

# JavaScript 2 - Libraries

## Agenda

- Closures
- jQuery
- Underscore/Lodash
- FullCalendar
- WebSockets
- Browser Libraries
  - Notifications
  - localStorage
  - JSON

## Closures

- <https://developer.mozilla.org/en-US/docs/Web/JavaScript/Closures>
- “associate some data (the lexical environment) with a function that operates on the data”.
- Isolates your environment.
- Environment lives from page load to page unload.

### Examples:

```
1 <script>
2
3 var counter = (function() {
4     var privateCounter = 0;
5     function changeBy(val) {
6         privateCounter += val;
7     }
8     return {
9         increment: function() {
10             changeBy(1);
11         },
12         decrement: function() {
```

```
13     changeBy(-1);
14     },
15     value: function() {
16         return privateCounter;
17     }
18 };
19 })();
20
21 </script>
```

## jQuery

- “Write less, do more”
- Allows for
  - HTML/DOM manipulation.
  - CSS manipulation.
  - HTML event methods.
  - Effects and animations.
  - AJAX
  - Utilities

### Examples:

#### HTML/DOM

```
1 <script>
2 $('target').action();
3 </script>
```

#### CSS Manipulation

```
1 <script>
2 $('target').css('property', 'value');
3 </script>
```

#### AJAX

```
1 <script>
2 // Normal JavaScript.
```

```
3 var xhttp = new XMLHttpRequest();
4 xhttp.onreadystatechange = function() {
5     if (this.readyState == 4 && this.status == 200) {
6         document.getElementById("demo").innerHTML =
7             this.responseText;
8     }
9 };
10 xhttp.open("GET", "ajax_info.txt", true);
11 xhttp.send();
12
13
14 // Get
15 $.get("demo_test.asp", function(data, status){
16     alert("Data: " + data + "\nStatus: " + status);
17 });
18
19 $.post("url", {}, function(data, status) {
20 // Do stuff here with response...
21 });
22
23 //More advanced
24 $.post('url', {})
25 .done(function(data) {
26 //This is good. Do some code!
27 }).fail(function(data, status) {
28 // Not good!
29 }).always(function() {
30 //What do we need this for?
31 });
32
33 // Synchronous vs. asynchronous?
34 </script>
```

## EVENTS

```
1 <script>
2 $('selector').event(function() {
3 // Do something here...
4 });
5 </script>
```

## Underscore/Lodash

<http://underscorejs.org/>

<https://lodash.com/docs>

### Some code?

<http://underscorejs.org/underscore.js>

## FullCalendar

<https://fullcalendar.io/>

<https://fullcalendar.io/docs/usage/>

## WebSockets

- TCP-based protocol.
- Uses HTTP for initial handshake.
  - Communication goes through port 80/443.
- Lower overhead.
- Facilitates 'real-time' data transfer to/from server.
  - Allows for standard way for server to send data to client without client requesting it.
  - Keeps connection open, two-way (bi-directional) ongoing conversation (state full vs stateless).
- Supported by most browsers.
- <https://socket.io/docs/>
- Demo + HW description.

# ~~TypeScript Script (Maybe)~~

## Built in Browser Libraries

- Notifications
- LocalStorage
- JSON

### Notifications

```
1 <script>
2 // Source: https://developer.mozilla.org/en-US/docs/Web/API/notification
3
4 function notifyMe() {
5     // Let's check if the browser supports notifications
6     if (!("Notification" in window)) {
7         alert("This browser does not support desktop notification");
8     }
9
10    // Let's check whether notification permissions have already been granted
11    else if (Notification.permission === "granted") {
12        // If it's okay let's create a notification
13        var notification = new Notification("Hi there!");
14    }
15
16    // Otherwise, we need to ask the user for permission
17    else if (Notification.permission !== "denied") {
18        Notification.requestPermission(function (permission) {
19            // If the user accepts, let's create a notification
20            if (permission === "granted") {
21                var notification = new Notification("Hi there!");
22            }
23        });
```

```
24     }
25
26     // At last, if the user has denied notifications, and you
27     // want to be respectful there is no need to bother them any more.
28 }
29
30 </script>
```

## LocalStorage

```
1 <script>
2 var key = "id";
3 var value = 5;
4
5 // Set item.
6 localStorage.setItem(key, value);
7
8 // Get item.
9 value = localStorage.getItem(key);
10
11 // Remove/Clear item.
12 localStorage.removeItem(key)
13 // or
14 localStorage.clear(key)
15
16 </script>
```

## JSON

```
1 <script>
2 var json = {'name': 'Firefox'};
3
4 // Convert JavaScript object to JSON string.
5 var json_string = JSON.stringify(json);
```

```
6  
7 // Convert JSON string to JavaScript object.  
8 var json = JSON.parse(json_string);  
9  
10 </script>
```