	CSE 482L: Internet and Web Technology Lab	Lab Manual
	Faculty: naqib.hussain@northsouth.edu Instructor: shabbir.ahmed01@northsouth.edu Updated: 2023-Oct-25	4

Goal: Introductory JavaScript (JS):[a]

- Basics, loading inline vs block vs external
- DOM access, function calls, event handling, array
- Debugging by using browser's DevTools

JS Basics

File name: **my_first_js.html**

```
<!DOCTYPE html>
<html>
<body>
    <h2>My First JavaScript</h2>
    <button type="button" onclick="window.alert('assalamualaikum')">
        Click me to see a greetings popping up.</button>
</body>
</html>
```

What does the `onclick` attribute do? Find out what each of the above methods do. This example demonstrates an inline JS that executes on an event (click). But you may load JS as a block of code or as an external file.

File name: **in_block_js_within_script_tag.html**

```
<!DOCTYPE html>
<html>
<head></head>
<body>
    <h2>My First JavaScript</h2>
    <button type="button" id="mybutton">Click me to see an alert popup.</button>
    <script>
        document.getElementById('mybutton').addEventListener("click", function(){
            window.alert('assalamualaikum');
        });
    </script>
</body>
</html>
```

File name: **script_from_external.html**

```
<!DOCTYPE html>
<html>
<head></head>
<body>
    <h2>My First JavaScript</h2>
    <button type="button" id="mybutton">Click me to see an alert popup.</button>
    <script src="http://127.0.0.1/my_script.js"></script>
</body>
</html>
```

File name: **my_script.js**

```
document.getElementById('mybutton').addEventListener("click", function(){
    window.alert('assalamualaikum');
});
```

Save the HTML and the JS file in the same folder htdoc of XAMPP (or any other location that you use as web-root of your server) then browse to this html file. This is external JS loading. The above 3 examples do the same function. What are their differences with respect to execution? What approach seems more manageable when your html and JS file grow larger? In the above `script_from_external.html` file if you move the `<script>` tag from `<body>` to the `<head>` does it work? Can you explain why?

Accessing Document Object Model (DOM)

File name: **access_element_by_id.html**

```
<!DOCTYPE html>
<html>
<head></head>
<body>
    <h2 id="my_h2">...</h2>
    <button type="button" onclick="popup();">Click me to see who works</button>
    <script>
        function popup() {
            document.getElementById('my_h2').innerHTML = "My JS works hard!";
        }
    </script>
</body>
</html>
```

File name: **access_elements_by_class.html**

```
<!DOCTYPE html>
<html>
<head></head>
<body>
    <h2 class="my_h2">...</h2>
    <button type="button" onclick="popup();">Click me to see who works</button>
    <script>
        function popup() {
            var x = document.getElementsByClassName('my_h2');
            x[0].innerHTML = "My JS works hard!";
        }
    </script>
</body>
</html>
```

File name: **access_dom_elements_by_tag_names.html**

```
<!DOCTYPE html>
<html>
<head></head>
<body>
    <h2>...</h2>
    <button type="button" onclick="popup();">Click me to see who works</button>
    <script>
        function popup() {
            document.getElementsByTagName('h2')[0].innerHTML = "My JS works hard!";
        }
    </script>
</body>
</html>
```

File name: **js_manipulate_css.html**

```
<!DOCTYPE html>
<html>
<head></head>
<body>
    <h1>JS can access CSS properties of elements</h1>
    <div id="demo" style="background-color:red; height:10rem;
width:20rem;"></div><br>
    <button type="button" onclick="popup();">Click me to who works</button>
    <script>
        function popup() {
            document.getElementById("demo").style.backgroundColor = "blue";
        }
    </script>
</body>
</html>
```

Event Handling and Function Calls

File name: **onload_event.html**

```
<!DOCTYPE html>
<html>
<head></head>
<body onload="popup();">
    <h1>JS works at onload event</h1>
    <div id="demo" style="background-color:red; height:10rem;
width:20rem;"></div><br>
    <script>
        function popup() {
            window.alert("DOM is loaded! Now, the div will become blue.");
            document.getElementById("demo").style.backgroundColor = "blue";
        }
    </script>
</body>
</html>
```

File name: **onkeydown_up_event.html**

```
<!DOCTYPE html>
<html>
<head>
    <style>
        div{
            background-color:black; height:10rem; width:20rem;
            text-align:center; font-size: 4rem; color:white;
        }
    </style>
</head>
<body>
    <h1>JS captures keydown / keyup event</h1>
    <div id="demo">Type in the input box</div><br>
    <input type="textarea" onkeydown="keydownfunction();"
onkeyup="keyupfunction();">
    <script>
        const demo_div = document.getElementById("demo");
        function keydownfunction() {
            demo_div.style.backgroundColor = "red";
            demo_div.innerHTML = "Key down!";
            demo_div.style.color="black";
        }
        function keyupfunction() {
            demo_div.style.backgroundColor = "blue";
            demo_div.innerHTML = "Key up!";
            demo_div.style.color="white";
        }
    </script>
</body>
</html>
```

Array Handling

File name: **array_handling.html**

```
<!DOCTYPE html>
<html>
<head>
    <style>
        div {
            border-color: black; border-style: solid; margin-bottom: 1rem;
            height: 5rem; width: 20rem;
            text-align: center; font-size: 2rem; color: black;
        }
    </style>
</head>
<body>
    <h1>Array Handling by JS</h1>
    <div id="div_show"></div><br>
    <input type="number" name="anynumber" id="anynumber"><span> </span>
    <button onclick="my_push(document.getElementById('anynumber').value);">
      Push</button>
    <button onclick="my_pop();">Pop</button><br><br>
    <button onclick="my_show();">Show array</button><br>
    <script>
        points = new Array(); // this is an empty array
        div_show = document.getElementById("div_show");
        number_input = document.getElementById('anynumber');
        function my_push(number) {
            if (number != "") {
                points.push(number);
                div_show.innerHTML = "Pushed: " + number;
            }else
                div_show.innerHTML = "Nothing to push";
            number_input.value ="";
        }
        function my_pop() { div_show.innerHTML = "Poped: " + points.pop(); }
        function my_show() {
            if(points.length !=0){
                div_show.innerHTML = points.toString();;
            }else{
                div_show.innerHTML = "Empty";
            }

        }
    </script>
</body>
</html>
```

File name: **local_storage_demo.html**

```
<!DOCTYPE html>
<html>
<body>
    <p id="demo"></p>
    <script>
        localStorage.setItem("name", "Beautiful Bangladesh");
        document.getElementById("demo").innerHTML = localStorage.getItem("name");
    </script>
</body>
</html>
```

Browse to the this file from your browser. When it displays Beautiful Bangladesh, open developer's tool and go to application tab. Then from the left panel under storage expand local storage as follows:

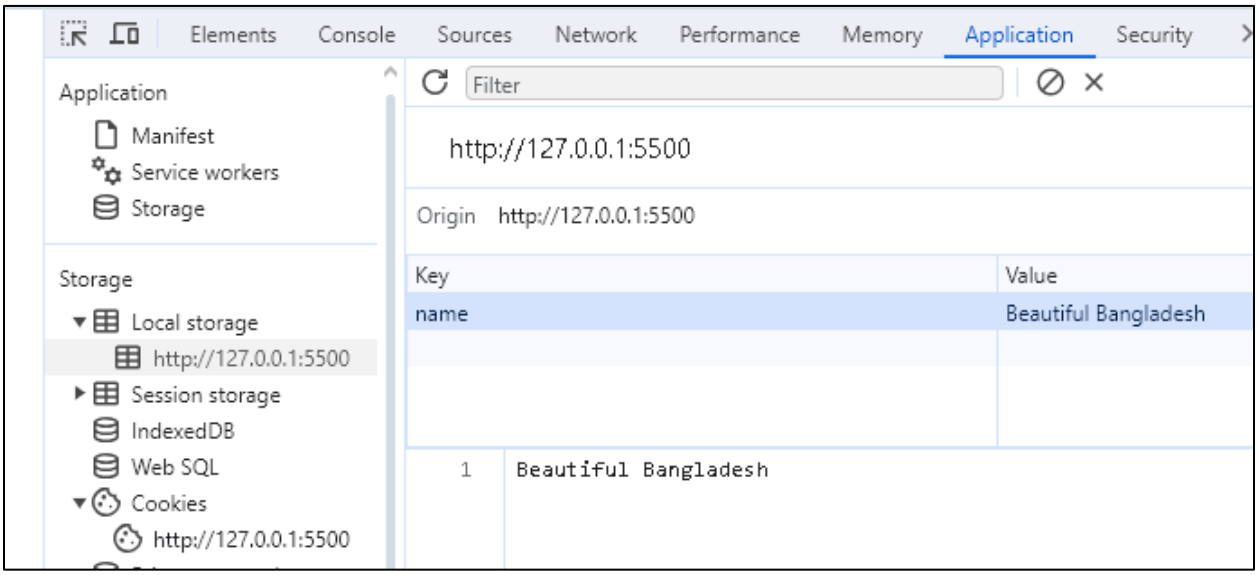


Figure 1: LocalStorage viewing from DevTool

File name: **browser_history_demo.html**

```
<!DOCTYPE html>
<html>
<head>
    <title>Demonstrates browser history handling</title>
</head>
<body>
    <h1>Demonstrates browser history handling</h1>
    <button onclick="myFunction()">Go Back</button>
```

```
    <script>
        function myFunction() {
            window.history.back();
        }
    </script>
</body>
</html>
```

Now press this button to make the browser go back to previous page stored in its history. Can you change his code to make to forward or go to a specific history?

File name: **form_validation_demo.html**

```
<!DOCTYPE html>
<html>
<body>
    <h2>JavaScript Validation</h2>
    <p>For a number less than 570 or greater than 2023, an error msg will showup.</p>
    <div id="notif" style="border-style:solid;border-color:black;height:2rem;width:20rem;"></div>
    <br>
    <input id="anynumberid" type="number" min="570" max="2023" required>
    <button onclick="myFunction()">OK</button>

    <script>
        function myFunction() {
            const my_input = document.getElementById("anynumberid");
            const my_notif = document.getElementById("notif");
            if (!my_input.checkValidity()) {
                my_notif.innerHTML = my_input.validationMessage;
            } else {
                my_notif.innerHTML = "Input OK";
            }
        }
    </script>
</body>
</html>
```

Use of REGEX

File name: **regex_demo.html**

```
<!DOCTYPE html>
<html>
<style>
    div {
        border-style: solid; border-color: black;
        height: 2rem; width: 20rem;
        text-align: center;
    }
</style>
<body>
    <h2>JavaScript Regular Expressions</h2>
    <div id="notif"></div>
    <p>Input a valid email address or a BD mobile number</p>
    <label for="email_mobile"></label>
    <input type="text" id="email_mobile"><br><br>
    <button type="button" onclick="my_checkValidity();">Test validity</button>
    <script>
        function my_checkValidity() {
            let text = document.getElementById("email_mobile").value;
            let my_notif = document.getElementById("notif");
            let mobile = /^\+?(88)?0?1[3456789][0-9]{8}\b/i;
            let email = /^[\w-\.]+@([\w-]+\.)+[\w-]{2,4}$/i;
            let valid_mobile = text.match(mobile);
            let valid_email = text.match(email);
            if(valid_mobile){
                my_notif.innerHTML = "Valid mobile: " + valid_mobile;
                my_notif.style.backgroundColor = "lightgreen";
            }else if(valid_email){
                my_notif.innerHTML = "Valid email: " + valid_email;
                my_notif.style.backgroundColor = "lightgreen";
            }else{
                my_notif.innerHTML = "Not valid!";
                my_notif.style.backgroundColor = "red";
            }
        }
    </script>
</body>
</html>
```

Can you explain the pattern used in regex?

"^" means starting of the expression. "?" means matches an optional. Here after "+" there is a "?" this means + is optional. Similarly, after (88), there is "?" means 88 is optional. "0?" Means 0 is optional. 1 is mandatory. Then [3456789] is mandatory. [0-9] is the Range and {8} quantifier. \b: boundary of word, i.e., start-of-word or end-of-word.

JS Debugging

1. Copy these codes index.html and get-started.js and save these on same file name. Then put both of them under *htdocs* so that you can use `console.log();` or debugger keyword; or breakpoints
2. Use `console.log()` to display JavaScript values in the debugger window
3. Enable debugger. The debugger keyword stops the execution of JavaScript, and calls the debugging function.
4. In the debugger window, you can set breakpoints in the JavaScript code. At each breakpoint, JavaScript will stop executing, and let you examine JavaScript values.
5. Enable DevTools, event listener breakpoint on mouse click and track. Follow the steps to the basic workflow from https://developer.chrome.com/docs/devtools/javascript/ for debugging any JavaScript issue in DevTools and do the same works https://googlechrome.github.io/devtools-samples/debug-js/get-started

File name: **`index.html`**

```
<!-- referenced in: https://developers.google.com/web/tools/chrome-
devtools/javascript -->
<!doctype html>
<html>
<head>
    <title>Demo: Get Started Debugging JavaScript with Chrome DevTools</title>
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <style>
        h1 { font-size: 1.5em; }
        input, button {
            min-width: 72px;
            min-height: 36px;
            border: 1px solid grey;
        }
        label, input, button { display: block; }
        input { margin-bottom: 1em; }
    </style>
```

```
</head>
<body>
    <h1>Demo: Get Started Debugging JavaScript with Chrome DevTools</h1>
    <label for="num1">Number 1</label>
    <input placeholder="Number 1" id="num1">
    <label for="num2">Number 2</label>
    <input placeholder="Number 2" id="num2">
    <button>Add Number 1 and Number 2</button>
    <p></p>
    <script src="get-started.js"></script>
</body>
</html>
```

File name: **get-started.js**

```
function onClick() {
    if (inputsAreEmpty()) {
        label.textContent = 'Error: one or both inputs are empty.';
        return;
    }
    updateLabel();
}
function inputsAreEmpty() {
    if (getNumber1() === '' || getNumber2() === '') {
        return true;
    } else {
        return false;
    }
}
function updateLabel() {
    var addend1 = getNumber1();
    var addend2 = getNumber2();
    var sum = addend1 + addend2;
    label.textContent = addend1 + ' + ' + addend2 + ' = ' + sum;
}

function getNumber1() {
    return inputs[0].value;
}

function getNumber2() {
    return inputs[1].value;
```

```
}
var inputs = document.querySelectorAll('input');
var label = document.querySelector('p');
var button = document.querySelector('button');
button.addEventListener('click', onClick);
```

Lab Task

(a) Change the **`array_handling.html`** to work with named array. Seek help from https://www.w3schools.com/js/js_arrays.asp

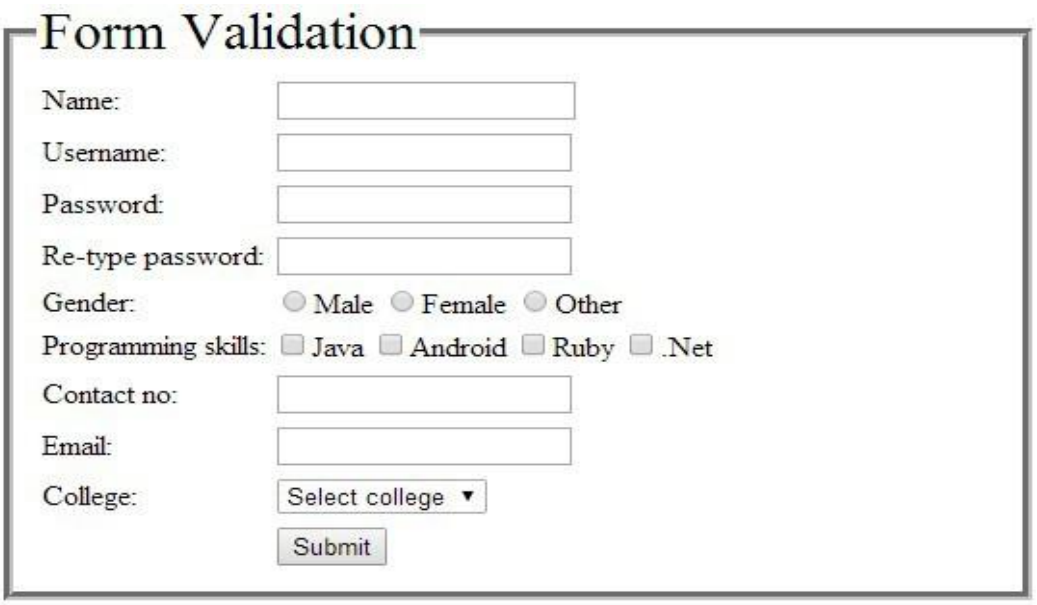
Named Array Example

```
const person = [];
person["firstName"] = "Muhammad";
person["lastName"] = "Abdullah";
person["age"] = 40;
```

(b) If there is any error use the DevTools debugger to debug your tasks.

Assignment

(a) Create a form as shown blow (Figure 2)
(b) Implement the following validation rules:
   i) Name cannot be empty
   ii) Username cannot be empty and cannot contain whitespace
   iii) Password length is between 8-32 chars
   iv) Gender is selected
   v) Contact no. contains numbers only



Practice to Learn More

1) Explore examples at W3schools (https://www.w3schools.com/js/default.asp) as much as possible. However, run examples from the following URLs and then make changes in the code, then bring all codes in the next class in a pen drive or google drive.

A. https://www.w3schools.com/js/js_dates.asp
B. https://www.w3schools.com/js/js_this.asp
C. https://www.w3schools.com/js/js_arrow_function.asp
D. https://www.w3schools.com/js/js_classes.asp
E. https://www.w3schools.com/js/js_object_definition.asp
F. https://www.w3schools.com/js/js_mistakes.asp

Figure 2: Form to validate

End Notes

[a] This Lab manual bases on W3chools: https://www.w3schools.com