Web Development Technologies:
The DOM and JavaScript functions to modify the DOM

Paul Fodor
CSE316: Fundamentals of Software Development
Stony Brook University

http://www.cs.stonybrook.edu/~cse316
Topics

- The DOM
- JavaScript functions to modify the DOM
The DOM

- The Document Object Model (DOM)
- Tree of objects and attributes created by web browser from structure of web page
The DOM

- JavaScript can:
  - Change any HTML element
  - Change any HTML attribute
  - Change any CSS Style
  - Remove any HTML element or attribute
  - Add new HTML elements and attributes
  - React to all existing HTML events
  - Create new HTML events
The HTML DOM

- In this model:
  - HTML elements are objects
  - HTML elements have properties (like members) that can be altered
  - The DOM provides methods that can access all HTML elements
  - The DOM defines events for all HTML elements

- As with OO Paradigm
  - Methods are actions that can be performed
  - Properties are values that can be altered
The HTML DOM

- References to any element or the below get methods start with `document`.
  - This represents the document itself.

- Finding elements:
  - `getElementById(id)` – Finds elements with an id matching the argument (id=‘id’)
  - `getElementByTagName(name)` – Finds elements based on their tag name `<tag>` matches `<name>`
  - `getElementByClass(name)` – Finds elements with a class matching the argument (class = ‘name’)
    - **Note:** `getElementXXX()` calls may return more than 1 object. We will use subscripts to access a specific object.

- Key Properties – ‘element’ is a variable holding an object returned by one of the above get methods
  - `element.innerHTML` – This is the HTML content of an element
  - `element.attribute` – (where ‘attribute’ is an attribute name)
  - `element.style.property` – (where ‘property’ is the name of a style setting)
The HTML DOM Example

```html
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<meta http-equiv="X-UA-Compatible" content="ie=edge">
<title>Document</title>
</head>
<body>
<h1>Hello, World!</h1>
<ul>
<li>List item 1</li>
<li>List item 2</li>
</ul>
<ul>
<li>List item 3</li>
<li>List item 4</li>
</ul>
<p id="demo"></p>
</body>
</html>
```
The HTML DOM Example

// in the Console
var x = document.getElementsByTagName("li");
// gets all the li elements
console.log(x);

document.getElementById("demo").innerHTML =
'1st list item: ' + x[0].innerHTML
+ '<br>2nd list item: ' + x[1].innerHTML
+ '<br>3rd list item: ' + x[2].innerHTML
+ '<br>4th list item: ' + x[3].innerHTML;

document.getElementById("demo").style.color = "red";
Hello, World!

- List item 1
- List item 2
- List item 3
- List item 4

1st list item: List item 1
2nd list item: List item 2
3rd list item: List item 3
4th list item: List item 4
The HTML DOM Example

```html
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<meta http-equiv="X-UA-Compatible" content="ie=edge">
<title>Document</title>
</head>
<body>
<h1>Hello, World! !</h1>
<p id="one">This is text from paragraph 1</p>
<p id="two">This is text from paragraph 2</p>
<p id="three">This is text from paragraph 3</p>
<p class="unusual">This is an UNUSUAL paragraph</p>
<h1>Below we capture the UNUSUAL paragraph by 'selector'</h1>
<hr>
<p id="demo"></p>
<hr>
</body>
</html>
```

// in the console:
```javascript
var x = document.querySelectorAll("p.unusual");
document.getElementById("demo").innerHTML = x[0].innerHTML;
```
The HTML DOM Example

<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<meta http-equiv="X-UA-Compatible" content="ie=edge">
<title>Document</title>
</head>
<body>
<h1>Hello, World! !</h1>
<p id="one">This is text from paragraph 1</p>
<p id="two">This is text from paragraph 2</p>
<p id="three">This is text from paragraph 3</p>
<p class="unusual">This is an UNUSUAL paragraph</p>
<h1>Below we capture the UNUSUAL paragraph by 'selector'</h1>
<hr>
<p id="demo">This is text from paragraph 1</p>
<p>This is text from paragraph 2</p>
<p>This is text from paragraph 3</p>
<p>This is an UNUSUAL paragraph</p>
</body>
</html>

// demo.js:
var x = document.querySelectorAll("p.unusual");
document.getElementById("demo").innerHTML = x[0].innerHTML;

Hello, World! !
Below we capture the UNUSUAL paragraph by 'selector'

This is an UNUSUAL paragraph
The HTML DOM Example

```html
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<meta http-equiv="X-UA-Compatible" content="ie=edge">
<title>Document</title>
</head>
<body>
<h1>Hello, World! !</h1>
<p id="one">This is text from paragraph 1</p>
<p id="two">This is text from paragraph 2</p>
<p id="three">This is text from paragraph 3</p>
<p class="unusual">This is an UNUSUAL paragraph</p>
<h1>Below we capture the UNUSUAL paragraph by 'selector'</h1>
<hr>
<p id="demo"></p>
<hr>
<script>
var x = document.querySelectorAll("p.unusual");
document.getElementById("demo").innerHTML = x[0].innerHTML;
</script>
</body>
</html>
```

Hello, World! !
This is text from paragraph 1
This is text from paragraph 2
This is text from paragraph 3
This is an UNUSUAL paragraph

Below we capture the UNUSUAL paragraph by 'selector'
This is an UNUSUAL paragraph
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <meta http-equiv="X-UA-Compatible" content="ie=edge">
  <title>Document</title>
</head>
<body>
<h1>Hello, World!</h1>
<ul>
  <li>List item 1</li>
  <li>List item 2</li>
</ul>
<ul>
  <li>List item 3</li>
  <li>List item 4</li>
</ul>
<p id="demo"></p>
<script>
document.write("Hello, today is: " + Date());
</script>
</body>
</html>

JavaScript `document.write()` will add text directly into an HTML page.

Hello, World!

- List item 1
- List item 2
- List item 3
- List item 4

Hello, today is: Mon Sep 07 2020 14:42:45 GMT-0400 (Eastern Daylight Time)
You can change an attribute by getting the element and assigning a value to the attribute.

```html
<IMG id="image1" src="Paul1.jpg">
```

Can change the `src` attribute with:
```javascript
document.getElementById("image1").src = "Paul2.jpg";
```
The HTML DOM – Changing Styles

• You can change style by assigning to any of the style properties on an element

• The format to access a style property is:
  
  <object>.style.<propertyname>

• <object> is the object returned by getElementByXXX() calls

• style is just a keyword to indicate the upcoming field is a style attribute

• <propertyname> is the name of the CSS property
The HTML DOM – Changing Styles - Example

```html
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<meta http-equiv="X-UA-Compatible" content="ie=edge">
<title>Document</title>
</head>
<body>
<h1>Hello, World!</h1>
<ul>
<li>List item 1</li>
<li>List item 2</li>
</ul>
<ul>
<li>List item 3</li>
<li>List item 4</li>
</ul>
<p id="demo"></p>
<script>
var x = document.getElementsByTagName("li");
document.getElementById("demo").innerHTML = '1st list item: ' + x[0].innerHTML
    + '<br>2nd list item: ' + x[1].innerHTML
    + '<br>3rd list item: ' + x[2].innerHTML
    + '<br>4th list item: ' + x[3].innerHTML;
document.getElementById("demo").style.color = "red";
x[1].style.color = "red";
x[3].style.color = "green";
x[1].style.fontSize="16pt";
x[1].style.borderStyle = "dashed";
x[1].style.borderWidth = "2pt";
</script>
</body>
</html>
```

(c) Paul Fodor (CS Stony Brook)
The HTML DOM - Events

• Can execute code when an event occurs.

• Events include:
  • When a user clicks the mouse (onclick=)
  • When a web page has loaded (onload=, onunload=)
  • When an image has been loaded (onload=)
  • When the mouse moves over an element (onmouseover=, onmouseout=)
  • When an input field is changed (onchange=)
  • When an HTML form is submitted (onsubmit=)
  • When a user strokes a key (onkeypress=)
The HTML DOM – Events - Example

```html
<!DOCTYPE html>
<html>
<body>
<h1 id="id1">My Heading 1</h1>
<!-- we can put JavaScript code in the onclick action -->
<button type="button" onclick="document.getElementById('id1').style.color = 'red'">Click Me!</button>
</body>
</html>
```

Before clicking the ‘Click Me!’ button

My Heading 1

Click Me!

After clicking the ‘Click Me!’ button

My Heading 1

Click Me!
The HTML DOM – Building HTML

• JavaScript can modify the DOM of a document and add or remove elements

• Use ‘createXXX()’ methods:
  • createElement() – Creates and returns a new element node
  • createTextNode() – Creates a node that holds text

• Use various methods to add or remove nodes:
  • addChild() – Adds a child to a node
  • insertBefore() – Inserts a new node before a specific child
  • removeChild() – Removes a node from the DOM
  • replaceChild() – Replaces one child node with another
The HTML DOM – Building HTML Example

<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<title>Document</title>
</head>
<body>
<div id="theDiv">
<h1>Hello, World! !</h1>
<p id="one">This is text from paragraph 1</p>
<p id="two">This is text from paragraph 2</p>
<p id="four">This is text from paragraph 4</p>
<button onclick="remv(document.getElementById('two'))">Click to remove p2</button>
</div>
<script src="demo.js"></script>
</body>
</html>

demo.js:
var newPara = document.createElement("p");
var content = document.createTextNode("This is a new paragraph.");
newPara.appendChild(content);
var divElem = document.getElementById("theDiv");
divElem.appendChild(newPara);
function remv(element) {
    element.parentNode.removeChild(element);
}

Before click
Hello, World! !
This is text from paragraph 1
This is text from paragraph 2
This is text from paragraph 4
[Click to remove p2]
This is a new paragraph.

After click
Hello, World! !
This is text from paragraph 1
This is text from paragraph 4
[Click to remove p2]
This is a new paragraph.
The HTML DOM – Building HTML Example

```html
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<title>Document</title>
</head>
<body>
<div id="theDiv">
<h1>Hello, World! !</h1>
<p id="one">This is text from paragraph 1</p>
<p id="two">This is text from paragraph 2</p>
<p id="four">This is text from paragraph 4</p>
<button onclick="repl(document.getElementById('two'))">Click to replace p2</button>
</div>
<script>
function repl(element) {
    var para = document.createElement("p");
    var node = document.createTextNode("This is a new paragraph!");
    para.appendChild(node);
    element.parentNode.replaceChild(para, element);
}
</script>
</body>
</html>
```
The HTML DOM

- More HTML DOM actions:
  https://www.w3schools.com/js/js_htmldom.asp