PHP, AJAX and jQuery

INTRODUCTION

- Before you start studying jQuery, you should have a basic knowledge of:
  - HTML
  - CSS
  - JavaScript

INTRODUCTION

- *jQuery* is a fast, lightweight and feature-rich JavaScript library created by John Resig at BarCamp NYC and released to the Internet in early 2006.
- It is currently headed by Timmy Wilson and maintained by a team of developers.
- It's free and open source software dual-licensed under the Massachusetts Institute of Technology (MIT) and GNU General Public License.
- It offers an easy-to-use library that simplifies JavaScript.
- You can make Document Object Model (DOM) element selections, modify and manipulate CSS, animate elements, and work with Ajax.
- All this functionality is available through a single JavaScript file you can download from the jQuery site

What is jQuery?

- jQuery is a, "write less, do more", JavaScript library.
- 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.
- The jQuery library contains the following features:
  - HTML/DOM manipulation
  - CSS manipulation
  - HTML event methods
  - Effects and animations
  - AJAX
  - And many more

Why jQuery?

- JQuery is a great tool which provides a rich set of AJAX methods to develop next generation web application.
- There are lots of other JavaScript frameworks out there, but jQuery seems to be the most popular, and also the most extendable.
- With JavaScript, you often need to write many lines of codes to locate an element of an HTML document, but with jQuery robust selector mechanism you can easily get the exact portion of the document to perform any manipulation.

ADVANTAGES

- The biggest advantage of jQuery comes from its selectors that allow you to traverse the DOM tree of an HTML document's structure in an efficient manner.
- Additionally, using the jQuery inbuilt methods you can create animations and effects like sliding transition, showing or hiding an element, etc. with a single line of code.
- **Save lots of time** — You can save lots of time and efforts by using the jQuery inbuilt effects and selectors and concentrate on other development work.
- **Easy to use** — jQuery is very easy to use. Anybody with the basic working knowledge of HTML, CSS and JavaScript can start development with jQuery.

ADVANTAGES

- **Simplify common JavaScript tasks** — jQuery considerably simplifies the common JavaScript tasks.
  - Now you can easily create feature rich and interactive web pages with fewer lines of codes.
  - A typical example is retrieving the information from a server and updates the page without refreshing.
- **Compatible with browsers** — jQuery is created with modern browsers in mind and it is compatible with all major modern browsers such as Mozilla Firefox, Google Chrome, Safari, Internet Explorer, and Opera.
- **Absolutely Free** — And the best part is, it is completely free to download and use.

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.
- 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-1.11.2.min.js"></script>
  </head>

jQuery CDN

- If you don't want to download and host jQuery yourself, you can include it from a CDN (Content Delivery Network).
- Both Google and Microsoft host jQuery.
- To use jQuery from Google or Microsoft, use one of the following:
- **Google CDN:**
- <head>
  <script
  src="http://ajax.googleapis.com/ajax/libs/jquery/1.11.2/jquery.min.
  js"> </script>
  </head>
- **Microsoft CDN:**
- <head>
  <script src="http://ajax.aspnetcdn.com/ajax/jQuery/jquery-
  1.11.2.min.js"></script>
  </head>

jQuery CDN

- **One big advantage of using the hosted jQuery from Google or Microsoft:**

- Many users already have downloaded jQuery from Google or Microsoft when visiting another site.
- As a result, it will be loaded from cache when they visit your site, which leads to faster loading time.
- Also, most CDN's will make sure that once a user requests a file from it, it will be served from the server closest to them, which also leads to faster loading time.

jQuery Syntax

- A jQuery statement typically starts with the dollar sign ($) and ends with a semicolon (;).
- In jQuery, the dollar sign ($) is just an alias for jQuery. Let's consider the following example code which demonstrates the most basic statement of the jQuery. E.g.

```
<script type="text/javascript">
    $(document).ready(function(){
    // Some code to be executed...
    alert("Hello World!");
    });
</script>
```

- The $(document).ready(*handler)* is typically known as ready event. Where the *handler* is a function passed to the ready() method to be executed safely as soon as the document is ready to be manipulated i.e. when the DOM hierarchy has been fully constructed.

jQuery Syntax

- The jQuery ready() method is typically used with an anonymous function. So, the above example can also be written in a shorthand notation like this:

```
<script type="text/javascript">
    $(function(){
    // Some code to be executed...
    alert("Hello World!");
    });
</script>
```

- Inside an event handler function you can write the jQuery statements to perform any action following the basic syntax, like: $(selector).action();

jQuery Syntax

- With jQuery you select (query) HTML elements and perform "actions" on them.
- 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".
- jQuery uses CSS syntax(**CSS selectors**) to select elements.

The Document Ready Event

- $(document).ready(function(){
  // jQuery methods go here...
  });
- You should place the jQuery code inside the document ready event so that your code executes when the document is ready to be worked on.
- This is to prevent any jQuery code from running before the document is finished loading (is ready).
- 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 Selectors

- Selecting the elements through a typical JavaScript approach could be very painful.
- The ability of making the DOM elements selection simple and easy is one of the most powerful feature of the jQuery.
- jQuery selectors are used to "find" (or select) HTML elements based on their id, classes, types, attributes, values of attributes and much more.
- It's based on the existing CSS Selectors, and in addition, it has some own custom selectors.
- All selectors in jQuery start with the dollar sign and parentheses: $().

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")
  - jQuery1.php
  - jQuery2.php

- Note: The jQuery supports almost all the selectors defined in the latest CSS3 specifications, as well as it has its own custom selectors. These custom selectors greatly enhance the capabilities selecting the HTML elements on a page.

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")
  - jquery3.php

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") : jQuery4.php
- **Some More Controls**
- $("*") : Selects all elements : jQuery5.php
- $(this)" : Selects the current HTML element : : jQuery6.php
- $("p.intro") : Selects all <p> elements with class="intro": j Query7.php
- $("p:first") : Selects the first <p> element: jQuery8.php
- $("ul li:first"): Selects the first <li> element of the first <ul>: jQuery9.php
- $("ul li:first-child"): Selects the first <li> element of every <ul>: jQuery10.php

Some more controls

- **Selecting Elements by Compound CSS Selector**
- You can also combine the CSS selectors to make your selection even more precise.
- For instance, you can combine the class selector with an element selector to find the elements in a document that has certain type and class.
- E.g. $("[href]"): Selects all elements with an href attribute: jQuery11.php
- $("a[target='_blank']"): Selects all <a> elements with a target attribute value equal to "_blank“: jQuery13.php
- $("a[target!='_blank']"): Selects all <a> elements with a target attribute value NOT equal to "_blank“: jQuery12.php

Some examples

- **jQuery Custom Selector**
- In addition to the CSS defined selectors, jQuery provides its own custom selector to further enhancing the capabilities of selecting elements on a page.
- $(":button"): Selects all <button> elements and <input> elements of type="button“: jQuery14.php
- $("tr:even"): Selects all even <tr> elements: jQuery15.php
- $("tr:odd"): Selects all odd <tr> elements: jQuery16.php
- compundSelect.php, customSelect.php

jQuery / DOM comparison

DOM method	jQuery equivalent
getElementById("id")	$("#id")
getElementsByTagName("tag")	$("tag")
getElementsByName("somename")	$("[name='somename']")
querySelector("selector")	$("selector")
querySelectorAll("selector")	$("selector")

jQuery Event Methods

- Events are often triggered by the user's interaction with the web page, such as:
  - When a link or button is clicked,
  - Text is entered into an input box or textarea,
  - Selection is made in a select box,
  - Key is pressed on the keyboard,
  - The mouse pointer is moved etc.
- In some cases, the Browser itself can trigger the events, such as the page load and unload events.

jQuery Event Methods

- An event represents the precise moment when something happens. E.g.
  - moving a mouse over an element, selecting a radio button, clicking on an element
- The term **"fires"** is often used with events. Example: "The keypress event fires the moment you press a key".

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

jQuery Syntax For Event Methods

- To assign a click event to all paragraphs on a page, you can do this:
  - $("p").click();
- The next step is to define what should happen when the event fires. You must pass a function to the event:
- $("p").click(function(){ // action goes here!! });
- **Commonly Used jQuery Event Methods**
- **$(document).ready()**
- The $(document).ready() method allows us to execute a function when the document is fully loaded.

jQuery mouse Methods

- **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. dblclick.php
- **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. mouseEnter.php

jQuery mouse Methods

- **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. mouseEnter.php
- **mousedown()**
- The mousedown() method attaches an event handler function to an HTML element.
- The function is executed, when the left 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 mouse button is released, while the mouse is over the HTML element.

jQuery mouse Methods

- **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. hover.php

jQuery Keyboard Methods

- A keyboard event is fired when the user press or release a key on the keyboard.
- **The keypress() Method**
- The jQuery keypress() method attach an event handler function to the selected elements (typically form controls) that is executed when the browser receives keyboard input from the user.
- **The keydown() Method**
- The jQuery keydown() method attach an event handler function to the selected elements (typically form controls) that is executed when the user first presses a key on the keyboard.
- **The keyup() Method**
- The jQuery keyup() method attach an event handler function to the selected elements (typically form controls) that is executed when the user releases a key on the keyboard.

jQuery Form Methods

- **focus()**
- The focus() method attaches an event handler function to an HTML form field. The function is executed when the form field gets focus. focus.php
- **blur()**
- The blur() method attaches an event handler function to an HTML form field. The function is executed when the form field loses focus. blurfocus.php
- **The change() Method**
- The jQuery change() method attach an event handler function to the <input>, <textarea> and <select> elements that is executed when its value changes. formChange.php
- **The submit() Method**
- The jQuery submit() method attach an event handler function to the <form> elements that is executed when the user is attempting to submit a form. submitForm.php

jQuery Effects - Hide and Show

- **Syntax:**
  - $(*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.
- **jQuery toggle()**
- With jQuery, you can toggle between the hide() and show() methods with the toggle() method.
- **Syntax:**
  - $(*selector*).toggle(*speed, callback*);
- Shown elements are hidden and hidden elements are shown.
- hideshow.php, hidePartText.php, hideSpeed.php

jQuery Effects - Fading

- jQuery fadeIn() Method
- The jQuery fadeIn() method is used to fade in a hidden element.
- **Syntax:**
  - $(*selector*).fadeIn (*speed, callback*);
- jQuery fadeOut() Method
- The jQuery fadeOut() method is used to fade out a visible element.
- **Syntax:**
  - $(*selector*).fadeOut (*speed,callback*);
- jQuery fadeToggle() Method
- The jQuery fadeToggle() method toggles between the fadeIn() and fadeOut() methods.
- **Syntax:**
  - $(*selector*).fadeToggle (*speed, callback*);

jQuery Effects - Fading

- jQuery fadeTo() Method
- The jQuery fadeTo() method allows fading to a given opacity (value between 0 and 1).
- The required opacity parameter specifies the final opacity of the target elements that can be a number between 0 and 1.
- The duration or speed parameter is also required for this method that specifies the duration of the fade to animation.
- **Syntax:**
  - $(*selector*).fadeTo (*speed, opacity, callback*);
- The required opacity parameter in the fadeTo() method specifies fading to a given opacity (value between 0 and 1).
- fadeIn.php, fadeOut.php, fadeTo.php, fadeToggle.php

jQuery Sliding Methods

- With jQuery you can create a sliding effect on elements.
- The jQuery slideUp() and slideDown() methods is used to hide or show the HTML elements by gradually decreasing or increasing their height (i.e. by sliding them up or down).
- **jQuery slideDown() Method**
- The jQuery slideDown() method is used to slide down an element.
- **Syntax:**
  - $(*selector*).slideDown (*speed, callback*);
- **jQuery slideUp() Method**
- The jQuery slideUp() method is used to slide up an element.
- **Syntax:**
  - $(*selector*).slideUp(*speed, callback*);

jQuery Sliding Methods

- **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*);
- slideUp.php, slideDown.php
- slideUp.php, slideDown.php, slideToggle.php

jQuery Effects - Animation

- **The animate() Method**
- The jQuery animate() method is used to create custom animations.
- The animate() method is typically used to animate numeric CSS properties, for example, width, height, margin, padding, opacity, top, left, etc.
- But the non-numeric properties such as color or background-color cannot be animated using the basic jQuery functionality.

jQuery Effects - Animation

- **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. Higher values indicate slower animations.
- The optional callback parameter is a function to be executed after the animation completes.

jQuery Effects - Animation

- **Manipulate Multiple Properties**
- There is one important thing to remember: all property names must be camel-cased when used with the animate() method:
  - You will need to write paddingLeft instead of padding-left, marginRight instead of margin-right, and so on.
- **Using Relative Values**
- It is also possible to define relative values (the value is then relative to the element's current value).
- This is done by putting += or -= in front of the value.

jQuery Effects - Animation

- **Using Pre-defined Values**
- You can even specify a property's animation value as "show", "hide", or "toggle“.
- **Using Queue Functionality**
- By default, jQuery comes with queue functionality for animations.
- This means that if you write multiple animate() calls after each other, jQuery creates an "internal" queue with these method calls. Then it runs the animate calls ONE by ONE.

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*);**

jQuery Method Chaining

- Chaining, allows to run multiple jQuery commands, one after the other, on the same element(s).
- To chain an action, you simply append the action to the previous action.
- chaining1.php, chaining2.php

jQuery - AJAX load() Method

- The jQuery load() method is a simple, but powerful AJAX method.
- The load() method loads data from a server and puts the returned data into the selected element.
- **Syntax:**
  - $(*selector*).load(*URL, data, callback*);
- The required URL parameter specifies the URL you wish to load.
- The optional data parameter specifies a set of querystring key/value pairs to send along with the request.
- The optional callback parameter is the name of a function to be executed after the load() method is completed.
- load.php

jQuery - AJAX load() Method

- It is also possible to add a jQuery selector to the URL parameter. (loadfn3.php)
- The optional callback parameter specifies a callback function to run when the load() method is completed. The callback function can have different parameters:
  - responseTxt - contains the resulting content if the call succeeds
  - statusTxt - contains the status of the call
  - xhr - contains the XMLHttpRequest object
- (loadfn2.php)