### National University of Computer & Emerging Sciences, Karachi Computer Science Department



|  |  |
| --- | --- |
| **Course Code: SE-3003** | **Course : Web Engineering Lab** |

**Spring 2023, Lab Manual – 07**

**LLO 02: Design and Implement a web application using JavaScript.**

**Contents:**

* Intro to Web Engineering
* Technologies
* Tools
* Introduction to jQuery
* jQuery syntax
* jQuery Fading Methods
* jQuery Animation

**Introduction to Web Engineering**

Web Engineering is the application of systematic and quantifiable approaches (concepts methods, techniques tools) to cost ‐ effective requirements analysis, design, implementation, testing, operation, and maintenance of **high quality Web applications.**

**Technologies to be studied**

* HTML
* CSS
* JavaScript
* Bootstrap
* JQuery
* PHP
* MySQL [Database]
* Laravel [PHP FRAMEWORK]

**Tools – IDEs**

* Visual Studio Code
* Adobe Dreamweaver
* Visual Studio

**7.1 Introduction to jQuery**

jQuery is a fast, small, and feature-rich JavaScript library. It makes things like HTML document traversal and manipulation, event handling, animation, and Ajax much simpler with an easy-to-use API that works across a multitude of browsers. With a combination of versatility and extensibility, jQuery has changed the way that millions of people write JavaScript.

jQuery is a lightweight, "write less, do more", JavaScript library. The purpose of jQuery is to make it much easier to use JavaScript on your website. jQuery takes a lot of common tasks that require many lines of JavaScript code to accomplish, and wraps them into methods that you can call with a single line of code.

jQuery also simplifies a lot of the complicated things from JavaScript, like AJAX calls and DOM manipulation.

The jQuery library contains the following features:

* HTML/DOM manipulation
* CSS manipulation
* HTML event methods
* Effects and animations
* AJAX
* Utilities

**Adding jQuery to Your Web Pages**

There are several ways to start using jQuery on your web site. You can:

* Download the jQuery library from jQuery.com
* Include jQuery from a CDN, like Google

## Downloading jQuery

There are two versions of jQuery available for downloading:

* Production version - this is for your live website because it has been minified and compressed
* Development version - this is for testing and development (uncompressed and readable code)

Both versions can be downloaded from [jQuery.com](http://jquery.com/download/).

The jQuery library is a single JavaScript file, and you reference it with the HTML <script> tag (notice that the <script> tag should be inside the <head> section):

<head>  
<script src="jquery-3.6.3.min.js"></script>  
</head>

**Tip:** Place the downloaded file in the same directory as the pages where you wish to use it.

**jQuery CDN**

If you don't want to download and host jQuery yourself, you can include it from a CDN (Content Delivery Network).

Google is an example of someone who host jQuery:

<head>  
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.3/jquery.min.js"></script>  
</head>

**7.2 jQuery Syntax**

The jQuery syntax is tailor-made for selecting HTML elements and performing some action on the element(s).

* Basic syntax is: $(selector).action()
* A $ sign to define/access jQuery
* A (selector) to "query (or find)" HTML elements
* A jQuery action() to be performed on the element(s)

**Examples:**

* $(this).hide() - hides the current element.
* $("p").hide() - hides all <p> elements.
* $(".test").hide() - hides all elements with class="test".
* $("#test").hide() - hides the element with id="test".

**The Document Ready Event**

You might have noticed that all jQuery methods in our examples, are inside a document ready event:

$(document).ready(function(){

// jQuery methods go here...

});

This is to prevent any jQuery code from running before the document is finished loading (is ready). It is good practice to wait for the document to be fully loaded and ready before working with it. This also allows you to have your JavaScript code before the body of your document, in the head section. Here are some examples of actions that can fail if methods are run before the document is fully loaded:

Trying to hide an element that is not created yet

Trying to get the size of an image that is not loaded yet

## jQuery hide() and show()

$("#hide").click(function(){  
  $("p").hide();  
});  
  
$("#show").click(function(){  
  $("p").show();  
});

*$(selector).hide(speed,callback);  
$(selector).show(speed,callback);*

The optional speed parameter specifies the speed of the hiding/showing, and can take the following values: "slow", "fast", or milliseconds.

The optional callback parameter is a function to be executed after the hide() or show() method completes (you will learn more about callback functions in a later chapter).

The following example demonstrates the speed parameter with hide():

$("button").click(function(){  
  $("p").hide(1000);  
});

## 7.3 jQuery Fading Methods

With jQuery you can fade an element in and out of visibility.

jQuery has the following fade methods:

* fadeIn()
* fadeOut()
* fadeToggle()
* fadeTo()

## jQuery fadeIn() Method

The jQuery fadeIn() method is used to fade in a hidden element.

Syntax:

$(selector).fadeIn(speed,callback);

The optional speed parameter specifies the duration of the effect. It can take the following values: "slow", "fast", or milliseconds. The optional callback parameter is a function to be executed after the fading completes. The following example demonstrates the fadeIn() method with different parameters:

$("button").click(function(){  
  $("#div1").fadeIn();  
  $("#div2").fadeIn("slow");  
  $("#div3").fadeIn(3000);  
});

## jQuery fadeOut() Method

The jQuery fadeOut() method is used to fade out a visible element.

Syntax:

$(selector).fadeOut(speed,callback);

The optional speed parameter specifies the duration of the effect. It can take the following values: "slow", "fast", or milliseconds.

The optional callback parameter is a function to be executed after the fading completes.

The following example demonstrates the fadeOut() method with different parameters:

$("button").click(function(){  
  $("#div1").fadeOut();  
  $("#div2").fadeOut("slow");  
  $("#div3").fadeOut(3000);  
});

## jQuery fadeToggle() Method

The jQuery fadeToggle() method toggles between the fadeIn() and fadeOut() methods. If the elements are faded out, fadeToggle() will fade them in. If the elements are faded in, fadeToggle() will fade them out.

**Syntax:**

$(selector).fadeToggle(speed,callback);

The optional speed parameter specifies the duration of the effect. It can take the following values: "slow", "fast", or milliseconds. The optional callback parameter is a function to be executed after the fading completes.

The following example demonstrates the fadeToggle() method with different parameters:

$("button").click(function(){  
  $("#div1").fadeToggle();  
  $("#div2").fadeToggle("slow");  
  $("#div3").fadeToggle(3000);  
});

## jQuery Sliding Methods

With jQuery you can create a sliding effect on elements.

jQuery has the following slide methods:

* slideDown()
* slideUp()
* slideToggle()

## jQuery slideDown() Method

The jQuery slideDown() method is used to slide down an element.

Syntax:

$(selector).slideDown(speed,callback); The optional speed parameter specifies the duration of the effect. It can take the following values: "slow", "fast", or milliseconds. The optional callback parameter is a function to be executed after the sliding completes.

The following example demonstrates the slideDown() method:

$("#flip").click(function(){  
  $("#panel").slideDown();  
});

**jQuery slideUp() Method**

The jQuery slideUp() method is used to slide up an element.

Syntax:

$(selector).slideUp(speed,callback);

The optional speed parameter specifies the duration of the effect. It can take the following values: "slow", "fast", or milliseconds. The optional callback parameter is a function to be executed after the sliding completes.

The following example demonstrates the slideUp() method:

$("#flip").click(function(){  
  $("#panel").slideUp();  
});

**jQuery slideToggle() Method**

The jQuery slideToggle() method toggles between the slideDown() and slideUp() methods.

If the elements have been slid down, slideToggle() will slide them up. If the elements have been slid up, slideToggle() will slide them down.

$(selector).slideToggle(speed,callback);

The optional speed parameter can take the following values: "slow", "fast", milliseconds. The optional callback parameter is a function to be executed after the sliding completes. The following example demonstrates the slideToggle() method:

$("#flip").click(function(){  
  $("#panel").slideToggle();  
});

**7.4 JQuery Animations - The animate() Method**

The jQuery animate() method is used to create custom animations.

*Syntax:*

*$(selector).animate({params},speed,callback);*

The required params parameter defines the CSS properties to be animated. The optional speed parameter specifies the duration of the effect. It can take the following values: "slow", "fast", or milliseconds. The optional callback parameter is a function to be executed after the animation completes. The following example demonstrates a simple use of the animate() method; it moves a <div> element to the right, until it has reached a left property of 250px:

$("button").click(function(){  
  $("div").animate({left: '250px'});  
});

## jQuery animate() - Manipulate Multiple Properties

Notice that multiple properties can be animated at the same time:

$("button").click(function(){  
  $("div").animate({  
    left: '250px',  
    opacity: '0.5',  
    height: '150px',  
    width: '150px'  
  });  
});

# **jQuery**Stop Animations

# **jQuery stop() Method**

The jQuery stop() method is used to stop an animation or effect before it is finished. The stop() method works for all jQuery effect functions, including sliding, fading and custom animations.

Syntax:

$(selector).stop(stopAll,goToEnd);

The optional stopAll parameter specifies whether also the animation queue should be cleared or not. Default is false, which means that only the active animation will be stopped, allowing any queued animations to be performed afterwards. The optional goToEnd parameter specifies whether or not to complete the current animation immediately. Default is false. So, by default, the stop() method kills the current animation being performed on the selected element. The following example demonstrates the stop() method, with no parameters:

$("#stop").click(function(){  
  $("#panel").stop();  
});

## jQuery Callback Functions

JavaScript statements are executed line by line. However, with effects, the next line of code can be run even though the effect is not finished. This can create errors. To prevent this, you can create a callback function. A callback function is executed after the current effect is finished.

***Typical syntax: $(selector).hide(speed,callback);***

Examples: The example below has a callback parameter that is a function that will be executed after the hide effect is completed:

$("button").click(function(){  
  $("p").hide("slow", function(){  
    alert("The paragraph is now hidden");  
  });  
});

The example below has no callback parameter, and the alert box will be displayed before the hide effect is completed:

$("button").click(function(){  
  $("p").hide(1000);  
  alert("The paragraph is now hidden");  
});

***Task***

*Task 01: Write a jQuery code to do some effect using locally used jQuery without Extension and same effect using extension in VS-Code.*

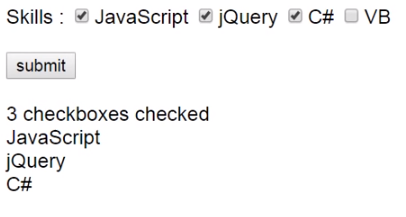
*Task 02: Create a HTML file having performing all functionalities of jQuery using your own logic using HTML TAGS, which discussed in this LAB.*

*Task 03: Write a jQuery code to animated progress bar when its reached to 100% then redirect to next page on button click.*

*Task 04: Write a jQuery code to animated div[500, 500] anti-clock wise including animated effect with some delay.*

*Task 05: Create a signup page including jQuery Validation ONLY.*

*Task 06: Create the checkboxes in HTML and validate using jQuery as like below image.*

**

*Task 07: Create a web base Quiz-App using jQuery and Bootstrap, the quiz has 5 questions, based on question attempt the progress-Bar filled and after the fifth question submission the result will be shown.*