DOCUMENT OBJECT MODEL & JAVASCRIPT

THOMAS SUSELO
TSUS@AUCKLANDUNI.AC.NZ
**DOCUMENT OBJECT MODEL (DOM)**

- **The HTML DOM** is a standard for how to get, change, add, or delete HTML elements.
- **Document Object Model** – an API that can be used by any language. A knowledge of the DOM will help with PHP or XML.
- Can be manipulated by other languages (e.g., Java, server-side JavaScript, Python, etc).
- It defines:
  - The HTML elements as objects.
  - The properties of all HTML elements.
  - The methods to access all HTML elements.
  - The events for all HTML elements.
• Objects are in a hierarchy
• The window is the parent for a given web page
• Document is the child with the objects that are most commonly manipulated
• Tree-structured
• Mostly access using JavaScript
DOM STRUCTURE

- Document
  - Root element: `<html>`
    - `<head>`
      - `<title>`: "My title"
    - `<body>`
      - `<a>`: "My link"
      - `<h1>`: "My header"
• 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
  • 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 is one of the 3 languages all web developers must learn:

1. HTML to define the content of web pages.
2. CSS to specify the layout of web pages.
3. JavaScript to program the behavior of web pages
   - You can place any number of scripts in an HTML document.
   - Scripts can be placed in the <body>, or in the <head> section of an HTML page, or in both.
DOM METHODS

• HTML DOM methods are actions you can perform (on HTML Elements).
• HTML DOM properties are values (of HTML Elements) that you can set or change.
• The HTML DOM can be accessed with JavaScript (and with other programming languages).
• In the DOM, all HTML elements are defined as objects.
• The programming interface is the properties and methods of each object.
• A method is an action you can do (like add or deleting an HTML element).
• A property is a value that you can get or set (like changing the content of an HTML element).

```html
<html>
<body>
    <p id="demo"></p>
    <script>
        document.getElementById("demo").innerHTML = "Hello World!";
    </script>
</body>
</html>
```
## FINDING HTML ELEMENTS & CHANGING HTML ELEMENTS

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>document.getElementById(id)</td>
<td>Find an element by element id</td>
</tr>
<tr>
<td>document.getElementsByTagName(name)</td>
<td>Find elements by tag name</td>
</tr>
<tr>
<td>document.getElementsByClassName(name)</td>
<td>Find elements by class name</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>element.innerHTML = new html content</td>
<td>Change the inner HTML of an element</td>
</tr>
<tr>
<td>element.attribute = new value</td>
<td>Change the attribute value of an HTML element</td>
</tr>
<tr>
<td>element.setAttribute(attribute, value)</td>
<td>Change the attribute value of an HTML element</td>
</tr>
<tr>
<td>element.style.property = new style</td>
<td>Change the style of an HTML element</td>
</tr>
</tbody>
</table>
## Adding and Deleting Elements & Adding Events Handlers

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>document.createElement(element)</code></td>
<td>Create an HTML element</td>
</tr>
<tr>
<td><code>document.removeChild(element)</code></td>
<td>Remove an HTML element</td>
</tr>
<tr>
<td><code>document.appendChild(element)</code></td>
<td>Add an HTML element</td>
</tr>
<tr>
<td><code>document.replaceChild(element)</code></td>
<td>Replace an HTML element</td>
</tr>
<tr>
<td><code>document.write(text)</code></td>
<td>Write into the HTML output stream</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>document.getElementById(id).onclick = function(){code}</code></td>
<td>Adding event handler code to an onclick event</td>
</tr>
</tbody>
</table>
FINDING HTML ELEMENTS

• There are a couple of ways to do this:

  – Finding HTML elements by id
    
    ```javascript
    var myElement = document.getElementById("intro");
    ```
  
  – Finding HTML elements by tag name
    
    ```javascript
    var x = document.getElementsByTagName("p");
    ```
  
  – Finding HTML elements by class name
    
    ```javascript
    var x = document.getElementsByClassName("intro");
    ```
  
  – Finding HTML elements by CSS selectors
    
    ```javascript
    var x = document.querySelectorAll("p.intro");
    ```
  
  – Finding HTML elements by HTML object collections
    
    ```javascript
    var x = document.forms["frm1"]; 
    var text = "";
    var i;
    for (i = 0; i < x.length; i++) {
      text += x.elements[i].value + "<br>";
    }
    document.getElementById("demo").innerHTML = text;
    ```
<!DOCTYPE html>
<html>
  <head>
    <title>Finding Element</title>
  </head>
  <body>
    <h1 id="id01">My First Page</h1>
    <p id="id02"></p>
    <script>
      document.getElementById("id02").innerHTML = document.getElementById("id01").innerHTML;
    </script>
  </body>
</html>
FINDING ELEMENTS

```html
<!DOCTYPE html>
<html>
<body>
<h1 id="id1">Heading Level 1</h1>
<button type="button"
onclick="document.getElementById('id1').style.color = 'red'">Red!</button>
<button type="button"
onclick="document.getElementById('id1').style.color = 'black'">Black!</button>
</body>
</html>
```
<form id="frm1" action="/action_page.php">
    First name: <input type="text" name="fname" value="Donald"><br>
    Last name: <input type="text" name="lname" value="Duck"><br><br>
    <input type="submit" value="Submit">
</form>

<p>Click "Try it" to display the value of each element in the form.</p>

<button onclick="myFunction()">Try it</button>

<p id="demo"></p>

<script>
    function myFunction() {
        var x = document.forms["frm1"];  
        var text = "";
        var i;
        for (i = 0; i < x.length ;i++) {
            text += x.elements[i].value + "<br>
        }
        document.getElementById("demo").innerHTML = text;
    }
</script>
CHANGING HTML

- https://www.w3schools.com/js/js_htmldom_html.asp
DOM EVENTS

- DOM allows JavaScript to react to HTML events
- Examples 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
PASSING PARAMETERS

- When passing parameter values, use an "anonymous function" that calls the specified function with the parameters:

```javascript
element.addEventListener("click", function(){ myFunction(p1, p2); });
```
This example demonstrates how to pass parameter values when using the `addEventListener()` method.

Click the button to perform a calculation.

```html
<!DOCTYPE html>
<html>
<body>
<p>This example demonstrates how to pass parameter values when using the `addEventListener()` method.</p>
<p>Click the button to perform a calculation.</p>
<button id="myBtn">Try it</button>
<p id="demo"></p>
<script>
var p1 = 5;
var p2 = 7;
document.getElementById("myBtn").addEventListener("click", function() {myFunction(p1, p2);});

function myFunction(a, b) {
    var result = a * b;
    document.getElementById("demo").innerHTML = result;
}
</script>
</body>
</html>
```
This example uses the `addEventListener()` method to add two click events to the same button.

Button HTML:
```html
<button id="myBtn">Try it</button>
```

JavaScript Code:
```javascript
var x = document.getElementById("myBtn");
x.addEventListener("click", myFunction);
x.addEventListener("click", someOtherFunction);

function myFunction() {
    alert("Hello World!");
}

function someOtherFunction() {
    alert("This function was also executed!");
}
```

End HTML Code:
SOURCES

• W3schools.com
Q/A

- tsus609@aucklanduni.ac.nz
- Piazza