Unit 2: HTML5, JQuery and Ajax

jQuery

jQuery is a JavaScript Library. jQuery simplifies JavaScript programming. jQuery is a lightweight, "write less, do more", and JavaScript library. The purpose of jQuery is to make it much easier to use JavaScript on website. jQuery was originally released in January 2006 at BarCampNYC byJohn Resig.

jQuery is a JavaScript library that allows web developers to add extra functionality to their websites. It is open source and provided for free under the MIT license. In recent years, jQuery has become the most popular JavaScript library used in web development.

With jQuery you select (query) HTML elements and perform "actions" on them. Basic syntax is:

$(selector).action ()

- A $ sign to define/ access jQuery.
- “$()” access an element in current html document.

A (selector) to "query (or find)" HTML elements.

A jQuery action () to be performed on the element(s).

Examples:

- $("p").hide() - hides all elements.
- $("#test").hide() - hides the element with id="test".
- $(".test").hide() - hides all elements with class="test".

JavaScript vs. jQuery

Let us try to understand the difference between JavaScript and jquery.

Example 1 - Hide an element with id "textbox“

```
//JavaScript
 document.getElementById(textbox).style.display = "none";
//jQuery
$(#textbox).hide();
```

Example 2 - Create a <h1> tag with "my text“

```
//JavaScript
 var h1 = document.CreateElement("h1");
 h1.innerHTML = "my text";
document.getElementsByTagName(body)[0].appendChild(h1);
//jQuery
$(body).append( $("<h1/>").html("my text") ;
```

The 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).

- Trying to hide an element that is not created yet.
- Trying to get the size of an image that is not loaded yet

Alternate Syntax :

```
$(function(){

// jQuery methods go here...

}
```

jquery Selectors:

jQuery selectors allow you to select and manipulate HTML With jQuery selectors you can find 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 type of selectors in jQuery, start with the dollar sign and parentheses: $().

Types of jquery selectors .

- Element selector Id
- (#) selector Class
- (.) selector

Element Selector:

The jQuery element selector selects elements based on their tag names. Example:

```
$(document).ready(function()
{
    $("#button").click(function()
   {
       $("p").hide();
    });
});
```

Id (#) Selector :

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

Example :

```
$(document).ready(function()
{
        $("#button").click(function()
        {
            $("#test").hide();
        });
});
```

Class (.) Selector:

The jQuery class selector finds elements with a specific class.

Example :

```
$(document).ready(function()
{
        $("#button").click(function()
        {
            $(".test").hide();
        });
});
```

More jquery Selectors :

Syntax	Description
$("*")	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 element of the first
$("ul li:first-child")	Selects the first element of every
$("[href]")	Selects all elements with an href attribute
$("a[target='_blank']")	Selects all <a> elements with a target attribute value equal to "_blank"
$("tr:even")	Selects all even <tr> elements
$("tr:odd")	Selects all odd <tr> elements

jquery Effects :

There are 3 types of jQuery Effects and they are:

- jQuery hide()
- jQuery show()
- jQuery toggle()

Syntax:

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

$(selector)show(speed, callback);

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

Speed and callback are optional parameters

jQueryEvent Methods:

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 Terminology:

- The **jQuery function** refers to the global jQuery object or the $ function depending on the context .
- A **jQuery object** the object returned by the jQuery function that often represents a group of elements.
- **Selected elements** refers to the DOM elements that you have selected for, most likely by some CSS selector passed to the jQuery function and possibly later filtered further.

jQuery Methods

1. DOM Manipulation

- before(), after(), append(), appendTo()

Example: Move all paragraphs in div with id "contents"

```
$("p").appendTo("#contents");
        $("h1").append(" Dom Manipulation");
       <body>
        <h1>jQuery Dom Manipulation</h1>
       <div id="contents">
              <p>jQuery is good</p>
               <p>jQuery is better</p>
              <p>jQuery is the best</p>
        </div> </body>
```

2. Attributes

- css(), addClass(), attr(), html(), val()

Example : Setting

```
$("img.logo").attr("align", "left");
$("p.copyright").html("&copy; 2009 ajaxray");
$("input#name").val("Spiderman");
```

Example : Getting

```
var allignment = $("img.logo").attr("align");
var copyright = $("p.copyright").html();
var username = $("input#name").val();
```

3. Events

- click(), bind(), unbind(), live()

Example: Binding all interactions on events.

```
$(document).ready(function(){
$("#message").click(function(){
        $(this).hide(); })
});
<span id="message" onclick="…"> blah blah </span>
```

4. Effects

- hide(), fadeOut(), toggle(), animate()

Example :When "show-cart" link clicked, slide up/down "cart" div.

```
$("a#show-cart").click(function(){
$("#cart").slideToggle("slow"); })
```

5. Ajax

- load(), get(), ajax(), getJSON()

Examples: Load a page in a container

$("#comments").load ("/get_comments.php"); $("#comments").load ("/get_comments.php", {max: 5});

jQuery Method Chaining

Chaining Methods, also known as Cascading, refers to repeatedly calling one method after another on an object, in one continuous line of code. This technique abounds in jQuery and other JavaScript libraries and it is even common in some JavaScript native methods.

Example:

```
$("#wrapper").fadeOut().html("Welcome, Sir").fadeIn();
or this:
str.replace("k", "R").toUpperCase().substr(0,4);
```

is not just pleasurable and convenient but also succinct and intelligible. It allows us to read code like a sentence, flowing gracefully across the page. It also frees us from the monotonous, blocky structures we usually construct.

jQuery chaining allows you to execute multiple methods in a single statement. By doing that, it removes the need for repeatedly finding the same element to execute code. It also makes the code more compact and readable.

To perform jQuery method chaining, you should append actions to one another. Usually each statement is run as a separate operation. Chaining in jQuery is used to link multiple statements together. A chained jQuery statement is executed as one operation. Therefore, it runs faster.