

**Lecture 2.3**  
**Introduction to DOM,**  
**Document, Window, History, Navigator, and Screen Objects**

**Course Instructor**  
**Engr. Madeha Mushtaq**

# Document Object Model (DOM)

- Once html element are rendered in the browser window, browser can not recognize them.
- To create interactive web pages it is vital that the browser continues to recognize HTML elements.
- JavaScript enabled browsers to do this because it recognizes and uses DOM.

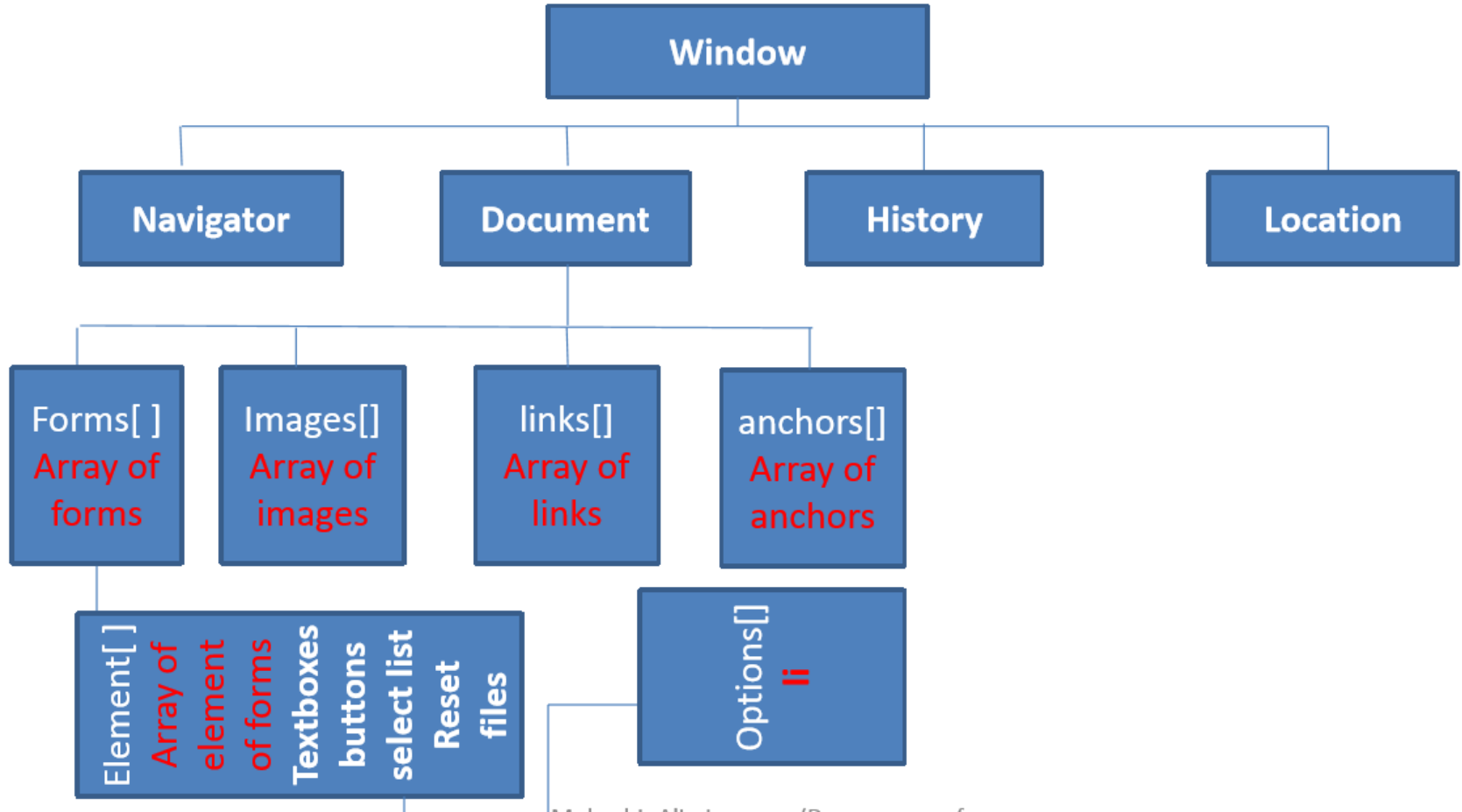
# Document Object Model (DOM)

- All HTML elements, along with their text and attributes, can be accessed through the DOM
  - The contents can be modified or deleted, and new elements can be created.

# Document Object Model (DOM)

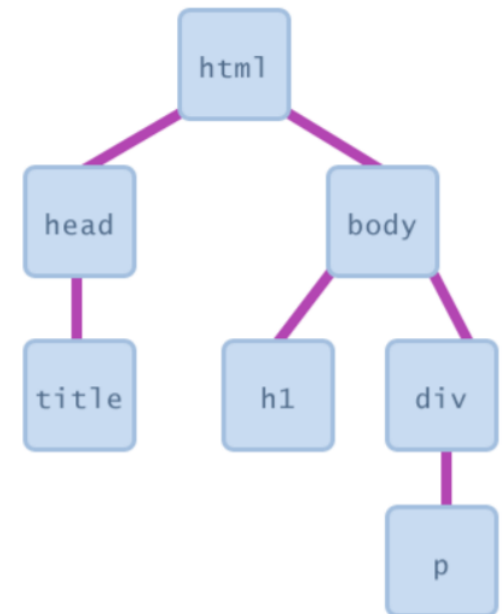
- The HTML DOM is platform and language Independent.
  - It can be used by any programming language like Java, JavaScript, and VBScript.
- The HTML DOM can be thought of as a hierarchy moving from the most general object to the most specific.

# Document Object Model (DOM)



# Document Object

- Document object represent the whole HTML document.
- Every HTML element in the document is an object.
- JS code can talk to these objects to examine elements' state
  - e.g. see whether a box is checked
- we can change state
  - e.g. insert some new text into a div
- we can change styles
  - e.g. make a paragraph red



# Document Object

- `document.forms[0].elements[0].value`
- `document.images[0].src`
- `document.links[0].href`

# Retrieving HTML elements

- The `getElementById()` method is the most used method of the DOM.
- To retrieve an element, it must have a unique id
- It retrieves a specified element of the HTML document and returns a reference to it.
  - `document.getElementById("element-id")`
- Other methods are `getElementsByClassName`, `getElementsByTagName`.

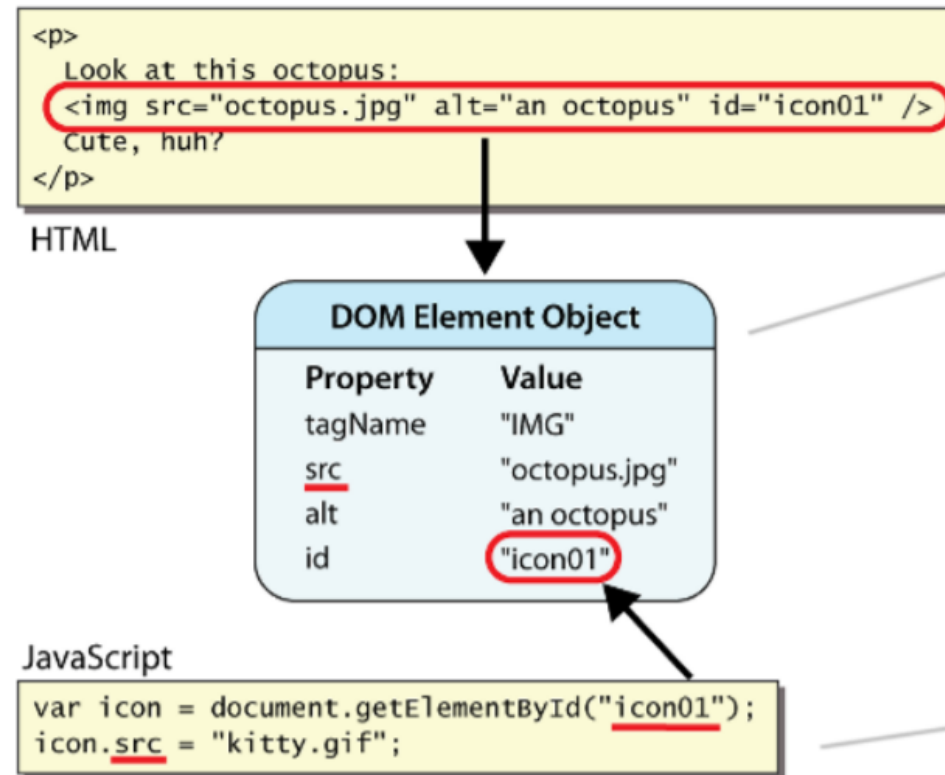


# Retrieving HTML elements

- The returned reference can be used to retrieve element attributes
  - `document.getElementById("elementid").attribute`
  - `document.getElementById("pic").src`
  - `document.getElementById("pic").height`

# Retrieving HTML elements

- Access/modify the attributes of a DOM object with object Name.attribute Name
- Most DOM object attributes have the same names as the corresponding HTML attribute.
- `img` tag's `src` property
- a tag's `href` property



# Retrieving HTML elements

```
<html>
<head>
  <title>Retrieving Values </title>
  <script type="text/javascript">
    function notEmpty() {
      var myTextField = document.getElementById('myText');
      if(myTextField.value != "")
        alert("You entered: " + myTextField.value)
      else
        alert("Would you please enter some text?")
    }
  </script>
</head>
<body>
  <input type='text' id='myText' />
  <input type='button' onclick='notEmpty()' value='Form Checker' />
</body>
</html>
```

# Retrieving the text of an element

- `innerHTML` property defines both the HTML code and the text that occurs between that element's opening and closing tags.
- `document.getElementById("element-id").innerHTML`

# Retrieving the text of an element

```
<html>
<head>
  <title>Retrieving text</title>
  <script type="text/javascript">
    function getText () {
      var a = document.getElementById("mypara");
      alert(a.innerHTML)
      a.innerHTML = "Hello World Again";
      alert(a.innerHTML)
      .....
    }
  </script>
</head>
<body>
  <p id = "mypara"> Hello world </p>
  <script>
    getText ();
  </script>
</body>
</html>
```

# Getting value of attributes

- `getAttribute()` method is used to retrieve values of attributes.
- `document.getElementById("elementid").getAttribute("attribute-name")`
- `document.getElementById("pic").getAttribute("src")`

# Getting value of attributes

```
<html>
<head>
  <title>Retrieving Values </title>
  <a href="https://www.google.com/" target="_blank" id="myLink">Google</a>
</head>

<body>
<script>
  var link = document.getElementById("myLink");

  // Getting the attributes values
  var href = link.getAttribute("href");
  alert(href); // Outputs: https://www.google.com/

  var target = link.getAttribute("target");
  alert(target); // Outputs: _blank
</script>
</script>
</body>
</html>
```

# Setting value of attributes

- `setAttribute()` method is used to set values of attributes
- `document.getElementById("elementid").setAttribute("attribute-name", "Value")`
- `document.getElementById("pic").setAttribute("src", "abc.jpg")`



# Setting value of attributes

```
SetAttribute_Link_4.html x
1 <html>
2 <head>
3   <title>Set Attributes </title>
4   <script>
5     function setF()
6     {
7       var link = document.getElementById("myLink");
8       var google = link.getAttribute("href");
9       alert(google);
10      link.setAttribute("href", "http://www.uetpeshawar.edu.pk/");
11      var uet = link.getAttribute("href");
12      alert(uet);
13    }
14  </script>
15 </head>
16
17 <body>
18   <a href="https://www.google.com/" target="_blank" id="myLink">Link</a>
19   <script>
20     setF();
21   </script>
22 </body>
23 </html>
```

# The Window object

- The window object represents the **browser window** in which the script is running.
- Window object Properties:
  - **window.innerHeight** - the inner height of the browser window (in pixels)
  - **window.innerWidth** - the inner width of the browser window (in pixels)
- Window object Methods:
  - window.open()
  - window.close()
  - moveTo()
  - resizeTo()

# The Window Object

```
<HTML>
<HEAD>
  <TITLE>Open a New Window</TITLE>
  <SCRIPT LANGUAGE="JavaScript">
    function OpenWindow()
    { var NewWindow = window.open("image.jpg", "NewWindow","width=350,height=400")
    }
  </SCRIPT>
</HEAD>
<BODY BGCOLOR=White>
  <SCRIPT>
    OpenWindow();
  </SCRIPT>
</BODY>
</HTML>
```

# The Window Object

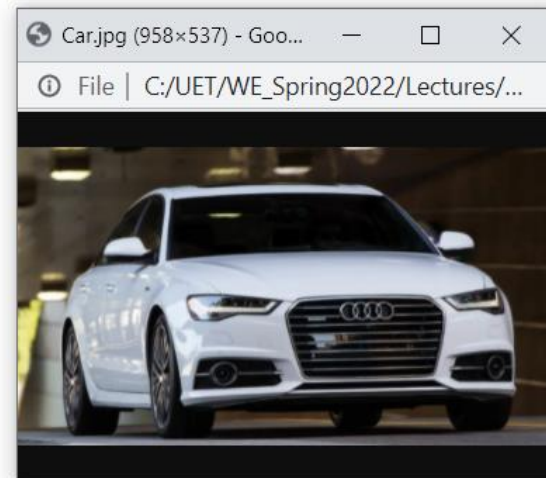
```
<HTML>
<HEAD>
<TITLE>Close Windows</TITLE>
<SCRIPT LANGUAGE="JavaScript">
var childwindow = null
function opennewwindow()
{
  childwindow = window.open("Car.jpg","childwindow", "width=300,height=200,toolbar=yes,scrollbars=yes,")
}
function closenewwindow()
{
  if (childwindow && !childwindow.closed)
  {
    childwindow.close()
  }
}
</SCRIPT>
</HEAD>
<BODY BGCOLOR=White>
<CENTER>
<H1> <A HREF="javascript:opennewwindow()">View Picture</A>
<BR>
<A HREF="javascript:closenewwindow()">Close Picture</A>
</H1>
</CENTER>
</BODY>
</HTML>
```

# The Window Object

- Output:

File | C:/UET/WE\_Spring2022/Lectures/Lecture%202/lect-2.3/CloseWindow\_6.html

[View Picture](#)  
[Close Picture](#)



# The History object

- The history object contains the URLs visited by the user (within a browser window)
- The history object is part of the window object and is accessed through the window.history property.
- Used to move forward and backward through the visitor's browsing history.

# The History object

- History object properties:
  - **Length**: Returns the number of URLs in the history list
- History object methods:
  - **back()**: Loads the previous URL in the history list
  - **forward()**: Loads the next URL in the history
  - **listgo()**: Loads a specific URL from the history list

# The History object

```
HistoryObject_7.html x
1 <html>
2 <head>
3   <title>History Object</title>
4   <script language = "javascript">
5     document.write(history.length)
6     function goback()
7     {
8       window.history.back()
9     }
10    function goforward()
11    {
12      window.history.forward()
13    }
14  </script>
15 </head>
16 <body>
17   <h1> This is the first page </h1>
18   <a href = "RetrieveHTMLElements_1.html"> Go to the next page</a> <br>
19   <input type = "button" value = "Go Back!" onclick = "javascript:goback()">
20   <input type = "button" value = "Go forward!" onclick = "javascript:goforward()">
21 </body>
22 </html>
```



# The Navigator object

- The navigator object contains information about the browser.
- Provides several properties that assist in the detection of various elements of the visitor's browser and environment.

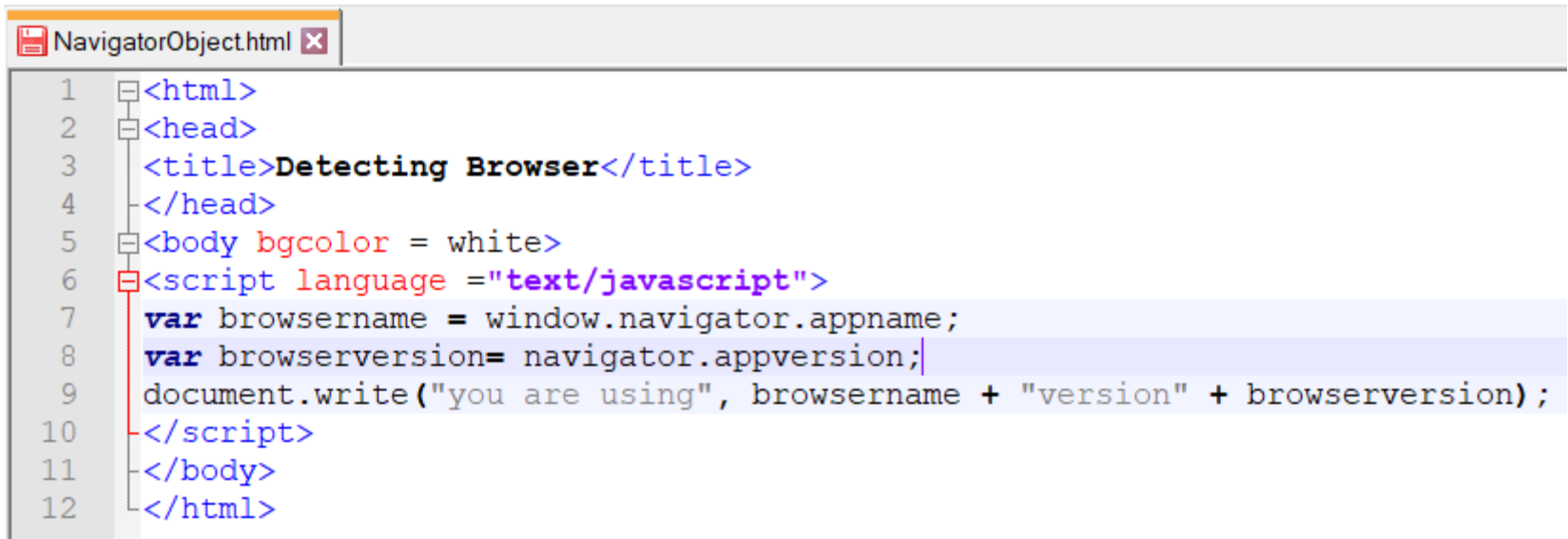
# The Navigator object

- Navigator object properties:
  - `appName`: Returns the code name of the browser
  - `appVersion`: Returns the name of the browser
  - `appVersion`: Returns the version information of the browser
- Navigator object methods:
  - `javaEnabled()`: Specifies whether or not the browser has Java enabled

# Detecting Users browser

- Used to write browser specific code.
- Can also be used to restrict users to use a specific browser.

# Detecting Users browser



```
1 <html>
2 <head>
3   <title>Detecting Browser</title>
4 </head>
5 <body bgcolor = white>
6 <script language = "text/javascript">
7   var browsername = window.navigator.appname;
8   var browserversion= navigator.appversion;
9   document.write("you are using", browsername + "version" + browserversion);
10 </script>
11 </body>
12 </html>
```

# The Screen object

- The screen object contains information about the visitor's screen.
- You might need this information to determine which images to display or how large the page can be.

# The Screen object

- The screen object properties:
- **availHeight**: Returns the height of the screen (excluding the Windows Taskbar)
- **availWidth**: Returns the width of the screen (excluding the Windows Taskbar)
- **colorDepth**: Returns the bit depth of the color palette for displaying images.

# The Screen object

- **height:** Returns the total height of the screen
- **width:** Returns the total width of the screen

# The Screen object

```
<html>
<head>
  <title>JavaScript Screen Object</title>
</head>
<body>

  <h1>JavaScript Screen Object Example</h1>
  <script type="text/javascript">
    document.write("<b>Total Height = </b>" + screen.height + "<br/>");
    document.write("<b>Total Width = </b>" + screen.width + "<br/>");
    document.write("<b>Available Width = </b>" + screen.availWidth + "<br/>");
    document.write("<b>Available Height = </b>" + screen.availHeight + "<br/>");
    document.write("<b>Screen Color Depth = </b>" + screen.colorDepth + "<br/>");
    document.write("<b>Screen Pixel Depth = </b>" + screen.pixelDepth + "<br/>");
  </script>

</body>
</html>
```



# References

- **Chapter 11** Beginning HTML, XHTML, CSS, and JavaScript, by Jon Duckett, Wiley Publishing; 2009, ISBN: 978-0-470-54070-1.
- **Chapter 3,6,11** Learn JavaScript, by Chuck Easttom, Wordware Publishing; 2002, ISBN 1-55622-856-2