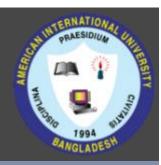# JS HTML DOM and Form Validation

Course Code: CSC 3222   Course Title: Web Technologies

Dept. of Computer Science
Faculty of Science and Technology

Lecture No:	12	Week No:	11	Semester:	Fall 2024-2025
Lecturer:	*MD.AL-AMIN (alamin@aiub.edu)*				

Lecture Outline

1. JS Control FLow
2. JS Array and Array methods
3. JS Objects
4. HTML DOM operations (Basic usage and CSS)
5. HTML form validation
6. JS Debugging

JS Control Flow

- If else statements are used for changing control flow of program.
- The syntax is familiar with other programming language.

```html
<html>
    <body>
        <button onclick="myFunction()">Try it</button>
        <p id="demo"></p>
        <script>
            function myFunction() {
              var hour = 19;
              var greeting;
              if (hour < 18) {
                greeting = "Good day";
              } else if (hour > 20){
                greeting = "Good night";
              } else {
                greeting = "Good evening";
              }

              document.getElementById("demo").innerHTML = greeting;
            }
        </script>
    </body>
</html>
```

JS switch

```
var x = 0;
var text;
switch (x) {
  case 0:
    text = "Off";
    break;
  case 1:
    text = "On";
    break;
  default:
    text = "No value found";
}
```

- Switch case is used in stead of `else if`
- Switch cases use **strict** comparison (===).
- The values must be of the same type to match.
- A strict comparison can only be true if the operands are of the same type.

```
var x = "0";            ──────────►  string
var text;
switch (x) {
  case 0:     ──────┐
    text = "Off";    │
    break;           ├──────►  number
  case 1:     ──────┘
    text = "On";
    break;
  default:
    text = "No value found";
}
```

- In this example there will be no match for `x`

JS Loops

For Loop

```js
for (i = 0; i < 5; i++) {
  text += "The number is " + i + "<br>";
}
```

while Loop

```js
while (i < 10) {
  text += "The number is " + i;
  i++;
}
```

do while Loop

```js
while (i < 10) {
  text += "The number is " + i;
  i++;
}
```

Must Read: https://www.w3schools.com/js/js_loop_for.asp
https://www.w3schools.com/js/js_loop_while.asp

JS Arrays

- JavaScript arrays are used to store multiple values in a single variable.
- Using an array literal is the easiest way to create a JavaScript Array.
  ```
  var array_name = [item1, item2, ...];
  var cars = ["Saab", "Volvo", "BMW"];
  ```
- You can also create array with new keyword.
  ```
  var cars = new Array("Saab", "Volvo", "BMW");
  ```
- You access an array element by referring to the index number.
  ```
  var name = cars[0];
  ```
- Array indexes start with 0.
- Changing an Array Element.
  ```
  cars[0] = "Opel";
  ```
- With JavaScript, the full array can be accessed by referring to the array name
  ```
  document.getElementById("demo").innerHTML = cars;
  ```
- Arrays are a special type of objects. The typeof operator in JavaScript returns "object" for arrays.

JS Arrays

- You can have objects in an Array. You can have functions in an Array. You can have arrays in an Array

```
myArray[0] = Date.now;
myArray[1] = myFunction;
myArray[2] = myCars;
```

- Looping through an array

```
var fruits, text, fLen, i;
fruits = ["Banana", "Orange", "Apple", "Mango"];
fLen = fruits.length;
text = "<ul>";
for (i = 0; i < fLen; i++) {
  text += "<li>" + fruits[i] + "</li>";
}
text += "</ul>";
```

- You can also use the `Array.forEach()`

  -
```
var fruits, text;
fruits = ["Banana", "Orange", "Apple", "Mango"];
text = "<ul>";
fruits.forEach(myFunction);
text += "</ul>";
function myFunction(value) {
  text += "<li>" + value + "</li>";
}
```

Array Methods

- The JavaScript method `toString()` converts an array to a string of (comma separated) array values.

```
var fruits = ["Banana", "Orange", "Apple", "Mango"];
document.getElementById("demo").innerHTML = fruits.toString();
```

- The `join()` method also joins all array elements into a string.

```
var fruits = ["Banana", "Orange", "Apple", "Mango"];
document.getElementById("demo").innerHTML = fruits.join(" * ");
```

- The `pop()` method removes and returns the last element from an array

```
var fruits = ["Banana", "Orange", "Apple", "Mango"];
var x = fruits.pop();              // Removes the last element ("Mango") from fruits
```

- The `push()` method adds and returns the value of a new element to an array (at the end)

```
var fruits = ["Banana", "Orange", "Apple", "Mango"];
var x = fruits.push("Kiwi");   //  the value of x is 5
```

HTML DOM [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

<button type="button"
onclick="document.getElementById('id1').style.color = 'red'">
Click Me!</button>
```

My Heading 1

Click Me!

- All Properties of Style: https://www.w3schools.com/jsref/dom_obj_style.asp

JS Form Validation

- HTML form validation can be done by JavaScript..
- Form has an event named onsubmit
- When a form is submitted this method is called. We need to return true if values are valid otherwise false.
- If true is returned the form will be submitted.

JS Form Validation

```html
<html>
<head>
<script>
function validateForm() {
  var fname = document.getElementById("fname").value;
  var lname = document.getElementById("lname").value;
  if (fname == "" || lname == "") {
    alert("Name must be filled out");
    return false;
  }
}
</script>
</head>
<body>

<form name="myForm" action="/action_page.php" onsubmit="return validateForm()"
method="post">
  First Name: <input type="text" id="fname" name="fname"><br>
  Last Name: <input type="text" id="lname" name="lname"><br>
  <input type="submit" value="Submit">
</form>

</body>
</html>
```

JS Debugging

- If your browser supports debugging, you can use console.log() to display JavaScript values in the debugger window
- The debugger keyword stops the execution of JavaScript, and calls (if available) the debugging function

```
var x = 15 * 5;
debugger;
document.getElementById("demo").innerHTML = x;
```

Books

1. W3Schools Online Web Tutorials; URL: http://www.w3schools.com
2. PHP Documentation; URL: http://www.php.net/docs.php
3. Sams Teach Yourself Ajax JavaScript and PHP All in One; Phil Ballard and Michael Moncur; Sams Publishing; 2010
4. JavaScript Phrasebook; Christian Wenz; Sams Publishing; 2007
5. PHP and MySQL Web Development, 4/E; Luke Welling and Laura Thomson; AddisonWesley Professional; 2009
6. JavaScript for Programmers Paul J. Deitel and Harvey M. Deitel; Prentice Hall; 2009
7. Beginning PHP5, Apache, and MySQL Web Development; Elizabeth Naramore, Jason Gerner, Yann Le Scouarnec, Jeremy Stolz and Michael K. Glass; Wiley Publishing; 2005
8. XML in a Nutshell, 3/E; Elliotte Rusty Harold and W. Scott Means; O'Reilly Media; 2004

References

1. https://www.w3schools.com/js/
2. https://www.springboard.com/blog/history-of-javascript/