callBackFunction)
   3. to change style properties upon events, use classList to add the class containing the style to the element
   4. to use a reference function, always pass the function reference (function) and not invoking the function (function())
   5. event.currentTarget returns the full html element where the eventListener is attached to
   6. preventDefault stops the default callback function attached to an event
   7. to add a particular eventListener to a group of elements, use forEach()
   8. target identifies the element where the event occurred
   9. Selecting elements dynamically
      1. Propagation
         1. Event bubbling
   10. Form Submission events
       1. To prevent default form submission behaviour such as refreshing the page, we use preventDefault() of the event object
7. **Web storage API**