Javascript

•••

Lecture 4

Eng. Sahar E. Hassan

Javascript: Select elements

- **Select element by id**

  ```
  var element = document.getElementById("elementID");
  ```

- **Select elements by class name**
  - It returns an array of elements having that class.

  ```
  var elements = document.getElementsByClassName("className");
  ```

- **Select elements by tag name**
  - It returns an array of elements with that tag.

  ```
  var elements = document.getElementsByTagName("p");
  ```

Javascript: Elements properties

Property	Description
innerHTML	Represents the html markup inside that element
innerText	Represents the text inside that element (discards html elements)
value	gets or sets the value of the element (input,select elements only)
children	Array of the children elements for the element
parentElement	The parent of the selected element in the DOM tree
previousElementSibling	The previous element in the dom tree
nextElementSibling	The next element in the dom tree
id	The id of the element

Javascript: Elements properties

Property	**Description**
name	The name of the element
src	The src attribute of the element (for applicable elements)
style	The style attributes of the element.

Javascript: Elements functions

Property	Description
append()	append value at the end of the element inside the element tags
prepend()	Add value at the start of the element inside the element tags
remove()	Remove the element from the dom tree
createElment()	Create an element in the run time (in memory only)
appendChild()	appends an element inside another element
getAttribute()	Gets an attribute name for an element

Javascript: Create elements

- To create an element you can use the document.createElement() method

Ex:

```
var myElement =  document.createElement("p"); //creates an element (in the memory)

myElement.innerHTML = "Hello World!"; //sets the content of the paragraph element

document.body.appendChild(myElement); //appends the element to the end of the body element
```

Javascript: Style property

- Each element has a ***style*** property. This property is an object that holds other properties for each style

Ex:

```
var myElem = document.getElementById(“myElemID”);

myElem.style.color=“red”; //this will color the text inside the element to red

myElem.style.border = “1px solid black”; //will set the border of the element
```

- If the style property has a dash (e.g: background-color) change it to camelCase notation

```
myElem.style.backgroundColor=“#AAAAAA”; //this will color the background

myElem.style.fontSize=“20px”;
```

Javascript: Events

- Events represent the interaction between the user and the elements on the page
- There are several types of events such as : click, change, focus ,blur,..etc
- Events require handlers in the Javascript code to be executed when the event is triggered.
- There are 2 method to attach events to elements, the first method is to attach the event within the HTML tag as an attribute (inline events)

Example:

```
<div onclick="clickHandler()"></div>
```

Javascript: Events

Event	Description
click	To be executed when the user clicks on the element
change	To be executed when the value of the element changes (specific to elements such as input,select and textarea)
blur	To be executed when the focus is removed from that element (blurred)
focus	To be executed when focus is moved to that element
keydown	To be executed when the user inputs a key on the element (as the user types in and before the key is written)
keyup	To be executed when the use inputs a key on the element (after the key is written)
load	To be executed the document is loaded

Javascript: Events

Event	**Description**
mouseOver	To be executed when the user hovers with the mouse over the element
mouseOut	To be executed when the user moves the mouse away from the element
submit	To be executed when the form is submitted (specific to the <form> element)

Javascript: Inline events notes

- You can also attach the events using code by using "onclick" property of elements

  Ex:

  ```
  myElement.onclick = function(){
          alert(“You clicked the element”);
    }
  ```

- Note that using this method, if you attach the event again, it will overwrite the previous event

  ```
  myElement.onclick = function(){
        alert(“GoodBye”);
  }
  ```

- The above code will alert “GoodBye” and will NOT alert “You clicked the element”

Javascript: Adding events listener

- The second method to add events is by using "addEventListener()" function on the element.
- This method is preferred because it does not involve typing JS code in the HTML and it also supports adding more than 1 event handler for the same event
- Syntax:

```
myElement.addEventListner("click",function(e){
    //code goes here
});
```

Javascript: Adding events listener (continue)

- You can use a function name as a reference to the function instead of using anonymous functions (note that it's without the () brackets)

```
Ex: myElement.addEventListener("click",clickHandler);
function clickHandler(e){
        //code goes here
};
```

- You can use the keyword ***this*** inside the event handler function to access the element who received the event.

```
myElement.addEventListener("click",function(e){
   e.preventDefault(); //to prevent the default behavior of click
});
```

Javascript: Remove events listener

- You can use this method to remove an event handler from an element

Syntax:

myElement.removeEventListener("click",clickHandler);

- In order to be able to remove the event handler, it must be attached using its name as a reference

Javascript: Lab 4

- Exercise 1:

  Task1

  - Refer to the files "exercise1.html" and "exercise1.js"
  - Ask the user to enter a sentence and print it in the element with id="listTitle“
  - Print the employees names in the span elements with class="empName" and the number of employee in the span elements with class="empNumber“

  Task 2

  - Create a function that draws a <ul> and <li> elements
  - The function should accept 1 parameter. The parameter is an array of strings
  - The function should take the array and create an <li> element for each element in the array. Then append the result to the document

Javascript: Lab 4

Exercise 2:

- Refer to the files in exercise2 folder
- Ask the user to enter a name to search for
- Check if the name is found inside the content of any of the <li> elements
- If it’s found write the sentence “found ” inside the element with id=“result” and write also its index
- if it’s not found write “Not found” inside the element with id=“result”

Javascript: Lab 4

- Exercise 3:
  - Refer to the files in exercise3 folder
  - Ask the user to enter the names of students separated by a comma
  - Using **document.createElement()** function, create <ul> element that contains multiple <li> element for each student name. so the result should be like this:

```
<ul>
    <li> name 1</li>
    <li> name2 </li>
    … and so on
</ul>
```

  - Append the result to the element with id dynamicList

Javascript: Lab 4

- Exercise 4:
  - Using prompt, ask the user to enter a list of colors separated by a comma (ex: red,green,yellow)
  - For each color, create a div and do the following for each div:
    - set its the background color to the that color
    - set its html to “Hello Javascript”
  - Then append the result divs to the element with id=“result”

Bonus: create another div then

set its color to a random color from the entered colors

and set its content to Random div

Then append it to the element with id=“result”

Javascript: Lab 4

- Exercise 5 :

  - Refer to the files in exercise5 folder

  - When the user clicks “Turn on” button, it should do the following:

1- hide the “off.png” image

2- show the “on.png” image

3-change the text on the button to “Turn off”

  - When the user clicks “Turn off” button, it should do the following:

1- show the “off.png” image

2- hide the “on.png” image

3-change the text on the button to “Turn on”

Javascript: Lab 4

- Exercise 6
- Refer to the files in exercise 6 folder
- The user can enter his data in the “userName” and “Message” fields. Then when he clicks “save”, it should :

  1- append a row in the table that contains 3 cells: Name, Message, delete button.

  2- erase the data in the “userName” and “message” text fields
- When the user clicks the delete button, it should delete the entire row
- Bonus: In the last cell (Actions), add also another button named edit, when clicked it should do the following:

  1- display the cell data (i.e. Name and Message)again in the “userName” and “Message” text fields

  2- remove the current row