ECAP472

Web Technologies

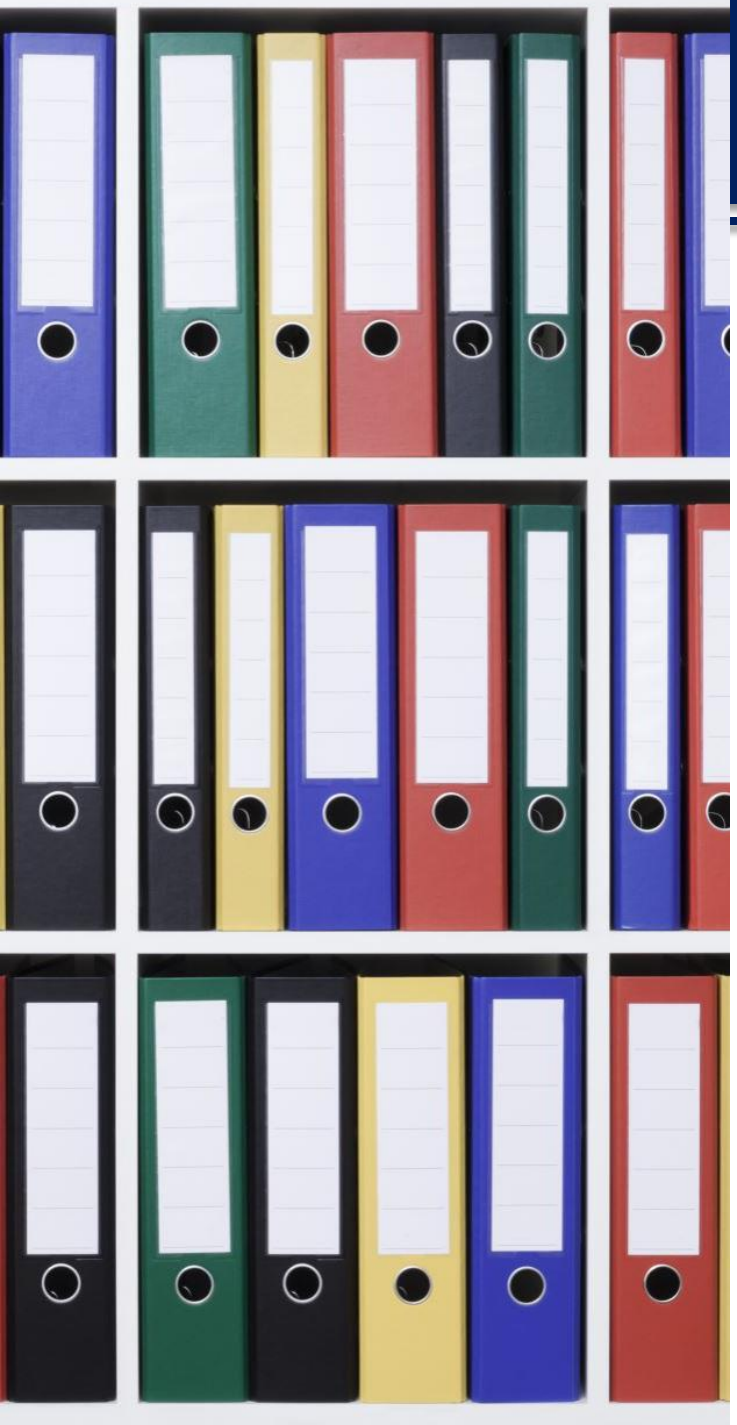
Dr. Pritpal Singh

Associate Professor

Learning Outcomes

After this lecture, you will be able to

- understand concept of document object model .



Document Object Model

“ The W3C Document Object Model (DOM) is a platform and language-neutral interface that allows programs and scripts to dynamically access and update the content, structure, and style of a document. ”

- *W3C*

Document Object Model

- The document object represents the whole html document.
- When html document is loaded in the browser, it becomes a document object. It is the root element that represents the html document. It has properties and methods. By the help of document object, we can add dynamic content to our web page.

Why DOM is Required?

- HTML is used to structure the web pages and Javascript is used to add behavior to our web pages. When an HTML file is loaded into the browser, the javascript can not understand the HTML document directly.
- So, a corresponding document is created (DOM). DOM is basically the representation of the same HTML document but in a different format with the use of objects.

Why DOM is Required?

Javascript interprets DOM easily i.e javascript can not understand the tags(<h1>H</h1>) in HTML document but can understand object h1 in DOM. Now, Javascript can access each of the objects (h1, p, etc) by using different functions.

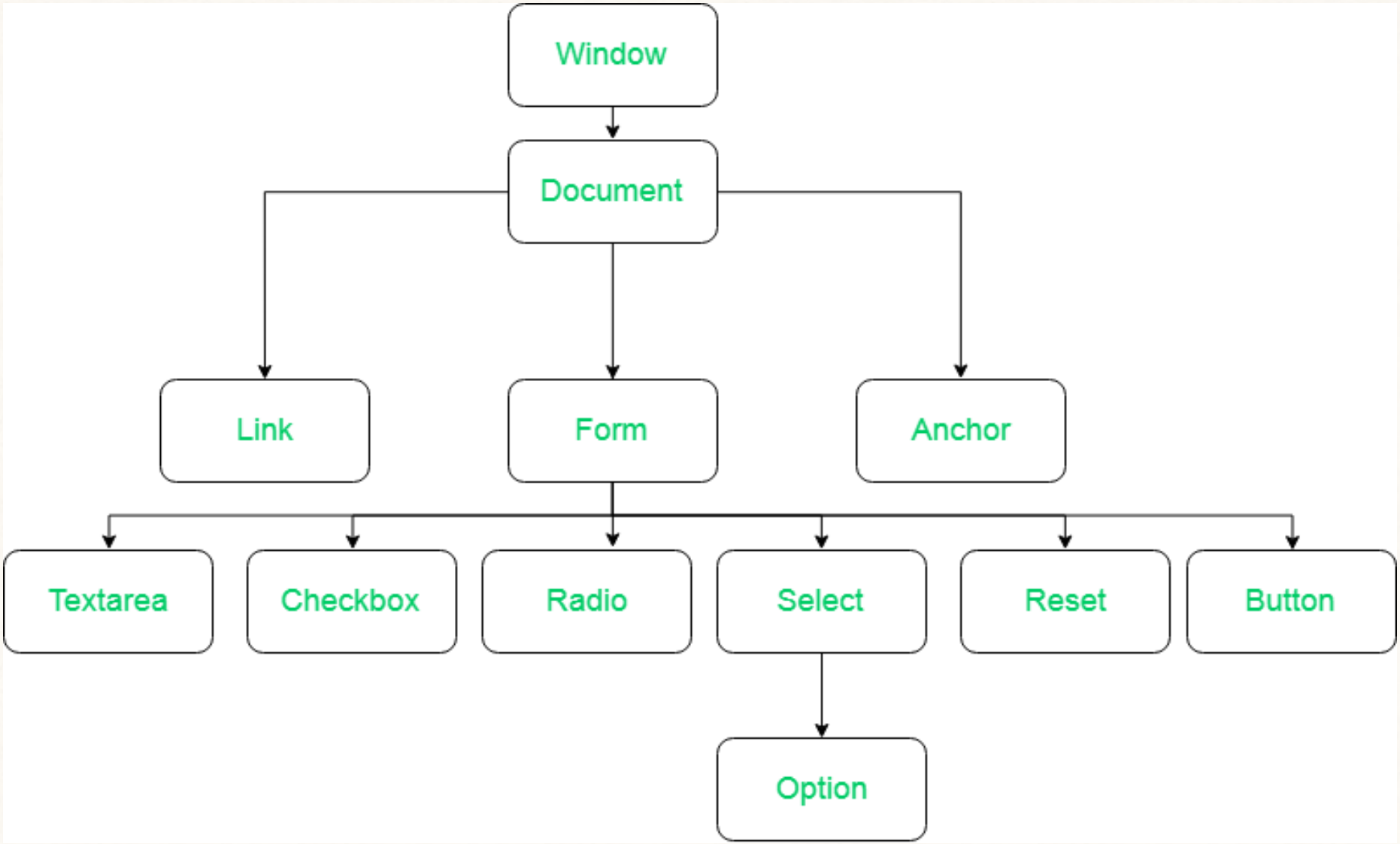
Why called an Object Model?

Documents are modeled using objects and the model includes not only the structure of a document but also the behavior of a document and the objects of which it is composed of like tag elements with attributes in HTML.

Properties of DOM

Let’s see the properties of the document object that can be accessed and modified by the document object.

Properties of DOM



Properties of DOM

- Window Object: Window Object is always at top of the hierarchy.
- Document Object: When an HTML document is loaded into a window, it becomes a document object.
- Form Object: It is represented by form tags.
- Link Object: It is represented by link tags.

Properties of DOM

- Anchor Object: It is represented by a HREF tags.
- Form Control Elements: Form can have many control elements such as text fields, buttons, radio buttons, and checkboxes, etc.

Methods of document object:

We can access and change the contents of document by its methods.

Method	Description
write("string")	writes the given string on the doucment.
writeln("string")	writes the given string on the doucment with newline character at the end.
getElementById()	returns the element having the given id value.
getElementsByName()	returns all the elements having the given name value.
getElementsByTagName()	returns all the elements having the given tag name.
getElementsByClassName()	returns all the elements having the given class name.

With the object model, JavaScript gets all the power it needs to create dynamic HTML

- JavaScript can change all the HTML elements in the page
- JavaScript can change all the HTML attributes in the page
- JavaScript can change all the CSS styles in the page
- JavaScript can remove existing HTML elements and attributes

With the object model, JavaScript gets all the power it needs to create dynamic HTML

- JavaScript can add new HTML elements and attributes
- JavaScript can react to all existing HTML events in the page
- JavaScript can create new HTML events in the page

JavaScript HTML DOM Events

HTML DOM allows JavaScript to react to HTML events:

- A JavaScript can be executed when an event occurs, like when a user clicks on an HTML element.
- To execute code when a user clicks on an element, add JavaScript code to an HTML event attribute

onclick=JavaScript

JavaScript HTML DOM Events

HTML DOM allows JavaScript to react to HTML events:

- A JavaScript can be executed when an event occurs, like when a user clicks on an HTML element.
- To execute code when a user clicks on an element, add JavaScript code to an HTML event attribute

onclick=JavaScript

JavaScript HTML DOM Events

HTML DOM allows JavaScript to react to HTML events:

- A JavaScript can be executed when an event occurs, like when a user clicks on an HTML element.
- To execute code when a user clicks on an element, add JavaScript code to an HTML event attribute

onclick=JavaScript

Example of HTML Events

- When a user clicks the mouse
- When a web page has loaded
- When an image has been loaded
- When the mouse moves over an element
- When an input field is changed
- When an HTML form is submitted
- When a user strokes a key

Example of HTML Events

- When a user clicks the mouse
- When a web page has loaded
- When an image has been loaded
- When the mouse moves over an element
- When an input field is changed
- When an HTML form is submitted
- When a user strokes a key

Example of HTML Events

- When a user clicks the mouse
- When a web page has loaded
- When an image has been loaded
- When the mouse moves over an element
- When an input field is changed
- When an HTML form is submitted
- When a user strokes a key

Example of HTML Events

- When a user clicks the mouse
- When a web page has loaded
- When an image has been loaded
- When the mouse moves over an element
- When an input field is changed
- When an HTML form is submitted
- When a user strokes a key

Example of HTML Events

- When a user clicks the mouse
- When a web page has loaded
- When an image has been loaded
- When the mouse moves over an element
- When an input field is changed
- When an HTML form is submitted
- When a user strokes a key

Example of HTML Events

- When a user clicks the mouse
- When a web page has loaded
- When an image has been loaded
- When the mouse moves over an element
- When an input field is changed
- When an HTML form is submitted
- When a user strokes a key

Example of HTML Events

- When a user clicks the mouse
- When a web page has loaded
- When an image has been loaded
- When the mouse moves over an element
- When an input field is changed
- When an HTML form is submitted
- When a user strokes a key

JavaScript HTML DOM - Changing CSS

- The HTML DOM allows JavaScript to change the style of HTML elements.
- To change the style of an HTML element, use this syntax:

```
document.getElementById(id).style.property =
                                    new style
```

Example:

The following example changes the style of a <p> element:

```
<html>
<body>
<p id="p2">Hello World!</p>
<script>
document.getElementById("p2").style.color =
"blue";
</script>
</body>
</html>
```

Using Events

- The HTML DOM allows you to execute code when an event occurs.
- Events are generated by the browser when "things happen" to HTML elements:
- An element is clicked on
- The page has loaded
- Input fields are changed

Using Events

- The HTML DOM allows you to execute code when an event occurs.
- Events are generated by the browser when "things happen" to HTML elements:
- An element is clicked on
- The page has loaded
- Input fields are changed

Using Events

- The HTML DOM allows you to execute code when an event occurs.
- Events are generated by the browser when "things happen" to HTML elements:
- An element is clicked on
- The page has loaded
- Input fields are changed

Using Events

- The HTML DOM allows you to execute code when an event occurs.
- Events are generated by the browser when "things happen" to HTML elements:
- An element is clicked on
- The page has loaded
- Input fields are changed

Using Events

- The HTML DOM allows you to execute code when an event occurs.
- Events are generated by the browser when "things happen" to HTML elements:
- An element is clicked on
- The page has loaded
- Input fields are changed

JavaScript Window – The Browser Object Mode

- The Browser Object Model (BOM) allows JavaScript to "talk to" the browser.
- There are no official standards for the Browser Object Model (BOM).
- Since modern browsers have implemented (almost) the same methods and properties for JavaScript interactivity, it is often referred to, as methods and properties of the BOM.

JavaScript Window – The Browser Object Mode

- The Browser Object Model (BOM) allows JavaScript to "talk to" the browser.
- There are no official standards for the Browser Object Model (BOM).
- Since modern browsers have implemented (almost) the same methods and properties for JavaScript interactivity, it is often referred to, as methods and properties of the BOM.

JavaScript Window – The Browser Object Mode

- The Browser Object Model (BOM) allows JavaScript to "talk to" the browser.
- There are no official standards for the Browser Object Model (BOM).
- Since modern browsers have implemented (almost) the same methods and properties for JavaScript interactivity, it is often referred to, as methods and properties of the BOM.

The Window Object

- The window object is supported by all browsers. It represents the browser's window.
- All global JavaScript objects, functions, and variables automatically become members of the window object.
- Global variables are properties of the window object.
- Global functions are methods of the window object.

The Window Object

- The window object is supported by all browsers. It represents the browser's window.
- All global JavaScript objects, functions, and variables automatically become members of the window object.
- Global variables are properties of the window object.
- Global functions are methods of the window object.

The Window Object

- The window object is supported by all browsers. It represents the browser's window.
- All global JavaScript objects, functions, and variables automatically become members of the window object.
- Global variables are properties of the window object.
- Global functions are methods of the window object.

The Window Object

- The window object is supported by all browsers. It represents the browser's window.
- All global JavaScript objects, functions, and variables automatically become members of the window object.
- Global variables are properties of the window object.
- Global functions are methods of the window object.

Changing the Value of an Attribute

- To change the value of an HTML attribute, use this syntax:

```
document.getElementById(id).attribute = new value
```

Example

Changes the value of the src attribute of an <img> element

```
<!DOCTYPE html>
<html>
<body>
<img id="myImage" src="smiley.gif">
<script>
document.getElementById("myImage").src = "la
ndscape.jpg";
</script>
</body>
</html>
```

Example Explained

- The HTML document above contains an <img> element with id="myImage"
- We use the HTML DOM to get the element with id="myImage"
- A JavaScript changes the src attribute of that element from "smiley.gif" to "landscape.jpg"

Example Explained

- The HTML document above contains an <img> element with id="myImage"
- We use the HTML DOM to get the element with id="myImage"
- A JavaScript changes the src attribute of that element from "smiley.gif" to "landscape.jpg"

Example Explained

- The HTML document above contains an <img> element with id="myImage"
- We use the HTML DOM to get the element with id="myImage"
- A JavaScript changes the src attribute of that element from "smiley.gif" to "landscape.jpg"

Dynamic HTML content

JavaScript can create dynamic HTML content

```
<!DOCTYPE html>
<html>
<body>
<script>
document.getElementById("demo").innerHTML =
"Date : " + Date(); </script>
</body>
</html>
```

JavaScript Forms

- JavaScript Form Validation
- HTML form validation can be done by JavaScript.
- If a form field (fname) is empty, this function alerts a message, and returns false, to prevent the form from being submitted

JavaScript Forms

- JavaScript Form Validation
- HTML form validation can be done by JavaScript.
- If a form field (fname) is empty, this function alerts a message, and returns false, to prevent the form from being submitted

JavaScript Forms

- JavaScript Form Validation
- HTML form validation can be done by JavaScript.
- If a form field (fname) is empty, this function alerts a message, and returns false, to prevent the form from being submitted

JavaScript Forms

- JavaScript Form Validation
- HTML form validation can be done by JavaScript.
- If a form field (fname) is empty, this function alerts a message, and returns false, to prevent the form from being submitted

JavaScript Can Validate Numeric Input

JavaScript is often used to validate numeric input:

Automatic HTML Form Validation

- HTML form validation can be performed automatically by the browser:
- If a form field (fname) is empty, the required attribute prevents this form from being submitted

Automatic HTML Form Validation

- HTML form validation can be performed automatically by the browser:
- If a form field (fname) is empty, the required attribute prevents this form from being submitted

Data Validation

- Data validation is the process of ensuring that user input is clean, correct, and useful.

Typical validation tasks are:

- ✔ has the user filled in all required fields?
- ✔ has the user entered a valid date?
- ✔ has the user entered text in a numeric field?

Data Validation

- Data validation is the process of ensuring that user input is clean, correct, and useful.

Typical validation tasks are:

- ✔ has the user filled in all required fields?
- ✔ has the user entered a valid date?
- ✔ has the user entered text in a numeric field?

Data Validation

- Data validation is the process of ensuring that user input is clean, correct, and useful.

Typical validation tasks are:

- ✔ has the user filled in all required fields?
- ✔ has the user entered a valid date?
- ✔ has the user entered text in a numeric field?

Data Validation

- Data validation is the process of ensuring that user input is clean, correct, and useful.

Typical validation tasks are:

- ✔ has the user filled in all required fields?
- ✔ has the user entered a valid date?
- ✔ has the user entered text in a numeric field?

Data Validation

- Data validation is the process of ensuring that user input is clean, correct, and useful.

Typical validation tasks are:

- ✔ has the user filled in all required fields?
- ✔ has the user entered a valid date?
- ✔ has the user entered text in a numeric field?

Data Validation

- Most often, the purpose of data validation is to ensure correct user input.
- Validation can be defined by many different methods, and deployed in many different ways.
- Server side validation is performed by a web server, after input has been sent to the server.
- Client side validation is performed by a web browser, before input is sent to a web server.

Data Validation

- Most often, the purpose of data validation is to ensure correct user input.
- Validation can be defined by many different methods, and deployed in many different ways.
- Server side validation is performed by a web server, after input has been sent to the server.
- Client side validation is performed by a web browser, before input is sent to a web server.

Data Validation

- Most often, the purpose of data validation is to ensure correct user input.
- Validation can be defined by many different methods, and deployed in many different ways.
- Server side validation is performed by a web server, after input has been sent to the server.
- Client side validation is performed by a web browser, before input is sent to a web server.

Data Validation

- Most often, the purpose of data validation is to ensure correct user input.
- Validation can be defined by many different methods, and deployed in many different ways.
- Server side validation is performed by a web server, after input has been sent to the server.
- Client side validation is performed by a web browser, before input is sent to a web server.

HTML Constraint Validation

- HTML5 introduced a new HTML validation concept called constraint validation.
- HTML constraint validation is based on:
- Constraint validation HTML Input Attributes
- Constraint validation CSS Pseudo Selectors
- Constraint validation DOM Properties and Methods

HTML Constraint Validation

- HTML5 introduced a new HTML validation concept called constraint validation.
- HTML constraint validation is based on:
- Constraint validation HTML Input Attributes
- Constraint validation CSS Pseudo Selectors
- Constraint validation DOM Properties and Methods

HTML Constraint Validation

- HTML5 introduced a new HTML validation concept called constraint validation.
- HTML constraint validation is based on:
- Constraint validation HTML Input Attributes
- Constraint validation CSS Pseudo Selectors
- Constraint validation DOM Properties and Methods

HTML Constraint Validation

- HTML5 introduced a new HTML validation concept called constraint validation.
- HTML constraint validation is based on:
- Constraint validation HTML Input Attributes
- Constraint validation CSS Pseudo Selectors
- Constraint validation DOM Properties and Methods

HTML Constraint Validation

- HTML5 introduced a new HTML validation concept called constraint validation.
- HTML constraint validation is based on:
- Constraint validation HTML Input Attributes
- Constraint validation CSS Pseudo Selectors
- Constraint validation DOM Properties and Methods

Contraint Validation HTML Input Attribues

Attribute	Description
disabled	Specifies that the input element should be disabled
max	Specifies the maximum value of an input element
min	Specifies the minimum value of an input element
pattern	Specifies the value pattern of an input element
required	Specifies that the input field requires an element
type	Specifies the type of an input element

That’s all for now…