

Chapter 6

How to script the DOM with JavaScript

Objectives

- DOM scripting properties and methods
- The FAQ application
- How to script forms and controls
- The Register application
- How to add and remove nodes from the DOM
- The Register application with a table



DOM scripting properties and methods



DOM scripting concepts

- The ***DOM(Document Object Model)*** is a hierarchical collection of nodes in the web browser's memory that represents the current web page.
- The DOM of a web page is built as the page is loaded by web browser.
- JavaScript can modify the web page in the browser by modifying the DOM.
- To modify DOM you can use properties and methods of it.

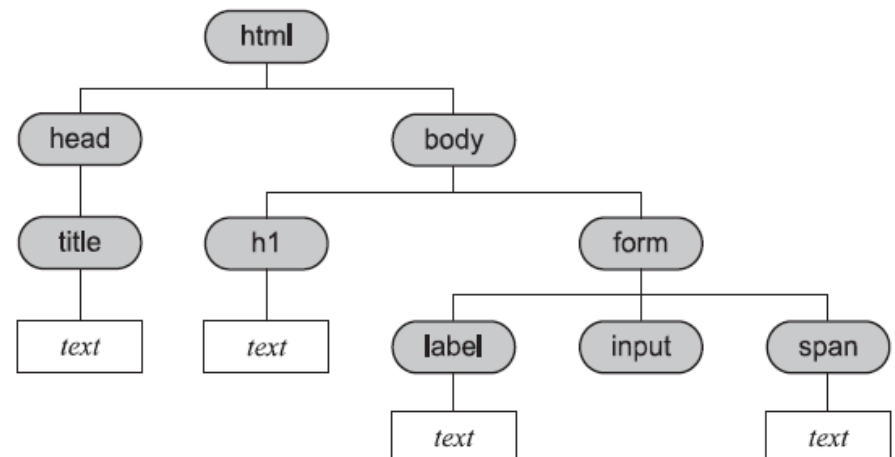


DOM scripting concepts (Cont.)

The code for a web page

```
<!DOCTYPE html>
<html>
<head>
  <title>Join Email List</title>
</head>
<body>
  <h1>Please join our email list</h1>
  <form id="email_form" name="email_form" action="join.html" method="get">
    <label for="email_address">Email Address:</label>
    <input type="text" id="email_address">
    <span id="email_error">*</span><br>
  </form>
</body>
</html>
```

The DOM for the web page



DOM scripting concepts (Cont.)

The DOM nodes that you commonly use

Type	Description
Document	Root node of the DOM. It can only have one Element node as a child node.
Element	An element in the web page. It can have Element, Text, and Comment nodes as child nodes.
Attr	An attribute of an element. Although it is attached to an Element node, it isn't considered a child node. It can have a Text node as a child node.
Text	The text for an element or attribute. It can't have a child node.



The properties of the Node interface

Some of the properties of the Node interface

Property	Description
nodeValue	For a Text, Comment, or Attribute node, this property returns the text that's stored in the node. Otherwise, it returns a null value.
parentNode	Returns the parent node of the current node if one exists. Otherwise, this property returns a null value.
childNodes	Returns an array of Node objects representing the child nodes of the current node.
firstChild	Returns a Node object for the first child node. If this node doesn't have child nodes, this property returns a null value.
lastChild	Returns a Node object for the last child node. If this node doesn't have child nodes, this property returns a null value.
nextElementSibling	Returns a Node object for the next sibling. If this node doesn't have a sibling element that follows it, this property returns a null value.



The properties of the Node interface(cont.)

HTML that contains element and text nodes

```
<body>
  <h1>Please join our email list</h1>
  <form id="email_form" name="email_form" action="join.html" method="get">
    <label for="email_address">Email Address:</label>
    <input type="text" id="email_address">
    <span id="email_error">*</span><br>
    <label>&nbsp;</label>
    <input type="button" id="join_list" value="Join our List">
  </form>
</body>
```

How to get the text of an HTML element with “email_error” as its id

```
var errorText = $("email_error").firstChild.nodeValue;
```

How to set the text of an HTML element with “email_error” as its id

```
$("email_error").firstChild.nodeValue = "Entry is invalid.";
```

How to set the text for the span tag to an empty string without using its id

```
$("email_address").nextElementSibling.firstChild.nodeValue = "";
```



The methods of Document and Element interfaces

- The methods of the Document and Element interfaces let you get array of elements

Common methods of the Document and Element interfaces

Method	Description
<code>getElementsByTagName (tagName)</code>	Returns an array of all Element objects descended from the document or element that have a tag that matches the specified tag.
<code>getElementsByName (name)</code>	Returns an array of all Element objects descended from the document or element that have a name attribute that matches the specified name.
<code>getElementsByClassName (classNames)</code>	Returns an array of all Element objects descended from the document or element that have a class attribute with a name or names that match the parameter. The <code>classNames</code> parameter can be a single name or a space-separated list of class names.



The methods of Document and Element interfaces(cont.)

- The methods of the Element interfaces also let you work with attributes

Common methods of the Element interface

Method	Description
hasAttribute(<i>name</i>)	Returns true if the Element has the attribute specified in name.
getAttribute(<i>name</i>)	Returns the value of the attribute specified in name or an empty string if an attribute of that name isn't set.
setAttribute(<i>name</i>, <i>value</i>)	Sets the attribute specified in name to the specified value. If the attribute doesn't already exist, it creates the attribute too.
removeAttribute(<i>name</i>)	Removes the attribute specified in name.



The methods of Document and Element interfaces(cont.)

How to create an array of all <a> tags in a document

```
var links = document.getElementsByTagName("a");
```

How to create an array of all li tags within a ul element (image_list)

```
var list = document.getElementById("image_list");  
var items = list.getElementsByTagName("li");
```

How to test for and get an attribute

```
var list = document.getElementById("image_list");  
if ( list.hasAttribute("class") ) {  
    var classAttribute = list.getAttribute("class");  
}
```

How to set an attribute

```
var image = document.getElementById("div");  
image.setAttribute("class", "open");
```

How to remove an attribute

```
var list = document.getElementById("image_list");  
list.removeAttribute("class");
```



The properties of the DOM HTML specification

- The HTML Specification provides shortcuts that make it easier to work with DOM nodes.
- Typical properties available with the DOM HTML specification

Element	Property	Attribute
all	id	The id attribute
	title	The title attribute
	className	The class attribute
	tagName	The name of tag like div, h1, h2 ..
<a>	href	The href attribute
Img	src	The src attribute
	alt	The alt attribute
input	disable	The disable attribute

How to code use DOM HTML specification

- How to get/set img tag src attribute

```
var imageElement = $("iamge");  
alert(imageElement.getAttribute(src));  
iamgeElement.setAttribute(src, "lures.jpg")
```

- Or you can code

```
alert(imageElement. Src);  
iamgeElement.src = "lures.jpg";
```

- How to get id attribute of the first element in an array

```
links = document.getElementsByTagName("a");  
var firstLinkId = links[0].id;
```



How to code use DOM HTML specification(cont.)

- How to get href attribute of an <a> element
`var target = $("first_link").href;`
- How to set/get attribute of an element with two class names
`$("div").className = "Open plus";`
`var classNames = $("div").className;`
- How to disable and enable an element
`$("bntPlay").disable = true;`
`$("bntPlay").disable = false;`



The FAQs application

JavaScript FAQs

- + [What is JavaScript?](#)
- + What is jQuery?
- + Why is jQuery becoming so popular?



The HTML for FAQs application

```
<body>
  <main id="faqs">
    <h1>JavaScript FAQs</h1>
    <h2><a href="#" >What is JavaScript?</a></h2>
    <div>
      <p>JavaScript is a is a browser-based programming language
        that makes web pages more responsive and saves round trips to the server.
      </p>
    </div>
    <h2><a href="#">What is jQuery?</a></h2>
    <div>
      <p>jQuery is a library of the JavaScript functions that you're most likely
        to need as you develop websites.
      </p>
    </div>
    <h2><a href="#">Why is jQuery becoming so popular?</a></h2>
    <div>
      <p>Three reasons:</p>
      <ul>
        <li>It's free.</li>
        <li>It lets you get more done in less time.</li>
        <li>All of its functions are cross-browser compatible.</li>
      </ul>
    </div>
  </main>
</body>
```



The JavaScript for FAQs application

```
"use strict";
var $ = function(id) { return document.getElementById(id); };

// the event handler for the click event of each h2 element
var toggle = function() {
    var h2 = this;           // clicked h2 tag
    var div = h2.nextElementSibling; // h2 tag's sibling div tag

    // toggle plus and minus image in h2 elements by adding or removing a class
    if (h2.hasAttribute("class")) {
        h2.removeAttribute("class");
    } else {
        h2.setAttribute("class", "minus");
    }

    // toggle div visibility by adding or removing a class
    if (div.hasAttribute("class")) {
        div.removeAttribute("class");
    } else {
        div.setAttribute("class", "open");
    }
};
```



The JavaScript for FAQs application (cont.)

```
window.onload = function() {  
    // get the h2 tags  
    var faqs = $("faqs");  
    var h2Elements = faqs.getElementsByTagName("h2");  
  
    // attach event handler for each h2 tag  
    for (var i = 0; i < h2Elements.length; i++ ) {  
        h2Elements[i].onclick = toggle;  
    }  
    // set focus on first h2 tag's <a> tag  
    h2Elements[0].firstChild.focus();  
};
```



How to script forms and controls



How forms work

- A form in web browser

Please join our email list

Email Address: *

Re-enter Email Address: *

First Name *

- HTML code for the form

```
<form id="email_form" name="email_form" action="join.html" method="get">
  <label for="email_address1">Email Address:</label>
  <input type="text" id="email_address1" name="email_address1">
  <span>*</span><br>

  <label for="email_address2">Re-enter Email Address:</label>
  <input type="text" id="email_address2" name="email_address2">
  <span>*</span><br>

  <label for="first_name">First Name</label>
  <input type="text" id="first_name" name="first_name">
  <span>*</span><br>

  <label>&nbsp;</label>
  <input type="button" id="join_list" value="Join our List">
</form>
```

How forms work (cont.)

- Attributes of the form element

Attribute	Description
name	A name that can be referred by client-side or server-side code
action	The URL of the file that will process the data in the form.
method	The HTTP method for submitting the form data. It is get or set, default is get



How to script Textbox, Textarea and Select objects

- The web interface
- HTML code

First Name:

Comments:

Country:

Select an option	▼
Select an option	
USA	
Canada	
Mexico	

```
<label for="name">First Name:</label>
<input type="text" name="name" id="name"><br>

<label for="comments">Comments:</label>
<textarea name="comments" id="comments" rows="5" cols="40"></textarea>
<br>

<label for="country">Country:</label>
<select name="country" id="country">
  <option value="">Select an option</option>
  <option>USA</option>
  <option>Canada</option>
  <option>Mexico</option>
</select>
<br>
```

How to script Textbox, Textarea and Select objects (cont.)

- JavaScript code

```
var name =$("#name").value;
var comment =$("#comment").value;

if(name.length==0){alert("Please enter a name.");}
if(comment.length==0){alert("Please enter a comment.");}

var country =$("#country").value;
if(country=="usa"){/*USA processing*/}
else if(country=="can"){/*Canada processing*/}
else if(country=="mex"){/*Mexico processing*/}
else {alert("Please select a country.");}
```



How to script Radio and Checkbox objects

- Two properties of a Radio or Checkbox object

Property	Description
value	The contents of the value attribute for the button or check box. Return a string.
checked	If set to true, the button or check box is selected. If set to false, it isn't selected.

- Web interface

Contact me by: ☒ Text ☐ Email ☐ Mobile phone ☐ Don't contact me
Terms of Service: ☐ I accept



How to script Radio and Checkbox objects (cont.)

- HTML code

```
<label>Contact me by:</label>
<input type="radio" name="contact" id="text" value="text" checked>Text
<input type="radio" name="contact" id="email" value="email">Email
<input type="radio" name="contact" id="mobile" value="mobile">Mobile phone
<input type="radio" name="contact" id="none" value="none">Don't contact me
<br>
<label>Terms of Service:</label>
<input type="checkbox" name="terms" id="terms" value="yes">I accept <br>
```

- JavaScript code

```
var contact;
if($("#text").checked){contact= $("#text").value;}
if($("#email").checked){contact= $("#email").value;}

if(contact == "text"){/*text processing*/}
else if(contact=="email") {/*email processing*/}
else {alert("You must select a contact method")}

var accept = $("#accept").checked;
if(accept){
    /*accept processing*/
}else{
    alert("You cannot use the web store at this time");
}
```



How to use the methods and events for forms and controls

- Two methods that are commonly used with forms

Method	Description
submit()	Submits the form and its data to the server.
reset()	Resets the controls in the form to their starting values

- Two methods that are commonly used with controls

Method	Description
focus()	Moves the focus to the control.
blur()	Removes the focus from the control.



How to use the methods and events for forms and controls(cont.)

- Common control events

Event	Description
onfocus	The control receives the focus.
onblur	The control loses the focus.
onclick	The user clicks the control.
ondblclick	The user double-clicks the control
onchange	The value of the control changes
onselect	The user selects text in a text box or text area



How to use the methods and events for forms and controls(cont.)

- Statements that use the **reset()** and **submit()** methods

```
$( "registration_form" ).reset();
$( "registration_form" ).submit();
```
- An event handler for the **onchange** event of a select list

```
var investmentChange = function(){
    calculateClick();
    $( "investment" ).blur();
}
```



How to use the methods and events for forms and controls(cont.)

- An event handler for the **dbclick** event of a text box

```
var yearDbclick = function(){  
    $("years").value = "";  
};
```
- An **onload** event handler that assigns event handlers to events

```
window.onload = function(){  
    $("investment").onchange =investmentChange;  
    $("years").ondbclick = yearsDbclick;  
    $("years").focus();  
}
```



The Registration application

Page 184 -187



How to add and remove nodes from the DOM



How to use the innerHTML property of the Element interface

- The property of the Element interface for setting or returning HTML content

Property	Description
innerHTML	Sets or returns the HTML content of an element.

- How to get the HTML content of an element

```
var listHTML = $("image_list").innerHTML;
```
- How to remove the HTML content of an element

```
$("image_list").innerHTML = "";
```



How to use the innerHTML property of the Element interface(cont.)

- An application that display a list of images
 - HTML Code

```
<main>
  <h1>Create Image List</h1>
  <ul id="image_list"></ul>
</main>
```

– JavaScript code

```
<script>
  "use strict";
  var $ = function(id) {
    return document.getElementById(id);
  };
  window.onload = function() {
    $("image_list").innerHTML = "<li><img src='images/p1.jpg'></li>";
    $("image_list").innerHTML += "<li><img src='images/p2.jpg'></li>";
    $("image_list").innerHTML += "<li><img src='images/p3.jpg'></li>";
    $("image_list").innerHTML += "<li><img src='images/p4.jpg'></li>";
  };
</script>
```



How to use the innerHTML property of the Element interface(cont.)

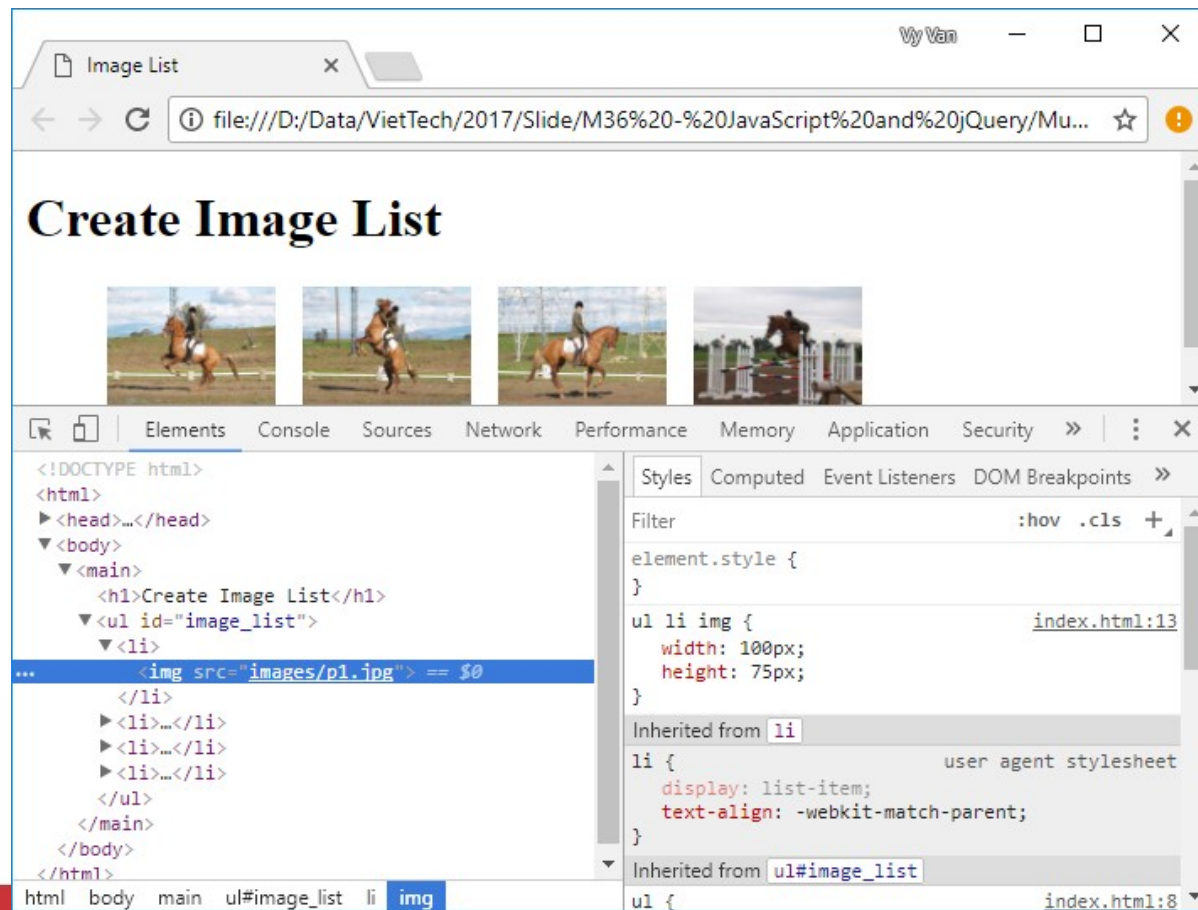
- An application that display a list of images
 - Web interface

Create Image List



How to view the changes to the DOM in Chrome

- You can use Chrome's development tools to see the changes to the DOM that your JavaScript has made.



The Register application with a table

Page (192 – 195)



Summary

- The **DOM(Document Object Model)** is a hierarchical collection of nodes in the web browser's memory that represents the current web page.
- The DOM of a web page is built as the page is loaded by web browser.
- JavaScript can modify the web page in the browser by modifying the DOM.
- To modify DOM you can use properties and methods of it.

