jQuery Fundamentals

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

Day2

jQuery Selectors & Filters

cont.

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.

Selecting Specific Siblings

- If you're looking to select the adjacent sibling of a particular element.
- Then you can use the adjacent sibling combinator (+).
- Similar to the child (>) combinator, the sibling combinator expects a selector expression on each side.
- The righthand expression is the subject of the selector, and the lefthand expression is the sibling you want to match.

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

cont.

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 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

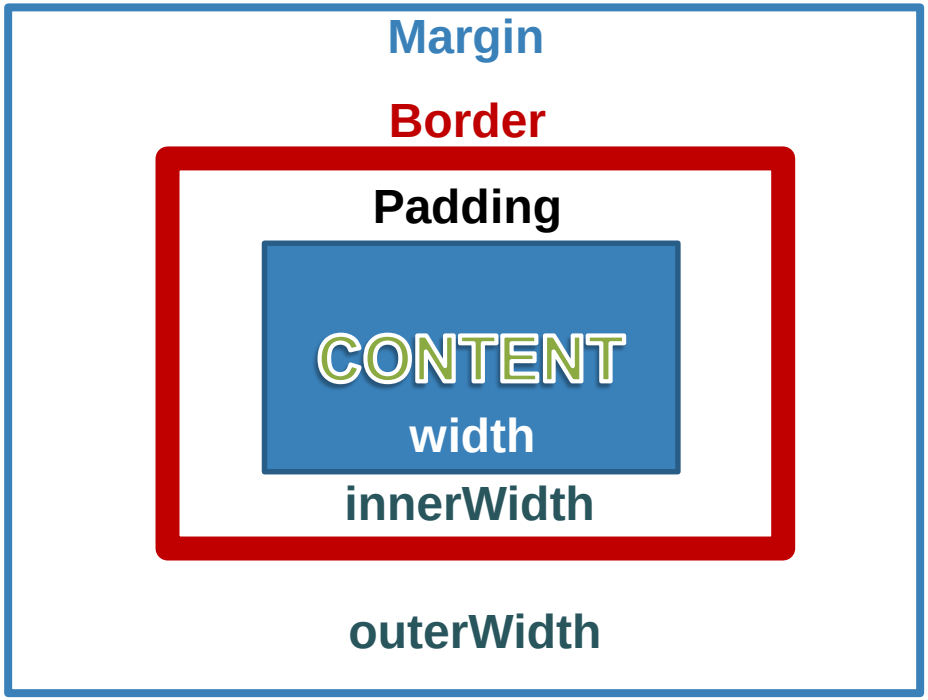
Attribute Methods that handles *Style*

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.

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(coll)
  - inArray
  - unique→uniqueSort
  - 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()**

https://api.jquery.com/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

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

jQuery Events

Shorthand Syntax

- .submit()
- .change()
- .focus()
- .blur()
- .focusout()

- .mouseover()
- .mouseout()
- .mouseenter()
- .mouseleave()
- .mousemove()
- .dblclick()

- .keydown()
- .keypress()
- .keyup()
- .scroll()
- .resize()
- .error()

jQuery Event Object

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

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

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 Binding ".on()" & 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);
```

Event Delegation

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

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

Miscellaneous Event Methods

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

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

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

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

jQuery Animations & 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

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 supports us with simple basic Effect methods which can be used in:
  - Showing and Hiding elements
  - Fading
  - Sliding
  - etc..
- We can create animated transitions using jQuery UI with these set of pre-built effects
  - **Movement:**
    - Bounce, Scale, Shake, Size
  - **Feedback:**
    - Highlight, Pulsate, Transfer
  - **Show/Hide:**
    - Blind, Clip, Drop, Explode, Fold, Puff, Scale, Slide

jQuery UI Effects

Blind	**Blinds the element away or shows it by blinding it in.**
Bounce	**Bounces the element vertically or horizontally n-times.**
Clip	**Clips the element on or off, vertically or horizontally.**
Drop	**Drops the element away or shows it by dropping it in.**
Explode	**Explodes the element into multiple pieces.**
Fold	**Folds the element like a piece of paper.**
Highlight	**Highlights the background with a defined color.**
Scale	**Shrink or grow an element by a percentage factor.**
Shake	**Shakes the element vertically or horizontally n-times.**
Size	**Resize an element to a specified width and height.**
Slide	**Slides the element out of the viewport.**
Transfer	**Transfers the outline of an element to another.**

jQuery UI Effects "Show/Hide"

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

- ***effect*** → values: 'blind', 'clip', 'drop', 'explode', 'fold', 'puff', 'slide', 'scale', 'size', 'pulsate'.
- ***speed*** → "slow", "normal", or "fast"
- ***callback*** → a function to be executed whenever the animation completed

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

Using .animate() Method

- **The *animate( )* method performs a custom animation of a set of numeric CSS properties.**
  - **top, left, width, height, opacity, fontSize, borderWidth**

- Properties can be animated
  - by number, percent, etc.
  - relatively ("+=200px", "-=20%", etc.)
  - by keyword: "hide", "show", or "toggle"

Using .animate() Method

selector.***animate***(params, [duration, [easing, [complete]]]);

selector.***animate***(params, options);

Using .animate() Method

- Example, slowly moving an element 300px to the right

```
$('.toMove').animate({
  left: '+=300px'
}, 800);

// same as
$('.toMove').animate({
  left: '+=300px'
}, {
  duration: 800
});
```

Complete

- Function executed when the animation ends
- Called once for *each* animated element

- Example:

```
$("#divTestArea3").fadeIn(2000, function()
{
  $("#divTestArea3").fadeOut(3000);
});
```

Easing

- Changes the velocity at different points in the animation
- A number of standard equations first created by Robert Penner
  - Available with jQuery only
    - linear
    - Swing
  - More easing functions are available within
    - jQuery- UI or
    - stand-alone (plug-in)at http://gsgd.co.uk/sandbox/jquery/easing/

http://easings.net/

Options object allows for fine tuning

- **duration**: A string or number determining how long the animation will run.
- **easing**: A string indicating which easing function to use for the transition. ("linear", "swing". More with plugin.)
- **queue**: A Boolean indicating whether to place the animation in the effects queue. If false, the animation will begin immediately.
- **specialEasing**: A map of one or more of the CSS properties defined by the properties argument and their corresponding easing functions (added 1.4).
- **step**: A function to be called for each step of the animation.
- **progress**: A function to be called after each animation step
- **complete**: A function to call once the animation is complete. (callback function.)
- …

Animation

- Animation can be chained
- By default multiple animations occur:
  - in sequence for the same element(s)
  - simultaneously for different element(s)
- .is(':animated')) identify elements that are currently animated

jQuery Interactions

jQuery Interactions

- **Draggable** - Makes items draggable by the mouse
- **Droppable** - Makes drop targets for draggables
- **Sortable** - Makes a list of items mouse sortable
- **Selectable** - Makes a list of items mouse and keyboard selectable
- **Resizable** - Makes an element resizable

jQuery Widgets

jQuery Widgets

- Accordion
- Autocomplete
- Button
- Datepicker
- Dialog
- Progressbar
- Slider
- Tabs
- …

jQuery Plugins

jQuery Plugins

- Plugins extend jQuery object
- Treat plugins as black box
- Plugins Considerations
  - Documentation
  - Updates
  - Support
- Finding Plugins
  - Search for specific functionality
  - http://plugins.jquery.com
  - http://jquery-plugins.net/popular

Assignment