jQuery Fundamentals

Eng. Niveen Nasr El-Den
SD & Gaming CoE
iTi

Day 1

jQuery

- jQuery is a fast and concise JavaScript Library created by John Resig in 2006 at Rochester Institute of Technology with a nice motto:

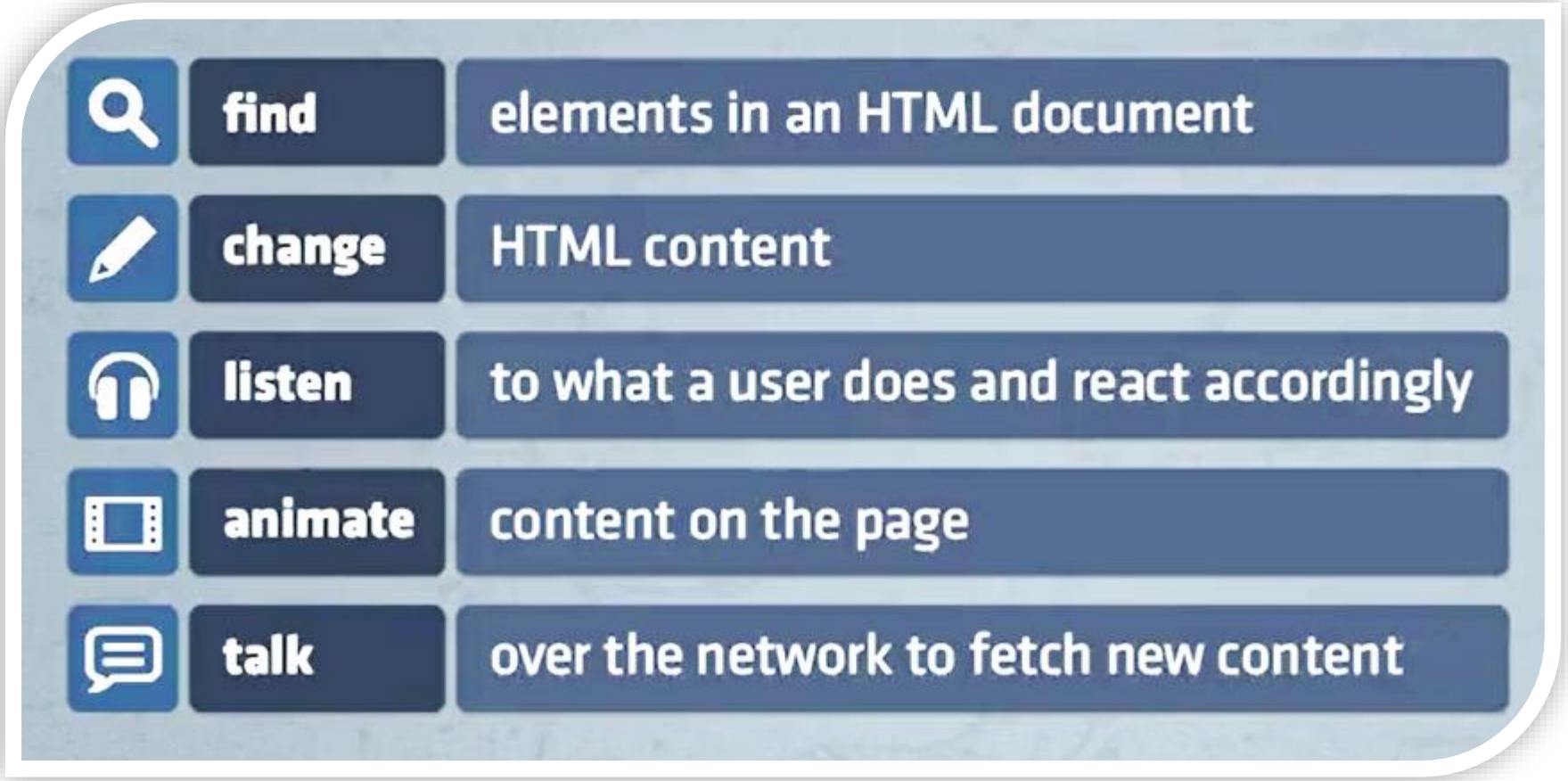
  "Write less, do more."

- It is one of many available libraries that
  - Provide functions for manipulating the web page with fairly good performance
  - Help to keep your JS code clean
- Current release 3.2.1 (March 2017)

jQuery

- Powerful JavaScript library
  - Simplify common JavaScript tasks
  - Access parts of a page
  - Modify the appearance of a page
  - Alter the content of a page
  - Change the user's interaction with a page
  - Add animation to a page
  - Provide AJAX support
  - Abstract away browser quirks

Simply jQuery Makes it easy to:



Why jQuery

- Rich Internet Applications (RIA)
- Dynamic HTML (DHTML)
- Friendly and Elegant API's
- DOM traversing is very easy especially for complex search criteria
- The core library size is very small
- jQuery UI Extension provides very rich controls
- Cross Browser Support

```
$ ('#something').hide().css('background', 'red').fadeIn();
```

Things jQuery Provides

- Select DOM elements on a page
- Set properties of DOM elements, in groups ("Find something, do something with it")
- Creates, deletes, shows, hides DOM elements
- Defines event behavior on a page (click, mouse movement, dynamic styles, animations, dynamic content)
- Reduces browser inconsistencies
- Plugins are available to cover all needs

Using jQuery

- Installation
  - just download the jquery-3.x.x.js file and put it in your website folder get it from "www.jquery.com"
  - Use development version in developing phase
    - Useful for debugging
  - Use production (minified) version for deploying
- Reference the JS file in your HTML
  - preferably insert it within the <head> tag.

```
<script type="text/javascript" src="jquery-3.x.x.min.js">
</script>
```

Using jQuery via CDN

- Google

```
<script type="text/javascript“
src="http://ajax.googleapis.com/ajax/libs/jquery/3.2.1/jquery.min.js">
</script>
```

- Microsoft

```
<script type="text/javascript“
src="http://ajax.microsoft.com/ajax/jquery/jquery-3.2.1.min.js">
</script>
```

Using jQuery via CDN

- jQuery

```
<script type="text/javascript“
        src=“http://code.jquery.com/jquery-3.2.1.min.js">
</script>
```

- Then, access the jQuery functions via the $ object

```
$(document).ready(
        function() {$("div.someClass").show();}
);
```

How jQuery works

```
<head>
<script type="text/javascript" src="jquery.js"></script>

<script type="text/javascript">
$(document).ready(function(){

    $(".button").click(function(){
        $("#panel").slideDown("slow");
    });

});
</script>
</head>
```

Include the jquery file

The "ready event" (Binds a function to be executed whenever the DOM is ready)

This part can be written in an external .js file.

Where do you want to bind the function?
It can be CSS class, ID, Selectors (ie. DIV, H1, A, P, LI...)

This function will be triggered when an element with class="button" is clicked

What would like to do with #panel?
In this case, slide it down with "slow" speed.

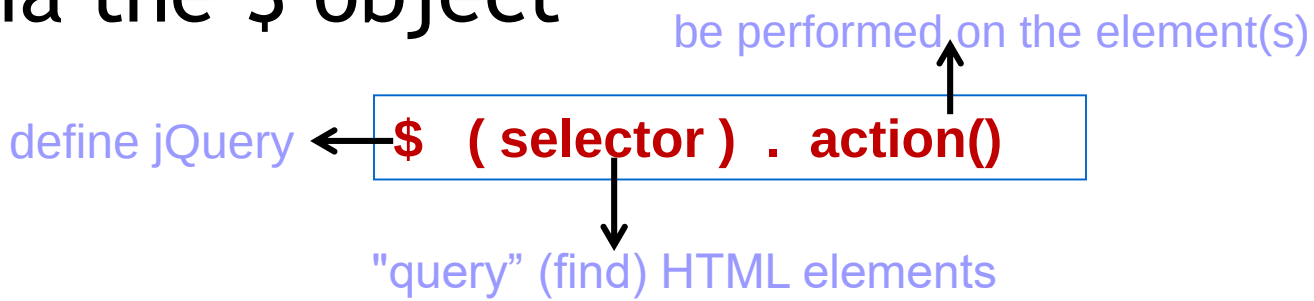
Where do you want to apply this function?
In this case, it is the element with id="panel"

```
$("#panel")
```

The quotation marks can be either single or double.
ie. (".class") or ('.class')

How jQuery works

- Set up a basic HTML page and add jQuery
- Create a “ready” function
- Write functions to tell jQuery what to do via the $ object



- The jQuery syntax is tailor made for finding HTML elements and do some action on the element(s)

The Ready Function

- It checks the document and waits until document is ready to be manipulated

- Many ways to specify the ready function
  - $(document).ready(function(){…};);
  - $('document').ready(function(){…};);
  - $().ready(function(){…};) → not recommended
  - $(function(){…};) → recommended in the latest version
  - etc.

- Generally

  selector.*ready* (func)

  func → function to be executed whenever the DOM is ready to be traversed and manipulated.

jQuery

- jQuery Library supports us with a variety of utilities and features

1. **Selectors**
2. **Attributes**
3. **DOM Manipulation**
4. **Traversing**
5. **CSS**
6. **Events**
7. **UI**
8. **Ajax**
9. **Animation**
10. **Plug-ins**

jQuery Selectors & Filters

jQuery Selector

- jQuery can help you find elements based on their ID, classes, types, attributes, values of attributes and much, much more
- *A jQuery Selector* is a function which makes use of expressions to find out matching elements from a DOM based on the given criteria.
- The jQuery library harnesses the power of *Cascading Style Sheets (CSS) selectors* to let us quickly and easily access elements or groups of elements in the Document Object Model (DOM).
- All type of selectors available in jQuery, always start with the dollar sign and parentheses: *$().*

jQuery Selector

- The factory function **$(,)** makes use of following building blocks while selecting elements in a given document
  - The #id selector
  - The .class selector
  - The element selector
  - etc..
- The jQuery selector takes two parameters
  - selector,
  - context

jQuery Selector

jQuery	Description
Name	Selects all elements which match with the given element Name.
#ID	Selects a single element which matches with the given ID
.Class	Selects all elements which match with the given Class.
Universal (*)	Selects all elements available in a DOM.
Multiple Elements E, F, G	Selects the combined results of all the specified selectors E, F or G.

jQuery Selector

Selector	Result
$('*')	all elements in the document.
$("p > *")	all elements that are children of a paragraph element.
$("#specialID")	the element with id="specialID".
$(".specialClass")	all the elements that have the class of *specialClass*
$("p:empty")	all elements matched by <p> that have no children.
$("p a.specialClass")	matches links with a class of *specialClass* declared within <p> elements.
$("input[name=myname]")	all elements matched by <input> that have a name value exactly equal to *myname*.
$(":radio")	all radio buttons in the form.
$("li:even“)	all elements matched by that have an even index value.

Context parameter

- The context is where the jQuery selector runs

$ (selector , context) = $ (context) . find (selector)

Similar to ➔ $ (context selector)

- Example:
  $(“a”,”div”); Similarly as $(“div a”)
  It means select all anchors (a) but only those in div

- Context parameter is confusing and not worth using

Selecting Elements by Index Order

- Depending on what you want to do, you have the following filters.
- These may look like CSS pseudo-classes, but in jQuery they're called filters

filter	Result
:first	Matches the first selected element
:last	Matches the last selected element
:even	Matches even elements (zero-indexed)
:odd	Matches odd elements (zero-indexed)
:eq(n)	Matches a single element by its index (n)
:lt(n)	all elements matched with an index below n
:gt(n)	all elements matched with an index above n

Attribute Selectors

- Results of a selector statement can be filtered based on attribute contents
  - $('[attr]') element with attr
  - $('[attr="val"]') attr equals val
  - $('[attr!="val"]') attr does not equal val
  - $('[attr~="val"]') attr has val with space-sep.
  - $('[attr^="val"]') attr begins with val
  - $('[attr$="val"]') attr ends with val
  - $('[attr*="val"]') attr contains val
  - $('[attr1][attr2]') element with attr1 and attr2

Form Selectors

Selector	Description
:input	Finds all form elements (input, select, textarea and buttons)
:text	Returns all text elements
:password	Returns all password elements
:radio	Returns all radio button elements
:checkbox	Returns all checkbox elements
:submit	Returns all submit elements
:reset	Returns all reset elements
:image	Returns all image elements
:button	Returns all button elements
:enabled	Returns all enabled form elements
:disabled	Returns all disabled form elements
:checked	Returns all checked form elements
:selected	Returns all selected form elements

CSS Form Selectors

- $(':text') <input type="text">
- $(':password') <input type=" password ">
- $(':radio') <input type="radio">
- $(':button') <input type="button">
- $(':selected') <option selected="selected">
- etc.

:contains(txt) Selector

- Used to select an element based on string it contains
- it’s case sensitive.

- Example:
  $('span:contains("A")');

:not() Selectors

- jQuery gives us the :not filter, which you can use in the following way:
  $('div:not(#content)');
  → Select all DIV elements except #content

- The selector can be as complex as you like:
  $('a:not(div.important a, a.nav)');
  →Selects anchors that do not reside within 'div.important' or have the class 'nav'

Custom Misc. Selectors

Selector	Description
:header	Selects all header elements (h1→h6)
:animated	Selects currently animated elements
:has(sel)	Selects elements that contain at least one matching the passing sel parameter
:visible	Filters the returning selection results with only visible elements
:hidden	Filters the returning selection results with only hidden elements

Other Selectors

- :nth-child(index)
- :first-child
- :last-child
- :only-child
- etc…

filter() Method

- The benefit of using this function is that you can make a combination of more than one filter
- It is used Instead of writing the filter expression within selector methods $()
- Example:
  - $(‘ul li:eq(1)’)
    →is same as ('ul li').filter(':eq(1)');
  - $('ul li').filter(':eq(0),:eq(2)');
    → selecting 1st and 3rd items in an unordered list

filter() Method

- Used to select an element varying characteristics such as programmatic states not expressible as selector expressions.
  - Pass function to filter() method as a parameter instead of a text.

- Example:

```
$('div').filter(function(){
    var width = jQuery(this).width();
    return width > 100 && width < 200;
});
```

→Select all DIV elements with a width between 100px and 200px:

siblings( ) Method

- To select and filter all siblings of an element occurring either next or previous.
- You can pass an optional selector expression to filter the selection
- Example:
  $('h1').siblings('h2,h3,p');
  →Selects all H2, H3, and P elements that are siblings of H1 elements..

next( ) Method

- To make the same functionality of the sibling combinator (+) without using it, you can use the next( ) method.
- The next( ) method can make a nice alternative to the selector syntax, especially in a programmatic setting when you're dealing with jQuery objects as variables.
- Example

```
var topHeaders = jQuery('h1');
topHeaders.next('h2').css('color','red');
```

nextAll( ) Method

- Sometimes you'll want to target siblings dependent on their position relative to other elements
  - for example
    To select all list items beyond the second (after li.selected), you could use the following method
    $('li.selected').nextAll('li');

- The nextAll() method, just like siblings(), accepts a selector expression to filter the selection before it's returned. If you don't pass a selector, then nextAll() will return all siblings of the subject element that exist after the subject element, although not before it.

children( ) method

- Selecting children in a more programmatic environment should be done using jQuery's children() method, to which you can pass a selector to filter the returned elements.

- This would select all direct children of the #content element:

```
$('#content').children();
```

- The preceding code is essentially the same as $('#content > *')

jQuery Attributes Methods

Attributes & Properties

- The basic components we can manipulate when it comes to DOM elements are the ***properties*** and ***attributes*** assigned to those elements.

- Most of these attributes are available through JavaScript as DOM node properties.

- Some of the more common properties are:
  - className
  - tagName
  - id
  - href
  - title
  - scr

Attributes & Properties

- **Getting/setting & removing attributes**
  - **.attr(attr_nm [,val])**
  - **.removeattr(attr_nm)**
- **Getting/setting elements content & values**
  - **.text([“str”])**
  - **.val([“val”])**
  - **.html([“str”])**

Attributes & Properties

- **Styling methods**
  - **.addClass()**
  - **.removeClass()**
  - **.hasClass()**
  - **.toggleClass()**
  - **.css()**

jQuery attr() Method

- Used to either *get* or *set* the value of an attribute from the first element in the matched set or from all matched elements.

```
<img id="myImage" src="image.gif“  title="This is an image"/>
```

- Get an Attribute value → *attr*(“attName”)

```
var imgTitle = $(“img").attr(“title");
```

- Set an Attribute value: → *attr*(name, value)

```
$("#myimg").attr("src", "/images/jquery.jpg");
```

jQuery attr() Method

Attribute Methods that handles *attribute values*

Method	Description
attr(props)	**Set a key/value object as properties to a matched elements.** **e.g. $('img').attr({src:"", title:""…});**
attr(key,fn)	**Set a single property to a computed value, on all matched elements.** **key:** The name of the property to set. **func:** A function returning the value to set
removeAttr(nm)	**Remove an attribute from each of the matched elements** **e.g. $('img').removeAttr(title);**

jQuery element content Methods

jQuery Methods that handles *html and text content*

Method	Description
html()	Get the html contents (innerHTML) of the first matched element.
html(val)	Set the html contents of every matched element.
text()	Get the combined contents of all matched elements.
text(val)	set the text contents of all matched elements

jQuery element content Methods

jQuery Methods that handles *form elements values*

Method	Description
val()	Get the input value of the first matched element
val(val)	Set the value attribute of every matched element if it is called on <input> but if it is called on <select> with the passed <option> value then; passed option would be selected, if it is called on check box or radio box then all the matching check box and adiobox would be checked.

jQuery Styling Methods

Attribute Methods that handles *Style*

Method	Description
addClass(class)	apply defined style sheets onto all the matched elements. You can specify multiple classes separated by space.
removeClass(class)	Removes all or the specified class(es) from the set of matched elements.

jQuery Styling Methods

Attribute Methods that handles *Style*

Method	Description
toggleClass(class)	Adds the specified class if it is not present, removes the specified class if it is present.
hasClass(class)	Returns true if the specified class is present on at least one of the set of matched elements.

css() Method

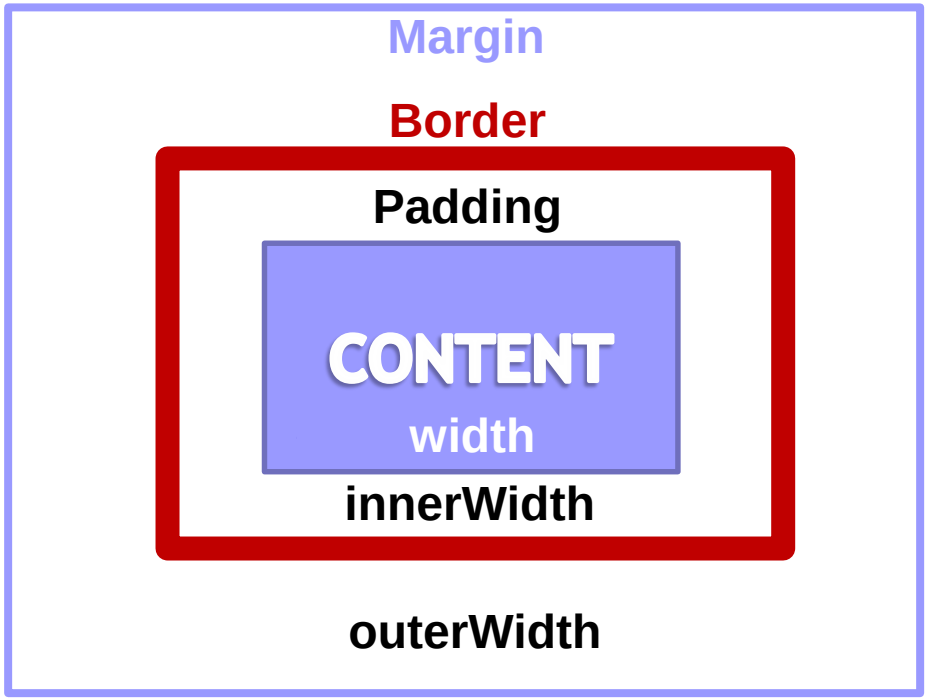
- Like the attr() method, but it works with the CSS styles

Method	Description
css(name)	**Get the value of the css named property for the 1st matched element**
css(prop, val)	**Set the css prop with the given val to all matched elements.** **e.g.** **$("#cssSpan").css("background-color", "Yellow");**
css(props)	**Set the css properties for all matched elements.** **e.g.** **$("#cssSpan").css({'background-color':'Yellow',** **'color':'black'});**

Other CSS functions

- .position()
- .scrollTop([val])
- .scrollLeft([val])

- .height([val])
- .width([val])
- .innerHeight()
- .innerWidth()
- .outerHeight(margin)
- .outerWidth(margin)



Method Chaining

- Calling one method after another, where each method affects the parent calling object
- These Methods always end up with “this” referring to calling object

- Example
  obj.m1().m2()….mn()

Other Functions

- Array Functions
  - makeArray
  - inArray
  - unique
  - merge
  - map
  - grep
- TestingFunctions
  - isArray
  - isFunction
  - isNumeric
  - type
  - etc..

each() Method

- Iterate over a jQuery object, array, javascript object properties; executing a function for each matched element.
- Syntax:

  $.each(coll,function(idx,elem){});

  →where: coll is either an array or javascript object properties..

  $(selector).each(function(idx,elem){});

```
<ul>
   <li>trip</li>
   <li>vacation</li>
   <li>abroad</li>
</ul>
```

```
$(document).ready(function(){
  $('li').each(function(index) {
    alert(index + ': ' + $(this).text());
  });
});
```

DOM Manipulation

DOM Manipulation

- Create Elements
- Insert Elements
  - Element Content
- Remove Elements

Clone Elements

- Clone elements via **.clone()**
  - Clone selected element
  - .clone(**true**) to copy events and data, too

- Example:
  $('#myid')**.clone()**

Insert Elements

- add and insert elements
  - inside others with **.append()**, **.appendTo()**, **.prepend()** and **.prependTo()**
  - before and after others with **.after()**, **.insertAfter()**, **.before()** and **.insertBefore()**
  - around others with **.wrap()**, **.wrapAll()** and **.wrapInner()**
  - in place of others with **.replaceWith()** and **.replaceAll()**

.append() & .prepend() Methods

- .append()

  e.g. $('#TestList').append('
Appended item
')

  The new item will be inserted as the last term

- .prepend()

  e.g. $('#TestList').prepend('
Appended item
')

  The new item will be inserted as the first item of the list

.append() & .prepend() Methods

- Both append() and prepend() methods take an infinite amount of new elements as parameters.

```
var item1 = $("<li></li>").text("Item 1");
var item2 = "<li>Item 2</li>“;
var item3 = document.createElement("li");

item3.innerHTML = "Item 3“;

$("#TestList").append(item1, item2, item3);
```

.before() & .after() Methods

- Used to insert things before or after one or several elements.
- Both after() and before() allows you to use HTML strings, DOM elements and jQuery objects as parameters and an infinite amount of them as well.
- Example:
  $('input.test1').before('Before')
  An italic tag will be inserted before each input element on the page using the "test1" class.

  $('input.test1').after('After')
  A bold tag will be inserted after each input element on the page using the "test1" class.

Methods variations

- There are variations of append() and prepend() methods, called appendTo() and prependTo().
  - $('#TestList').append('
Appended item
')
  - $('
Appended item
').appendTo('#TestList')
- There are variations of before() and after() methods, called insertBefore() and insertAfter().
  - $('#test').before('Before')
  - $('Before').insertBefore('#test')

.remove() & .empty() Methods

- . remove() method will delete the selected element(s)
  - $("#test").remove();
- . empty() method will only delete all child elements of the selected element(s).
  - $("#test").empty();
- The . remove() method comes with one optional parameter, allowing to filter the elements to be removed, using any of the jQuery selector syntaxes.
  - $("#test").remove(".bold");

jQuery Events

Shorthand Syntax

- .submit()
- .change()
- .focus()
- .focusin()
- .blur()
- .focusout()
- .mouseover()
- .mouseout()
- .mouseenter()
- .mouseleave()
- .mousemove()
- .dblclick()
- .keydown()
- .keypress()
- .keyup()
- .scroll()
- .resize()
- .error()

jQuery Event Object

- jQuery defines an event object with the most important properties needed cross-browsers

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

Event Binding and Delegation

- .on()
  - new in jQuery 1.7
  - the future of jQuery event handling
  - one event handler method to rule them all
  - use instead of .bind() and .live() and .delegate()
- .off()
  - unbinds event handlers bound by .on()

Event Delegation

```
$('#wrapper').on('click', 'button', function(event) {
  // button got clicked
});

// remove click listeners
$('#wrapper').off('click', 'button');
```

Event Binding ".on()" and Removing ".off()"

```
$('button')
.on('click', clickListener)
.on('click', otherListener);

// remove all click listeners
$('button').off('click');

// remove only the specified listener
$('button').off('click', clickListener);
```

Miscellaneous Event Methods

- .tigger()
  - Trigger events programmatically without waiting to user

```
$('button').trigger('click');
```

- .one()
  - Similar to bind but the event handler works once

jQuery Animations & Effects

jQuery UI

- **jQuery UI provides a comprehensive set of:**
  - ***Effects***
    - **Animated transitions and easing for rich interactions.**
  - ***Interaction plugins***
    - **Complex behaviors like drag and drop, resizing, selection and sorting.**
  - ***UI Widgets***
    - **Full-featured UI controls, each has a range of options and is fully themeable**

jQuery UI Effects

- jQuery provides a simple interface for doing various kind of amazing effects.

- jQuery methods allow us to quickly apply commonly used effects, which fall into 2 categories:
  - jQuery Effect Methods
  - UI Library Based Effects

jQuery & jQuery UI Effects

- jQuery provides a simple interface for doing various kind of amazing effects.

- jQuery methods allow us to quickly apply commonly used effects, which fall into 2 categories:
  - jQuery Effect Methods
  - UI Library Based Effects

jQuery Effects

- jQuery supports us with simple basic Effect methods which can be used in:
  - Showing and Hiding elements
  - Toggling the elements
  - Fading
  - Sliding
  - Custom Animation

Show/Hide Animation

[selector].*show* (speed , [callback]);

- ***speed*** → "slow", "normal", or "fast"
- ***callback*** → a function to be executed whenever the animation completed

[selector].*hide* (speed, [callback]);

Fading Animation

fadeIn(speed, [callback])	**Fade in all matched elements by adjusting their opacity and firing an optional callback after completion.**
fadeOut(speed, [callback])	**Fade out all matched elements by adjusting their opacity to 0, then setting display to "none" and firing an optional callback after completion.**
fadeTo(speed, opacity, callback)	**Fade the opacity of all matched elements to a specified opacity and firing an optional callback after completion.**
fadeToggle(speed, [callback])	**Toggle Fading in and out**

Sliding Animation

slideDown(speed, [callback])	**Reveal all matched elements by adjusting their height and firing an optional callback after completion.**
slideUp(speed, [callback])	**Hide all matched elements by adjusting their height and firing an optional callback after completion.**
slideToggle(speed, [callback])	**Toggle the visibility of all matched elements by adjusting their height and firing an optional callback after completion.**

Assignment