

# jQuery

**instructor: Dr.Dendej Rakrattanatrai**

# Overview

- JavaScript library for
  - HTML document traversing/modifying
  - event handling
  - animating
  - Ajax
- Cross-browser
- Stop worrying about IE
- Sibling Projects
  - jQuery UI (Useful Widgets)
  - jQuery plugins (Useful additions)

# Overview

- official website => <http://jquery.com>
- Outstanding documentation
- Lightweight (~19 KB for minified format)
- Simple code
- Many plugins and special effects
- Ajax support
- Cross-browser support
- The idea: find stuff and do something to it

# Getting Started: Two Ways

- Download jQuery and include it in your HTML pages:

```
<script src="jquery-1.3.2.min.js"  
      type="text/javascript"></script>
```

- Use jQuery via Google Code:

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

- Internet connection is required for the latter case to be realized, More details:  
<http://code.google.com/apis/libraries/devguide.html#jquery>

# Local Copy jQuery

- There are 2 variations.
  - One for development
  - The other for real production (minified)
- For examples
  - jquery-1.4.4.js (size: 183KB)
  - jquery-1.4.4.min.js (size: 78.6KB → White space removed)
- Just adding the src link to the distribution file .e.g.:
  - `<script src="jquery-1.4.4.min.js" type="text/javascript"></script>`

# Loading jQuery - CDN

- System of computers with copy of data
- When client requests data closest machine is used
- With jQuery, browser may have already cached jQuery file
- Using Google CDN
  - `<script src= "http://ajax.googleapis.com/  
ajax/libs/jquery/1.4.4/jquery.min.js"  
type="text/javascript"></script>`
- For Microsoft CDN
  - `<script src="http://ajax.microsoft.com/  
ajax/jquery/jquery-1.4.4.min.js"  
type="text/javascript"></script>`

# Launching on Document Ready

- `$(document).ready(function() {  
 // Your code here  
});`
- Checks and waits until the document object is ready to be manipulated (that is, everything including images are loaded).

# Document Ready Function

```
<script type="text/javascript">
    $(document).ready(function() {
        alert("Displayed when page loaded");
    });
</script>
```

## Shortcut version

```
<script type="text/javascript">
    $(function() {
        alert("Displayed when page loaded");
    });
</script>
```



# jQuery Function

- jQuery exposes one function called jQuery
- (which does a lot more than shown below)
- `jQuery("h1")`
  - returns array of all h1 elements
- `jQuery(".nav")`
  - returns array of all elements with class 'nav'
- `jQuery("#important")`
  - returns the element with id 'important'
- `jQuery('div.nav p')`
  - returns all p elements inside div with class 'nav'

# jQuery Function alias

- jQuery is aliased to \$
- Shorter
- Other JavaScript libraries
- use same alias

<code>jQuery("h1")</code>	<code>\$("h1")</code>
<code>jQuery(".nav")</code>	<code>\$(".nav")</code>
<code>jQuery("#important")</code>	<code>\$("#important")</code>
<code>jQuery('div.nav p')</code>	<code>\$('div.nav p')</code>

# Selectors

- jQuery uses CSS-style selectors to find specific elements in the DOM
  - `$('h1,h2,h3')`
  - `$('div#content p.blurb')`
  - `$("a")`
  - `$("div")`
  - `$('#myForm :radio:checked')`
    - filter a form element
  - `$('li:first')`
    - find a specific element
- Notice, both single quote and double quotes can be used around the elements

# jQuery Output

- jQuery selection methods all return jQuery collections which actually are arrays.
- For examples, any array methods can be applied with those jQuery output right away:
  - `$('div').length`
  - `$('div')[0]`
  - `$('div')[1]`
  - `$('div').size()`
  - `$('div').each(function() {  
 console.log(this);  
});`

# Syntax

- `$("#header")`
  - Get the element with `id="header"`
- `$("h3")`
  - Get all `<h3>` element
- `$("div#content .photo")`
  - Get all element with `class="photo"`
  - nested in the `<div id="content">`
- `$("ul li")`
  - Get all `<li>` element nested in all `<ul>`
- `$("ul li:first")`
  - Get only the first `<li>` element of the `<ul>`

# Magic Selectors

➤ Start with ":"

- :first
- :last
- :header
- :hidden
- :contains
- :radio
- :input

# HTML Manipulation

- `$('div').html()`
- `$('div').html('<p>Hi!</p>')`
- `$('div').append('<p>Hi!</p>')`
- `$('div').after('<p>Hi!</p>')`
- `$('div').remove()`
- `$('div').empty()`

# Attributes

- `$('img').attr('alt')`
- `$('img').attr('alt', 'Logo')`
- `$('img').attr({ src: 'logo.png', alt: 'Logo' });`
- `$('select').val()`
- `$('input').val('First Name')`



# Creating new Elements

- Example #1

```
...  
<script type="text/javascript" charset="utf-8">  
  $(function(){  
    var element = $("<p>Hello World</p>");  
    element.css("color", "#f00").insertAfter("#header");  
  });  
</script>  
...  
<div id="header">  
<h1>Heading</h1>  
...
```

**Heading**

Hello World

- Example #2

```
$("p.message").html("New <b>Text</b> for paragraph")  
  set contents of paragraph(s) with class "message"  
  to "New <b>Text</b> for paragraph"
```

# Changing the Class of Elements

- `$('p').addClass('test')`
- `$('p').removeClass('test')`
- `$('p').toggleClass('test')`
- For example
- `$('h1').addClass('realBold')`
  - Add the class 'realBold' to all h1 tags

# CSS

- `$('div').css('color', 'red')`
- `$('div').height()`
- `$('div').height('20px')`
- `$('div').innerHeight()`
- `$('div').outerHeight()`
- For Example
- `$("p").css("color","red");`
  - Set text in all paragraphs to color red
- `$("p").css({"color":"red", "font-size": "2em"});`
  - Set text in all paragraphs to color red and font size 2em

# Events

- `$('button').click(); // button click`
- `$('button').click(function(){ // button click  
    $(this).css('color', 'red'); // with handler  
});`
- `$('form').submit(); // form submit`

# Special Effects

- `$('div').hide('slow')`
- `$('div').hide('fast')`
- `$('div').show('slow')`
- `$('div').show('fast')`
- `$('div').slideDown('slow')`
- `$('div').slideDown('fast')`
- `$('div').fadeOut(1000)`
- `$('div').toggle()`
- `$('div').fadeIn('slow')`

# DOM Traversal

- `${div.section).parent()`
- `${div.section).next()`
- `${div.section).prev()`
- `${div.section).nextAll('p')`
- `$('h1:first').parents();`

# Ajax

- `$.get(url, parameters, callback)`
- `$.post(url, parameters, callback)`