

University Mohamed Boudiaf - Msila  
Faculty of Mathematics and Computer Science  
Department of Computer Science  
1st Year Master - RTIC



# Web Technologies

## jQuery

Mohamed Kamel

Academic Year 2022/2023

# Table of Contents

- 1 Introduction
- 2 Syntax
- 3 Selectors
- 4 Events
- 5 Effects
- 6 Content Manipulation
- 7 Traversing

# Introduction

jQuery is a JavaScript library, that simplifies programming. With jQuery we can do:

- HTML/DOM manipulation
- CSS manipulation
- HTML event methods
- Effects and animations
- AJAX

# Installation

- JQuery is a JavaScript file
- You simply need to include the file into your webpage

```
1      <script src="jquery-3.6.0.min.js">
      </script>
```

---

- Or link directly to the online version

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

---

# Syntax

Basic syntax is:

```
1 $(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)

## jQuery example

```
1 $(this).hide() //hides the current element
2 $("p").hide() //hides all <p> elements
3 $(".test").hide() //hides all elements with
   class="test"
4 $("#test").hide() //hides the element with
   id="test"
```

---

# Calling

```
1 $(document).ready(function(){  
2  
3     // jQuery methods go here...  
4  
5 });
```

---

# element Selector

The jQuery element selector selects elements based on the element name, We can select all `<p>` elements like this:

## Selecting all `<p>` elements

```
1      $("p")
```

In the following example, all `<p>` elements will be hidden when user clicks on any button

## Example

```
1      $(document).ready(function(){  
2          $("button").click(function(){  
3              $("p").hide();  
4          });  
5      });
```

# #id Selector

The jQuery element selector selects the element based on its id, We can select the element with id="test" like this:

## Selecting all <p> elements

```
1      $("#test")
```

---

In the following example, the element with id="test" will be hidden when user clicks on any button

## Example

```
1      $(document).ready(function(){  
2          $("button").click(function(){  
3              $("#test").hide();  
4          });  
5      });
```

---



# .class Selector

The jQuery .class selector finds elements with a specific class. We can select elements with specific like this:

## Selecting all <p> elements

```
1      $(".test")
```

In the following example, the element with id="test" will be hidden when user clicks on any button

## Example

```
1      $(document).ready(function(){  
2          $("button").click(function(){  
3              $(".test").hide();  
4          });  
5      });
```

# More Selectors

- `$("*")` Selects all elements
- `$(this)` Selects the current HTML element
- `$("p.intro")` Selects all `<p>` elements with `class="intro"`
- `$("p:first")` Selects the first `<p>` element
- `$("ul li:first")` Selects the first `<li>` element of the first `<ul>`
- `$("ul li:first-child")` Selects the first `<li>` element of every `<ul>`
- `$("[href]")` Selects all elements with an href attribute

# Events

jQuery is made to respond to events in an HTML page

Some event:

- Mouse Events
  - **click**, when clicking on an element
  - **dblclick**, when double clicking on an element
  - **mouseenter**, when mouse pointer enters an element
  - **mouseleave**, when mouse pointer leaves an element
- Keyboard Events
  - **keypress**, when a keyboard key is pressed and released
  - **keydown**, when a keyboard key pressed
  - **keyup**, when a keyboard key is released

# Events

- Form Events
  - **submit**, when a form is submitted
  - **change**, when content of form element, selection or checked state have changed
  - **focus**, when an element gets focus
  - **blur**, when an element loses focus
- Document/Window Events
  - **load**, when an object has loaded
  - **resize**, when the document view is resized
  - **scroll**, when an element's scrollbar is being scrolled
  - **unload**, when a page has unloaded (closed)

## Example

```
1      $("#mybutton").click(function(){
2          alert("Hello World");
3      });
```

# Effects

- Simple Show/Hide:
  - **show()**, shows a hidden element
  - **hide()**, hides a visible element
- Fade
  - **fadeIn()**, fades in a hidden element
  - **fadeOut()**, fades out a visible element
  - **fadeToggle()**, toggles between fadeIn() and fadeOut() methods
  - **fadeTo()**, fades to a given opacity

# Effects

- Slide:
  - **slideDown()**, slides down an element
  - **slideUp()**, slides up an element
  - **slideToggle()**, toggles between slideDown() and slideUp() methods
- Animation:
  - **animate()**, creates a custom animations

## Syntax

```
1 $(selector).animate({params}, speed,  
    callback);
```

- **stop()**, stops animations or effects before it is finished

# Animation

Moving a <div> element to the position left = 250px

## Simple Animation Example

```
1      $("button").click(function(){
2          $("#mydiv").animate({left: '250px'});
3      });
```

---

Multiple properties can be animated at the same time

## Animation Example

```
1      $("button").click(function(){
2          $("div").animate({
3              left: '+=100px',
4              opacity: '0.5',
5              height: '150px',
6              width: '150px'
7          });
8      });
```

# Animation

- **Callback function**, A callback function is executed after the current effect is 100% finished.

## Callback Example

```
1      $("button").click(function(){
2          $("#mypara").hide("slow", function(){
3              alert("The paragraph is now hidden");
4          });
5      });
```

---

- **Chaining**, Chaining allows us to run multiple jQuery methods (on the same element) within a single statement.

## Chaining Example

```
1      $("#mypara").slideUp(2000).slideDown(2000)
      ;
```

---



# Content Manipulation

- **text()** - Sets or returns the text content

```
1      $("#mytextarea").text("New Text Content");
```

- **html()** - Sets or returns the content (including HTML markup)

```
1      $("#mydiv").html("<a href='example.html'>  
      Link</a>");
```

- **val()** - Sets or returns the value of form fields

```
1      $("#myinput").val("New Value");  
2      alert($("#myinput").val());
```

# Content Manipulation

- **append()** - Inserts content at the end of the selected elements
- **prepend()** - Inserts content at the beginning of the selected elements
- **after()** - Inserts content after the selected elements
- **before()** - Inserts content before the selected elements

```
1      $("p").append("New text at the end.");
2      $("p").prepend("New text at the beginning.");
3      ;
4      $("img").after("Text after the image");
5      $("img").before("Text before the image");
```

# Content Manipulation

- **remove()** - Removes the selected element (and its child elements)
- **empty()** - Removes the child elements from the selected element

```
1      $("#mydiv").empty();  
2      $("#mydiv").remove();
```

---

# CSS Manipulation

- **addClass()**, Adds one or more classes
- **removeClass()**, Removes one or more classes
- **toggleClass()**, Toggles between addClass() and removeClass()
- **css()**, Sets or returns the style attribute

```
1  $("#mybutton1").click(function(){
2    $("h1").addClass("mycalss");
3    $("#mypara").addClass("myotherclass");
4  });
5  $("#mybutton2").click(function(){
6    $("h1").removeClass("mycalss");
7    $("#mypara").removeClass("myotherclass");
8  });
9  $("#mybutton3").click(function(){
10   $("h1").toggleClass("mycalss");
11   $("#mypara").toggleClass("myotherclass");
12   $("p").css("background-color", "red");
13  });
```

# Ancestors

- `parent()`, direct parent
- `parents()`, all parents
- `parentsUntil()`, all parents until the given selector

```
1 $("span").parent();  
2 $("span").parents();  
3 $("span").parentUntil("div");
```

---

# Descendants

- `children()`, all direct children
- `find()`

```
1 $("div").children();  
2 $("div").find("p");
```

---

# Ancestors

- **siblings()**, all sibling elements
- **next()**, the next sibling element
- **nextAll()**, all next sibling elements
- **nextUntil()**, all next sibling elements until the given selector
- **prev()**
- **prevAll()**
- **prevUntil()**

```
1 $("h2").siblings("p");  
2 $("h2").next();  
3 $("h2").nextAll();  
4 $("h2").nextUntil("h6");
```

---

Questions ?