The jQuery JavaScript Library

Hamid Zarrabi-Zadeh

Web Programming – Fall 2013
Outline

• Introduction
• Using jQuery
• DOM Manipulation
• jQuery Effects
• DOM Tree Traversal
• jQuery Utilities
What is jQuery?

- A light-weight JavaScript library designed to simplify writing JavaScript code
- The most popular library in use today, used by over 65% of the 10,000 most visited websites
- Major alternative libraries include
  - Prototype
  - MooTools
  - YUI
  - Ext JS
jQuery Features

- jQuery Main Functionalities
  - DOM Manipulation
  - Event Handling
  - Effects
  - AJAX
  - Utilities
  - Lots of plug-ins
jQuery Pros and Cons

• jQuery Pros
  – write less, do more
  – multi-browser support
  – light-weight
  – fast and optimized

• jQuery Cons
  – load overhead
Using jQuery
Including jQuery

- To include jQuery, just load it into your page

\[
\text{<script src="jquery.js"></script>}
\]

- For a faster load, use minified and gzipped version, from a content delivery network

\[
\text{<script src="http://ajax.googleapis.com/ajax/libs/jquery/1.10.2/jquery.min.js"></script>}
\]
Usage Style

- With jQuery, we typically find (query) elements and manipulate them
- Main usage style:
  - $(selector).action();
- Examples

```javascript
$('p').hide();

$(document).ready(function() {
    // main code
});
```
Chaining

- The output of $() function is a jQuery object
- jQuery actions are chainable, as they usually return a jQuery object

```javascript
$('img').addClass('test').hide();
```
Selectors

- jQuery uses CSS-like selectors
- We can select elements by tag name, id, class, attributes, etc.

```javascript
$('p')
$('id')
$('.class')
$('parent > child')
$('[attr=value]')
$('p:first')
$('tr:odd')
$(':button')
$(document.body)
```
Events

• With jQuery, we can easily attach handlers to events

```javascript
$('p').click(function() {
    $(this).hide();
});

$('p').on('click', function() { ... });
```
DOM Manipulation
Get/Set Content

- jQuery methods for DOM manipulation include
  - `text()`
  - `html()`
  - `val()`

```javascript
alert($('p').text());
$('p').html('Hello!');
```
Attributes

- You can get/set element attributes by `attr()` method

```javascript
alert($('p').attr('title'));

$('p').attr('title', 'Test');

$('a').attr({
  title: 'Test',
  href: 'http://example.com'
});
```
Styles

- jQuery methods for CSS manipulation include:
  - `addClass`
  - `removeClass`
  - `toggleClass`
  - `css`

```javascript
$('p').addClass('active').css('color', 'blue');

$('p').css('color'); // returns blue
```
Dimensions

- jQuery has the following dimension methods
  - width()
  - height()
  - innerWidth()
  - innerHeight()
  - outerWidth()
  - outerHeight()
jQuery Effects
jQuery Effects

• jQuery provides methods for basic effects, including
  – hide and show
  – fade
  – slide
  – animate!
Hide/Show

- We can hide/show elements by `hide()`, `show()` and `toggle()` methods

```javascript
$('p').click(function() {
    $(this).hide();
});
```

- The methods accept optional `speed` and `callback` parameters

```javascript
$('p').hide(1000, function() {
    $(this).show();
});
```
Fading

- jQuery has the following fading methods
  - fadeIn()
  - fadeOut()
  - fadeToggle()
  - fadeTo()

- All with optional speed and callback parameters

```javascript
$('p').fadeOut();
$('p').fadeOut(1000);
$('p').fadeOut('slow');
$('p').fadeOutTo(1000, .3);
```
Sliding

- The following sliding methods are available
  - slideDown()
  - slideUp()
  - slideToggle()
- With optional speed and callback parameters

```javascript
$('p').slideUp();
$('p').slideDown(1000);
```
Animation

- Custom animations can be created with `animate()` function

```javascript
$('p').click(function() {
  $(this).animate({height: '200px'}, 'slow');
})
```

- You can change several styles at once
- animations are queued
DOM Tree Traversal
Parents

- We can access parents of an element using:
  - parent()
  - parents()
  - parentsUntil()
  - closest()

```javascript
$('p').parents();
$('p').parentsUntil('div');
```
Children

- We can access children of an element using:
  - `children()`
  - `find()`

```javascript
$('p').children();
$('p').find('span');
```
• We can access *siblings* of an element using:
  - siblings()
  - next()
  - nextAll()
  - nextUntil()
  - prev()
  - prevAll()
  - prevUntil()
Filtering

- Filtering methods
  - `first()`
  - `last()`
  - `eq()`
  - `filter()`
  - `not()`

```javascript
$('p').first()
$('p').eq(2)  // the 3rd <p> element

$('p').filter('.highlight')
$('p').not(':selected')
```
jQuery Utilities
Each

• The `each()` method iterates over a jQuery object and executes a function for each element

```javascript
$('li').each(function(index) {
    $(this).attr('title', 'Item ' + index);
});
```

• You can use `$.each()` to iterate over arrays and collections

```javascript
$.each([2, 4, 16], function(index, value) {
    alert(index + ': ' + value);
});
```
• The $.extend() method is used to merge the contents of two or more objects
  - $.extend(target [, object1 ] [, objectN ])

```javascript
var object = $.extend({}, object1, object2);

function test(params) {
  var defaults = { ... };
  $.extend(defaults, params);
  ...
}
```
References

- jQuery API Documentation
  - [http://api.jquery.com](http://api.jquery.com)
- W3Schools
  - [http://www.w3schools.com/jquery](http://www.w3schools.com/jquery)