**Tutorial 3 Learning JQuery**

**1. jQuery?**

jQuery is a fast and concise JavaScript Library created by John Resig in 2006 with a nice motto: **Write less, do more**.

jQuery simplifies HTML document traversing, event handling, animating, and Ajax interactions for rapid web development.

jQuery is a JavaScript toolkit designed to simplify various tasks by writing less code. Here is the list of important core features supported by jQuery −

* **DOM manipulation** − The jQuery made it easy to select DOM elements, negotiate them and modifying their content by using cross-browser open source selector engine called **Sizzle**.
* **Event handling** − The jQuery offers an elegant way to capture a wide variety of events, such as a user clicking on a link, without the need to clutter the HTML code itself with event handlers.
* **AJAX Support** − The jQuery helps you a lot to develop a responsive and featurerich site using AJAX technology.
* **Animations** − The jQuery comes with plenty of built-in animation effects which you can use in your websites.
* **Lightweight** − The jQuery is very lightweight library - about 19KB in size (Minified and gzipped).
* **Cross Browser Support** − The jQuery has cross-browser support, and works well in IE 6.0+, FF 2.0+, Safari 3.0+, Chrome and Opera 9.0+
* **Latest Technology** − The jQuery supports CSS3 selectors and basic XPath syntax.

**1.1 How to use jQuery?**

There are two ways to use jQuery.

**Local Installation** − You can download jQuery library on your local machine and include it in your HTML code.

Go to the <https://jquery.com/download/> to download the latest version available.

Now put downloaded **jquery-3.7.1.min.js** file in a directory of your website, e.g. /jquery.

**CDN Based Version** − You can include jQuery library into your HTML code directly from Content Delivery Network (CDN).

<html>

<head>

<title>The jQuery Example</title>

<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.7.1/jquery.min.js">

</script>

<script type = "text/javascript" language = "javascript">

$(document).ready(function() {

$("div").click(function() {alert("Hello, world!");});

});

</script>

</head>

<body>

<div id = "mydiv">

Click on this to see a dialogue box.

</div>

</body>

</html>

**1.2 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".

**1.3 jQuery Selectors**

jQuery selectors allow you to select and manipulate HTML element(s).

jQuery selectors are used to "find" (or select) HTML elements based on their name, id, classes, types, attributes, values of attributes and much more. It's based on the existing [CSS Selectors](https://www.w3schools.com/cssref/css_selectors.asp), and in addition, it has some own custom selectors.

All selectors in jQuery start with the dollar sign and parentheses: $().

**1.3.1 The element Selector**

The jQuery element selector selects elements based on the element name.

You can select all <p> elements on a page like this: $("p")

**Example**

When a user clicks on a button, all <p> elements will be hidden:

|  |
| --- |
| <html>  <head>  <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.2.1/jquery.min.js"></script>  <script>  $(document).ready(function()  {  $("button").click(function()  {  $("p").hide();  });  });  </script>  </head>  <body>  <h2>This is a heading</h2>  <p>This is a paragraph.</p>  <p>This is another paragraph.</p>  <button>Click me to hide paragraphs</button>  </body>  </html> |

**1.3.2 The #id Selector**

The jQuery #id selector uses the id attribute of an HTML tag to find the specific element.

An id should be unique within a page, so you should use the #id selector when you want to find a single, unique element.

To find an element with a specific id, write a hash character, followed by the id of the HTML element*:* ***$("#test")***

|  |
| --- |
| <html>  <head>  <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.2.1/jquery.min.js"></script>  <script>  $(document).ready(function(){  $("button").click(function(){  $("#test").hide();  });  });  </script>  </head>  <body>  <h2>This is a heading</h2>  <p>This is a paragraph.</p>  <p id="test">This is another paragraph.</p>  <button>Click me</button>  </body>  </html> |

**1.3.3 The .class Selector**

The jQuery class selector finds elements with a specific class.

To find elements with a specific class, write a period character, followed by the name of the class: ***$(".test")***

|  |
| --- |
| <!DOCTYPE html>  <html>  <head>  <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.2.1/jquery.min.js"></script>  <script>  $(document).ready(function(){  $("button").click(function(){  $(".test").hide();  });  });  </script>  </head>  <body>  <h2 class="test">This is a heading</h2>  <p class="test">This is a paragraph.</p>  <p>This is another paragraph.</p>  <button>Click me</button>  </body>  </html> |

**1.4 jQuery Event**

All the different visitor's actions that a web page can respond to are called events.

An event represents the precise moment when something happens.

Examples:

* moving a mouse over an element
* selecting a radio button
* clicking on an element

|  |  |  |  |
| --- | --- | --- | --- |
| **Mouse Events** | **Keyboard Events** | **Form Events** | **Document/Window Events** |
| click | keypress | submit | load |
| dblclick | keydown | change | resize |
| mouseenter | keyup | focus | scroll |
| mouseleave |  | blur | unload |

**1.4.1 Commonly Used jQuery Event Methods**

**$(document).ready()**

The $(document).ready() method allows us to execute a function when the document is fully loaded.

**click()**

The click() method attaches an event handler function to an HTML element. The function is executed when the user clicks on the HTML element.

**dblclick()**

The dblclick() method attaches an event handler function to an HTML element. The function is executed when the user double-clicks on the HTML element.

**mouseenter()**

The mouseenter() method attaches an event handler function to an HTML element. The function is executed when the mouse pointer enters the HTML element.

**mouseleave()**

The mouseleave() method attaches an event handler function to an HTML element. The function is executed when the mouse pointer leaves the HTML element.

**mousedown()**

The mousedown() method attaches an event handler function to an HTML element. The function is executed, when the left, middle or right mouse button is pressed down, while the mouse is over the HTML element.

**mouseup()**

The mouseup() method attaches an event handler function to an HTML element. The function is executed, when the left, middle or right mouse button is released, while the mouse is over the HTML element.

**hover()**

The hover() method takes two functions and is a combination of the mouseenter() and mouseleave() methods. The first function is executed when the mouse enters the HTML element, and the second function is executed when the mouse leaves the HTML element.

|  |
| --- |
| <!DOCTYPE html>  <html>  <head>  <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.2.1/jquery.min.js"></script>  <script>  $(document).ready(function(){  $("#p1").hover(function(){  alert("You entered p1!");  },  function(){  alert("Bye! You now leave p1!");  });  });  </script>  </head>  <body>  <p id="p1">This is a paragraph.</p>  </body>  </html> |

**1.5 jQuery Effects**

Hide, Show, Toggle, Slide, Fade, and Animate. WOW!

**1.5.1 jQuery hide() and show()**

With jQuery, you can hide and show HTML elements with the hide() and show() methods.

|  |
| --- |
| <html>  <head>  <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.2.1/jquery.min.js"></script>  <script>  $(document).ready(function(){  $("#hide").click(function(){  $("p").hide();  });  $("#show").click(function(){  $("p").show();  });  });  </script>  </head>  <body>  <p>If you click on the "Hide" button, I will disappear.</p>  <button id="hide">Hide</button>  <button id="show">Show</button>  </body>  </html> |

**1.5.2 Fading**

With jQuery you can fade an element in and out of visibility. jQuery has the following fade methods:

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

|  |
| --- |
| <html>  <head>  <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.2.1/jquery.min.js"></script>  <script>  $(document).ready(function(){  $("button").click(function(){  $("#div1").fadeToggle();  $("#div2").fadeToggle("slow");  $("#div3").fadeToggle(3000);  });  });  </script>  </head>  <body>  <p>Demonstrate fadeToggle() with different speed parameters.</p>  <button>Click to fade in/out boxes</button><br><br>  <div id="div1" style="width:80px;height:80px;background-color:red;"></div>  <br>  <div id="div2" style="width:80px;height:80px;background-color:green;"></div>  <br>  <div id="div3" style="width:80px;height:80px;background-color:blue;"></div>  </body>  </html> |

**1.5.3 Sliding**

With jQuery you can create a sliding effect on elements. jQuery has the following slide methods:

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

|  |
| --- |
| <html>  <head>  <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.2.1/jquery.min.js"></script>  <script>  $(document).ready(function(){  $("#flip").click(function(){  $("#panel").slideToggle("slow");  });  });  </script>  <style>  #panel, #flip  {  padding: 5px;  text-align: center;  background-color: #e5eecc;  border: solid 1px #c3c3c3;  }  #panel  {  padding: 50px;  display: none;  }  </style>  </head>  <body>  <div id="flip">Click to slide the panel down or up</div>  <div id="panel">Hello world!</div>  </body>  </html> |

**1.5.4 Animation**

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.

|  |
| --- |
| <!DOCTYPE html>  <html>  <head>  <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.2.1/jquery.min.js"></script>  <script>  $(document).ready(function(){  $("button").click(function(){  var div = $("div");  div.animate({height: '300px', opacity: '0.4'}, "slow");  div.animate({width: '300px', opacity: '0.8'}, "slow");  div.animate({height: '100px', opacity: '0.4'}, "slow");  div.animate({width: '100px', opacity: '0.8'}, "slow");  });  });  </script>  </head>  <body>  <button>Start Animation</button>  <p>By default, all HTML elements have a static position, and cannot be moved. To manipulate the position, remember to first set the CSS position property of the element to relative, fixed, or absolute!</p>  <div style="background:#98bf21;height:100px;width:100px;position:absolute;">  </div>  </body>  </html> |

**1.6 jQuery - Add Elements**

Four jQuery methods that are used to add new content:

* 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

|  |
| --- |
| <html>  <head>  <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.2.1/jquery.min.js"></script>  <script>  $(document).ready(function(){  $("#btn1").click(function(){  $("p").append(" <b>Appended text</b>.");  });  $("#btn2").click(function(){  $("ol").append("<li>Appended item</li>");  });  });  </script>  </head>  <body>  <p>This is a paragraph.</p>  <p>This is another paragraph.</p>  <ol>  <li>List item 1</li>  <li>List item 2</li>  <li>List item 3</li>  </ol>  <button id="btn1">Append text</button>  <button id="btn2">Append list items</button>  </body>  </html> |

**1.7 jQuery - Remove Elements**

To remove elements and content, there are mainly two jQuery methods:

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

|  |
| --- |
| <html>  <head>  <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.2.1/jquery.min.js"></script>  <script>  $(document).ready(function(){  $("button").click(function(){  ***$("#div1").remove();***  });  });  </script>  </head>  <body>  <div id="div1" style="height:100px;width:300px;border:1px solid black;background-color:yellow;">  This is some text in the div.  <p>This is a paragraph in the div.</p>  <p>This is another paragraph in the div.</p>  </div>  <br>  <button>Remove div element</button>  </body>  </html> |

|  |
| --- |
| <html>  <head>  <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.2.1/jquery.min.js"></script>  <script>  $(document).ready(function(){  $("button").click(function(){  ***$("#div1").empty();***  });  });  </script>  </head>  <body>  <div id="div1" style="height:100px;width:300px;border:1px solid black;background-color:yellow;">  This is some text in the div.  <p>This is a paragraph in the div.</p>  <p>This is another paragraph in the div.</p>  </div>  <br>  <button>Empty the div element</button>  </body>  </html> |

**11. JSON**

JSON stands for **J**ava**S**cript **O**bject **N**otation

JSON is a lightweight data interchange format

JSON is language independent **\***

JSON is "self-describing" and easy to understand

**JSON Syntax Rules**

Data is in name/value pairs *("firstName":"John")*

Data is separated by commas

Curly braces hold objects *{"firstName":"John", "lastName":"Doe"}*

Square brackets hold arrays *"employees":[  {},  {},  {} ]*

**JSON Example**

{  
"employees":[  
  {"firstName":"John", "lastName":"Doe"},  
  {"firstName":"Anna", "lastName":"Smith"},  
  {"firstName":"Peter", "lastName":"Jones"}  
]  
}

**Converting a JSON Text to a JavaScript Object**

A common use of JSON is to read data from a web server, and display the data in a web page.

For simplicity, this can be demonstrated using a string as input.

First, create a JavaScript string containing JSON syntax

var text = ' { "employees" : [' +  
'{ "firstName":"John" , "lastName":"Doe" },' +  
'{ "firstName":"Anna" , "lastName":"Smith" },' +  
'{ "firstName":"Peter" , "lastName":"Jones" } ]}';

Then, use the JavaScript built-in function JSON.parse() to convert the string into a JavaScript object:

var obj = JSON.parse(text);

Finally, use the new JavaScript object in your page:

|  |
| --- |
| <!DOCTYPE html>  <html>  <body>  <h2>Create Object from JSON String</h2>  <p id="demo"></p>  <script>  var text = '{"employees":[' +  '{"firstName":"John","lastName":"Doe" },' +  '{"firstName":"Anna","lastName":"Smith" },' +  '{"firstName":"Peter","lastName":"Jones" }]}';  obj = JSON.parse(text);  document.getElementById("demo").innerHTML =  obj.employees[1].firstName + " " + obj.employees[1].lastName;  </script>  </body>  </html> |