jQuery

INTRODUCTION TO JQUERY

jQuery is a fast and concise JavaScript Library that simplifies HTML document traversing, event handling, animating, and Ajax interactions for rapid web development.

JQuery download (http://jQuery.com/)

Examples of JQuery Websites:

1. Google also using jQuery (www.google.com).
2. Entertainment Arts uses jQuery (www.ea.com).
3. Wordpress (wordpress.org)
4. Twitter (http://twitter.com)

WHAT IS JQUERY

1. jQuery is a free available, open source JavaScript library.
2. jQuery is used in creating highly responsive web pages.
3. Works across all modern browsers.
4. Abstracts away browser-specific features, allowing you to concentrate on design.
5. JQuery focuses on simplifying common scripting tasks
   i) Getting and manipulating page content. (DOM: document object model).
   ii) Working with modern browser event model (key task).
   iii) Adding sophisticated transition effects like fading, animation etc.
6. JavaScript has become essential to current web page development, but.., JavaScript is not a good language design
7. JavaScript has become bloated
   i) DOM navigation (css-like syntax)
   ii) Apply methods to sets of DOM elements
   iii) Builder model (chain method calls)
   iv) Extensible and there are tons of libraries
   v) Handles most browser differences
8. Writing JavaScript code is tedious, time-consuming, and error-prone
9. jQuery makes writing JavaScript much easier

BASIC JQUERY SELECTORS (SIMILAR TO CASCADING STYLE SHEETS)

SELECTOR	PURPOSE
Tag Name	Finds All Elements That Are Named Tag Name
#identifier	Finds All Elements with ID of identifier
.classname	Finds All Elements that have class attribute with the value of class name
tag.classname	Gets elements of type tag that have a class attribute with the value of a class name
tag#id.classname	Retrieves the tag element that has an ID of id and a class attribute with the value of class name
*	Finds all elements in the page

JQUERY (VS) DOM(JAVA SCRIPT)

```
<html>
<head>
<title>Document</title>
<script type="text/javascript" src="jQuery.js"></script>
<script type="text/javascript">
  $("document").ready(function() {
        });
</script>
<style type="text/css">
.a { color: Navy; }
.b { color: Maroon; }
</style>
</head>
<body>
        <ul id="list1">
                <li class="a">item 1</li>
                <li class="a">item 2</li>
                <li class="b">item 3</li>
                <li class="b">item 4</li>
        </ul>
<p class="a">This is paragraph 1</p>
<p>This is paragraph 2</p>
<p class="b">This is paragraph 3</p>
<p>This is paragraph 4</p>
</body>
</html>
```

The **hierarchy** and **combination** selectors allow you to get a little more advanced in selecting page content. You can select elements based on hierarchical relationships or on a series of common criteria.

SELECTOR	PURPOSE
Selector, Selector....	Finds all of the specified selectors
.class1.class2	Finds all elements with both .class1 and .class2 applied
parent>child	Finds all child elements that direct children of elements of type parent
anscestor descendant	Finds all descendant elements that are contained within elements of type anscestor
prev + next	Finds all next elements that are next to a previous elements
prev ~ siblings	Finds all sibling elements that come after prev and match siblings selector

JQUERY FILTERS

Filters work with selectors to provide even more fine-grained control over how elements are selected in the document like position or index.

FILTER	PURPOSE

:first	Selects only the first instance of the selector's returned set
:last	Selects only the first instance of the selector's returned set
:even	Selects only the even-numbered elements in the selector's returned set
:odd	Selects only the odd-numbered elements in the selector's returned set
:eq(n)	Filters out elements that are not positioned at the given index
:lt(n)	Includes elements that are past the given index
:gt(n)	Includes elements that are before the given index
:header	Selects all header elements (h1, h2, h3 etc.)
:animated	Selects all elements that are currently being animated in same way
:not(selector)	Includes elements that do not match the given selector

JQUERY ATTRIBUTES

You can filter the results of a selector statement based on attribute content.

FILTER	PURPOSE
[attribute]	Includes elements in the result set if they have the specified attribute
[attribute=value]	Includes elements in the result set if they have the specified attribute and it has the given value
[attribute!=value]	Includes elements in the result set only if they have the specified attribute and it doesn't have the given value
[attribute^=value]	Includes elements that have the specified attribute and it starts with the specified value
[attribute$=value]	Includes elements that have the specified attribute and it ends with the specified value
[attribute*=value]	Includes elements that have the specified attribute and it contains the specified value
[attrFilter1][attrFilterN]	Includes elements that match all of the specified attribute filters

CONTENT FILTERS

You can examine the content of selected elements or their visibility property to determine whether they should be included or excluded from the final set.

CONTENT FILTER	PURPOSE
:contains(text)	Filters the selection to only include elements that contains the text string
:empty	Filters the selection to only include empty elements
:has(selector)	Matches elements that contains at least one element that has the specified selector
:parent	Matches all elements that are parents(i.e. they contain at least one other element, including text)

VISIBILITY FILTERS

VISIBILITY SELECTORS	PURPOSE
:visible	Filters the selection to only include visible elements
:hidden	Filters the selection to only include hidden elements

CHILD FILTERS

FILTER	PURPOSE
:nth-child(index)	index starts from 1
:nth-child(even)	Gets only event children
:nth-child(odd)	Gets only odd children
:first-child	Matches elements who are the first child of their parent
:last-child	Matches elements who are the last child of their parent
:only-child	Matches elements who are the only child of their parent

FORM FILTERS

You can use form selectors to deal with Form elements. They work like other selectors but start with a colon(:) like a regular filter

SELECTOR	PURPOSE
:input	Finds all input, select, textarea, and button elements
:text	Finds all text elements
:password	Finds all password elements
:radio	Finds all radio button elements
:checkbox	Finds all checkbox elements
:submit	Finds all submit elements
:reset	Finds all reset elements
:button	Finds all button elements
:image	Finds all image elements
:file	Finds all file upload elements

You can also perform additional filtering of form elements, such as whether items are checked, selected, or enabled

SELECTOR	PURPOSE
:enabled	Matches all form elements that are enabled
:disabled	Matches all form elements that are disabled
:checked	Matches all form elements that are checked(radio buttons and checkboxes)
:selected	Matches all form elements that are selected

TRAVERSING DOCUMENT INFORMATION

You can traverse the information returned from a document easily

FUNCTION/PROPERTY	PURPOSE
size().length	The number of elements in the jQuery result set
get()	Returns an array of all matched DOM elements. Useful if you need to operate on the DOM elements themselves instead of using built-in jQuery functions
get(index)	Access a single matched DOM element at a specified index in the matched set

find(expression)	Searches for descendant elements that match the specified expression
each(fn)	Execute a function within the context of every matched element

MANIPULATING PAGE CONTENT

Once you have used selectors and filters to retrieve web page content, you usually want to do something with it. Sometimes you want to create new content to dynamically add into the page. jQuery has functions for creating, copying, deleting and moving content around, as well as wrapping page content in other content. jQuery provides cross-browser support for working with css ,including positioning and sizing information.

CREATING, SETTING AND GETTING CONTENT

To create new HTML element, you simply pass a string containing new HTML to the $() function:

```
var mystr=$("<h1>this is a HTML element</h1>");
var mystr="<h1>this is a HTML element</h1>";
var newelement=$(mystr);
```

In addition to this method, you can use the html() and text() methods to get and set content on.

FUNCTION	PURPOSE
html()	Returns the HTML content of the first matched element
html(newcontent)	Sets the HTML content of every matched element
text()	Returns the text content of the first matched element
text(newtext)	Sets the text content for all matched elements

MANIPULATING ATTRIBUTES

To inspect or change the value of attribute on elements, use jQuery's attr functions

FUNCTION	PURPOSE
attr(name)	Accesses property on the first matched element. This method makes it easy to retrieve a property value from the first matched element. If the element doesn't have an attribute with such a name, undefined is returned
attr(properties)	Sets a series of attributes on all matched elements using object notation syntax. This is the best used for setting large numbers of properties at once;
attr(key,value)	Sets a single property to a value on all matched elements
attr(key,fn)	Sets a single property a computed value, on all matched elements. Instead of supplying a string value, a function is provided that computes the value of the attribute
removeAttr(name)	Removes the named attribute from all the matched elements

INSERTING CONTENT

jQuery provides several functions for inserting content in the document, both before and after existing page elements

FUNCTION	PURPOSE
append(content)	Appends content to the inside of every matched element
appendTo(selector)	Appends all of the matched elements to another, specified set of elements
prepend(content)	Prepends content to the inside of every matched element
prependTo(selector)	Prepends all of the matched elements to another, specified set of elements
after(content)	Inserts content after each of the matched elements
before(content)	Inserts content before each of the matched elements
insertAfter(selector)	Inserts all of the matched elements after another, specified set of elements
insertBefore(selector)	Inserts all of the matched elements before another, specified set of elements

WRAPPING, REPLACING, REMOVING CONTENT

jQuery can warp the existing content in the page, replace content, copy content, and remove it

FUNCTION	PURPOSE
wrap(html)	Wraps each matched element with the specified HTML content
wrap(element)	Wraps each matched element with the specified content
wrapAll(html)	Wraps all the elements in the matched set with the specified HTML content
wrapAll(element)	Wraps all the elements in the matched set into a single wrapper element
wrapInner(html)	Wraps the inner child contents of each matched element (including text nodes) with HTML structure
wrapInner(element)	Wraps the inner child contents of each matched element (including text nodes) with an DOM structure
replaceWith(content)	Replaces all matched elements with the specified HTML or DOM elements
replaceAll(selector)	Replaces the elements matched by the specified selector with the matched elements
empty()	Removes all child nodes from the set matched elements
remove()	
	Removes all matched elements from the DOM
clone()	Clone matched DOM elements and selects the clones

WORKING WITH CSS INFORMATION

jQuery's CSS functions provide easy, cross-browser access for setting properties and working with positioning and sizing information. The css function allows you to retrieve and set CSS styles for a set of matched elements.

FILTER	PURPOSE
css(name)	Returns the value for the named CSS property for the first matched element

css(properties)	Sets the CSS properties of every matched element Using an object-notation. Syntax: var cssObj={ 'background-color':'red'} $(this).css(cssObj);
css(property,value)	Sets a single style property to a value on all matched elements. If a number is provided, it is automatically converted into a pixel value, with the following exceptions: z-index, font-weight, opacity, zoom, and line-height.

WORKING WITH CSS CLASSES

jQuery provides a set of functions for working with CSS classes on page elements classes can be easily added, removed, toggled and detected.

CSS FUNCTIONS	PURPOSE
addClass(class)	Adds the specified class(es) to each of the set of matched elements
hasClass(class)	Determine whether any of the matched elements are assigned the given class
removeClass(class)	Remove a single class, multiple classes, or all classes from each element in the set of matched elements
toggleClass(class)	Add or remove one or more classes from each element in the set of matched elements, depending on either the class's presence or the value of the switch argument

WORKING WITH CSS POSITIONING

The CSS positioning functions for provide cross-browser support for figuring out the positions of elements

CSS FUNCTIONS	PURPOSE
offset()	Get the current coordinates of the first element in the set of matched elements, relative to the document
offsetParent()	Get the closest ancestor element that is positioned
position()	Get the current coordinates of the first element in the set of matched elements, relative to the offset parent
scrollTop()	Get the current vertical position of the scroll bar for the first element in the set of matched elements
scrollLeft()	Get the current horizontal position of the scroll bar for the first element in the set of matched elements

WORKING WITH CSS SIZING INFORMATION

To retrieve cross-browser information for elements, use the jQuery size-related functions

CSS FUNCTIONS	PURPOSE
height()	Get the current computed height for the first element in the set of matched elements
width()	Get the current computed width for the first element in the set of matched elements
innerHeight()	Get the current computed height for the first element in the set of matched elements, including padding but not border
innerWidth()	Get the current computed width for the first element in the set of matched elements, including padding but not border
outerWidth()	Get the current computed width for the first element in the set of matched elements, including padding and border
outerHeight()	Get the current computed height for the first element in the set of matched elements, including padding, border, and optionally margin

EVENTS IN JQUERY

Event handling in jQuery is very simple. To handle events, you basically need to associate an event with a function called event handler. Traversing and manipulating DOM objects is easy in jQuery, and the jQuery Event Handling API makes working with events extremely simple.

The Event Handling API in jQuery is organized into a collection of functions. The jQuery event object is normalized and has been wrapped into the jQuery. Event object. The most common jQuery events include blur, change, click, dblclick, error, focus, keydown, keypress, keyup, load, mousedown, mouseenter, mouseleave, mousemove, resize, scroll, select, submit and unload.

These methods are used to register behaviors to take effect when the user interacts with the browser, and to further manipulate those registered behaviors.

JQUERY EVENT FUNCTIONS

Events are connected to or disconnected from elements using the bind() and unbind() functions.

```
$(selector).bind(event,data,handler)
$(selector).unbind(event,handler)
```

BIND():
Event: Defines the event that you want to be bound to for each element in the selector's event result set. Possible values are blur, change, click, dblclick, error, focus, keydown, keypress, keyup, load, mousedown, mouseenter, mouseleave, mousemove, resize, scroll, select,submit and unload.
handler: Specifies the handler function.

Data Optional. Defines a piece of data that will be passed to the handler function when the event happens and the handler function is called handler. Specifies the function that will handle the event

UNBIND():
Event: Defines the event that u want to be disconnected for each element in the selector's result set
handler: Specifies the handler function that was defined to handle the event

CONVENIENT JQUERY HELPER FUNCTIONS

In addition to that bind() and unbind() jQuery provides convenient helper functions
Several "helper" functions can perform common event-related tasks

```
$(selector).click(fn)
$(selector).click(fnOver,fnOut)
$(selector).toggle(fn1,fn2,fn3,fn4..)
```

FUNCTION	PURPOSE
click(fn)	A shortcut for defining a click function handler.There are also shortcuts for : blur, change, click, dblclick, error, focus, keydown, keypress, keyup, load, mousedown, mouseenter, mouseleave, mousemove, resize, scroll, select, submit and unload.
hover(fnOver,fnOut)	Helper function for hover behavior. fnOver is the function to call when the mouse enters, fnOut for when the mouse leaves
toggle(fn1,fn2,fn3..)	Helper function for implementing toggling behavior. jQuery will call each function on every other click, starting with fn1 then fn2,then fn3,etc..

JQUERY EVENT OBJECT

Writing event handling code is frustrating when it differs across browser. The jQuery event object smoothes these differences and provide a single object with the most important properties.

FUNCTION	PURPOSE
type	Type of the event ("click", e.g.)
target	Element that issued the event
data	Data passed to bind function
pageX, pageY	Coordinates of mouse when event happened, relative to document result Value returned by the last handler function
timestamp	Time when event occurred
preventDefault	Prevents the browser from executing the default action
isDefaultPrevented	Returns whether preventDefault() was ever called on this object
stopPropagation()	Stops the bubbling of an event to parent elements
isPropagationStopped()	Returns whether stopPropagation() was ever called on this object

MISCELLANEOUS JQUERY FUNCTIONS

For a specialized tasks, jQuery provides some miscellaneous functions

```
$(selector).one(type,data,handler)
$(selector).trigger(event,data)
$(selector).triggerHandler(event,data)
```

FUNCTION	PURPOSE
one(type,data,handler)	Works the same as bind(),but the event handler is only ever executed one time for each matched element
trigger(event,data)	Triggers an event on every matched element. This will also cause

	the default action of the browser to be executed. For example, passing 'click' to the trigger() will also cause the browser to act as through the item were clicked
triggerHandler (event,data)	Triggers all bound event handlers on an element (for a specific event type) without executing the browser's default actions, bubbling or live events. Only works on the first matched element in the result set for selector