

# Sessions, jQuery and Ajax

Web Development and Security (ZEIT3119)

Week 5

Dr. Reza Rafeh

# Revision

slido

Join at  
**slido.com**  
**#3986 139**



# Outline

- **Dynamic pages**
- **Browser Storage**
- Local Storage
- Session Storage
- Cookies
- **jQuery**
- jQuery/CSS Selectors
- **AJAX**
- load
- get
- post

# Revision: Dynamic Behavior by JS



Dynamic Tables:

## Student Information

First name:

Last name:

Add Student




First Name		Last Name	
Jack		Smith	
Maria		Brown	

## Student Information

First name:

Last name:

Add Student

First Name		Last Name	
Jack		Smith	
Maria		Brown	
James		Wilkinson	

# Dynamic Tables – Add a New Row

```
function addStudent(){
    let name = document.getElementById("fname").value;
    let lastname = document.getElementById("lname").value;
    let table = document.getElementById("student_table");
    let nrow = table.rows.length;
    table.insertRow(nrow);
    let row = table.rows[nrow];
    let cell1 = row.insertCell(0);
    let cell2 = row.insertCell(1);
    let cell3 = row.insertCell(2);
    cell1.innerHTML = name;
    cell2.innerHTML = lastname;
    cell3.innerHTML = '<button class="btn" id="trash_'+nrow+'" onclick=deleteStudent(this)><i
class="fa fa-trash"></i></button>';
}
```

# Dynamic Tables – Delete a Row

```
function deleteStudent(r) {  
    var i = r.parentNode.parentNode.rowIndex;  
    document.getElementById("student_table").deleteRow(i);  
}
```

# Store Data in Browser Storage

Stores data in browser storage for later use. It is useful for front-end developers who have no database.

Data can be used in different pages. So, it could be a mechanism of passing data to other pages.

Three ways to store data:

- Local Storage
- Session Storage
- Cookies

# Motivating Example

Admins can add/delete students but lecturers can only see students:

## Sign In

Email:

john@adfa.edu.au

Password:

.....

☐ Remember me

Sign in

## Sign In

Email:

reza@adfa.edu.au

Password:

.....

☐ Remember me

Sign in

Admin can add/delete – Buttons are active

Lecturer can't add/delete – Buttons are disabled

## Student Information

First name:

Last name:

Add Student

First Name

Last Name

Jack

Smith



Maria

Brown



## Student Information

First name:

Last name:

Add Student

First Name

Last Name

Jack

Smith



Maria

Brown





# Cookies vs Local Storage vs Session Storage

	Cookies	Local Storage	Session Storage
Capacity	4kb	10mb	5mb
Browsers	HTML4/HTML5	HTML5	HTML5
Accessible from	Any window	Any window	Same tab
Expires	Manually set	Never	On tab close
Storage location	Browser and server	Browser only	Browser only
Sent with requests	Yes	No	No

# Local Storage

```
localStorage.setItem('userType', 'student');  
console.log(localStorage.getItem('userType'));  
localStorage.removeItem('userType');
```

# Session Storage

```
sessionStorage.setItem('userType', 'student');  
console.log(sessionStorage.getItem('userType'));  
sessionStorage.removeItem('userType');
```

# Session Storage – Store Objects

```
UserData = {name: 'Reza', role: 'Lecturer'};
```

```
sessionStorage.setItem('user', userData); //This will store data as an object, not very useful
```

```
sessionStorage.setItem('user',JSON.stringify(userData)); //Better to store it as a string
```

```
JSON.parse(sessionStorage.getItem('user')); //Convert the string back to object
```

```
JSON.parse(sessionStorage.getItem('user')).name; //Returns user's name
```

# Cookies

```
document.cookie = 'userName=Reza';  
document.cookie = 'userName=Reza ; expires='+new Date(2023,4,4);  
function setCookie(cname, cvalue, exdays) {  
    const d = new Date();  
    d.setTime(d.getTime() + (exdays*24*60*60*1000));  
    let expires = "expires="+ d.toUTCString();  
    document.cookie = cname + "=" + cvalue + ";" + expires + ";path=/";  
}
```

**These don't work on a local file. They can be running on a server (local or remote)**

# jQuery

- jQuery is a powerful javaScript library
- There are three branches of jQuery; 1.x,2.x and 3.x, designed for different environment
- Many functions to make programming tasks easier
- Uses short-hand notations
- Outline for jQuery:
  - Introduction
  - Selectors
  - Events
  - Ajax

# Different Flavors of jQuery

## ➤ **Version 1.x**

- First stable release of jQuery
- Supports older web browser

## ➤ **Version 1.12 is the best**

## ➤ **Version 2.x**

- Does not support IE version 6 – 8 (does not support older versions)
- Faster and smaller than version 1.x

## ➤ **Version 3.x**

- Current version 3.6.0 launched in March 2021
- For loops are introduced in 3.x
- Code for showing and hiding have been upgrade (.show(),.hide())
- Increase compatibility with responsive design
- Download the latest jQuery from <https://jquery.com/download/>

# jQuery

- To use jQuery, you must include it in your HTML
- jQuery uses “\$” symbol which acts as the jQuery factory method
- For changing the font family of all paragraphs to monospace, use the following statement

```
$('p').css('font-family', 'monospace')
```

- Adding a border to a `<code>` element

```
$('code').css('border', '1px solid #aaa')
```

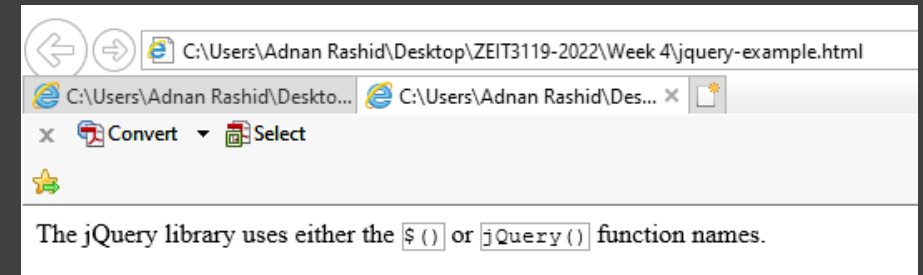
- Hiding the element with `id="test"`

```
$("#test").hide()
```



# jQuery Example

```
<!DOCTYPE html>
<html>
<head>
<script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.0/jquery.min.js"></script>
</head>
<body>
The jQuery library uses either the $()
or jQuery() function names.
<script>
$('code').css('border', '1px solid #aaa')
</script>
</body>
</html>
```



# jQuery Example

```
<!DOCTYPE html>
<html>
<head>
<script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.0/jquery.min.js"></script>
<script>
$(document).ready(function(){
  $("p").click(function(){
    $(this).hide();
  });
});
</script>
</head>
<body>
<p>If you click on me, I will disappear.</p>
<p>Click me away!</p>
</body>
</html>
```

If you click on me, I will disappear.

Click me away!

If you click on me, I will disappear.

# jQuery/CSS Selectors

- Styling one or more elements using CSS and applying jQuery on the same selected elements

- **Element Selector**

- Select the element manipulated by jQuery
- List its name within the parenthesis following by \$ symbol

`$(selector).action()`

- Changing the background color of all <blockquote> elements

`$('blockquote').css('background', 'red')`

- **ID Selector**

- Apply jQuery to the IDs by placing a “#” character in-front of the ID

`$('#advert').css('border', '3px dashed red')`

# jQuery/CSS Selectors

- **Class Selector**

- Manipulate groups of elements according to the class they use:

```
$('.new').css('text-decoration', 'underline')
```

- **Combining the Selectors**

- Just like CSS, selectors can be combined into a single jQuery using commas:

```
$('blockquote, #advert, .new').css('font-weight', 'bold')
```

- **Events**

- jQuery allows us to assign **handler functions** to **events** on selected **elements**

- Example

```
$("button") . click (function ( ){ alert("Hello"); });
```

# jQuery/CSS Selectors - Example

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<title>Second jQuery Example</title>
```

```
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.0/jquery.min.js"></script>
```

```
<script>
```

```
$(document).ready(function(){
```

```
    $("button").click(function(){
```

```
        $('blockquote').css('background', 'lime');
```

```
        $('#advert').css('border', '3px dashed red');
```

```
        $('.new').css('text-decoration', 'underline');
```

```
        $('blockquote, #advert, .new').css('font-weight', 'bold');
```

```
    });
```

```
});
```

```
</script>
```

```
</head>
```

```
<body>
```

Powerful and flexible as JavaScript is, with a plethora of built-in functions, it is still necessary to use additional code for simple things that cannot be achieved natively or with CSS, such as animations, event handling, and asynchronous communication.

This is an ad

This is my new website

Add classes to elements

Powerful and flexible as JavaScript is, with a plethora of built-in functions, it is still necessary to use additional code for simple things that cannot be achieved natively or with CSS, such as animations, event handling, and asynchronous communication.

This is an ad

This is my new website

Add classes to elements

# jQuery/CSS Selectors - Example

```
<!DOCTYPE html>

<html>

<head>

<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>

</head>

<body>

<h2>Click in and out of these fields</h2>

<input id='first'> <input> <input> <input>

<script>

$('#first').focus()

$('input').focus(function() { $(this).css('background', 'yellow') } )

$('input') .blur(function() { $(this).css('background', 'grey') } )

</script>

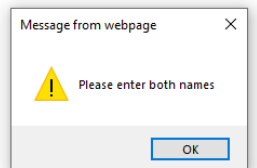
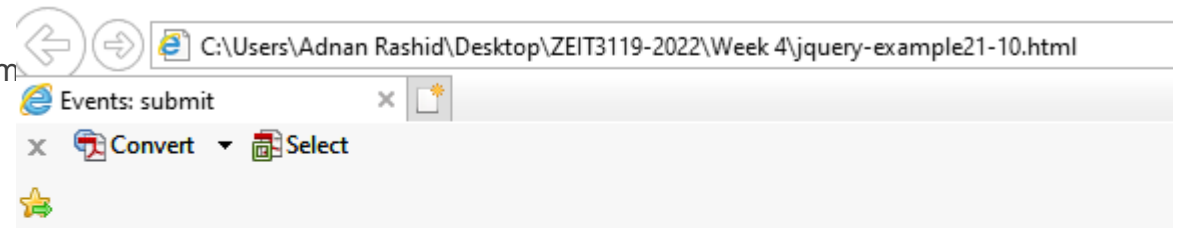
</body>

</html>
```

**Click in and out of these fields**

# jQuery/CSS Selectors - Example

```
<!DOCTYPE html>
<html>
<head>
<title>Events: submit</title>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.m
</head>
<body>
<form id='form'>
First name: <input id='fname' type='text' name='fname'><br>
Last name: <input id='lname' type='text' name='lname'><br>
<input type='submit'>
</form>
<script>
$('#form').submit(function()
{
if ($('#fname').val() == " ||
$('#lname').val() == ")
{
alert('Please enter both names')
return false
}
})
</script>
</body>
</html>
```



# jQuery/CSS Selectors - Example

## Summary

Selector	Example	Description
*	<code>\$("*")</code>	All elements
#id	<code>\$("#fred")</code>	The element with id="fred"
.class	<code>\$(".ethel")</code>	All elements with class "ethel"
HTML element type	<code>\$("p")</code>	All <p> elements
Parent descendant	<code>\$("div p")</code>	All <p> elements that are descendants of a <div>



# More jQuery Examples

- To hide the first column of a table

```
$('#items th:eq(0)').hide();
```

- To add an extra cell to each row

```
$('#items tbody tr').append('<td>Extra cell</td>');
```

- To hid contents of column 4

```
$('#items td:nth-child(4)').hide();
```

# jQuery and Ajax

- AJAX – Asynchronous JavaScript and XML
- It is about loading data in the background and displaying it on the webpage, without reloading the whole page
- Ajax is not a single technology, but rather a combination of HTML, CSS, the DOM, and JavaScript
- Applications using AJAX
  - Gmail
  - Google Maps
  - Youtube
  - Facebook tabs
- jQuery provides several methods for AJAX functionality
- Request text, HTML , XML from a remote server using both Http GET and Http POST
- A single line code is written for AJAX functionality while using jQuery

# jQuery and Ajax

## load()

- The jQuery load() method is simple, but powerful AJAX method
- It loads data from a server and puts the returned data into the selected element
- Syntax

`$(selector).load(URL,data,callback);`

- URL parameter – specifies the URL you wish to load
- data parameter (optional) - specifies a set of query string key/value pairs to send along with the request
- callback parameter (optional) - is the name of a function to be executed after the load() method is completed.

# jQuery and Ajax – load()

Here is the content of our example file: "demo\_test.txt":

```
<h2>jQuery and AJAX is FUN!!!</h2>
<p id="p1">This is some text in a paragraph.</p>
```

Let jQuery AJAX Change This Text

Get External Content

jQuery and AJAX is FUN!

This is some text in a paragraph.

Get External Content

```
<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.3/jquery.min.js">
</script>
<script>
$(document).ready(function(){
    $("button").click(function(){
        $("#div1").load("demo_test.txt");
    });
});
</script>
</head>
<body>
<div id="div1"><h2>Let jQuery AJAX Change This Text</h2></div>
<button>Get External Content</button>
</body>
</html>
```

# jQuery and AJAX

## GET() & POST()

- The jQuery `get()` and `post()` methods are used to request/send data from/to the server with an HTTP GET or POST request.

- Syntax

`$.get(URL, callback);`

`$.post(URL, data, callback);`

- URL – specifies the URL
- Callback (optional) – name of the function to be executed if the request succeeds
- Data (optional) - specifies some data to send along with the request

# jQuery and AJAX – get()

```
<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.3/jquery.min.js">
</script>
<script>
$(document).ready(function(){
  $("button").click(function(){
    $.get("demo_test.php", function(data, status){
      alert("Data: " + data + "\nStatus: " + status);
    });
  });
});
</script>
</head>
<body>
  <button>Send an HTTP GET request to a page and get the result back</button>
</body>
</html>
```

```
<?php
echo "This is some text from an external PHP
file.";
?>
```

← → ↻ ⓘ localhost

Send an HTTP GET request to a page and get the result back

## localhost says

Data: This is some text from an external PHP file.  
Status: success

OK



UNSW  
CANBERRA

# jQuery and AJAX – post()

```
<!DOCTYPE html>
<html>
<head>
  <title>AJAX Example</title>
  <script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>
</head>
<body>
  <h1>AJAX Example</h1>
  <input type="text" id="input-text">
  <button id="submit-btn">Submit</button>
  <div id="result"></div>
  <script>
    $(document).ready(function() {
      $('#submit-btn').click(function() {
        $.ajax({
          url: 'submit.php',
          method: 'POST',
          data: { text: $('#input-text').val() },
          success: function(result) {
            $('#result').html(result);
          }
        });
      });
    });
  </script>
</body>
</html>
```

```
<?php
if ($_SERVER['REQUEST_METHOD'] == 'POST') {
    $text = $_POST['text'];
    echo 'You entered: ' . $text;
}
?>
```

← → ↻ ⓘ localhost

## AJAX Example

## AJAX Example

You entered: Reza

# Final Note

- **Continue working on Project 1**
- **Based on the feedback you received from your peers, please revise the design and work on the front-end coding using HTML, CSS, JavaScript**
- **Please attend the labs and seek help from your lab demonstrator when you come up with any question or doubt**