



Web Application Development
Web Development

CS 228
CS 303

Fall 2015

numangift.wordpress.com/web-development-spring-2015

What is jQuery?

- jQuery is a fast, small, and feature-rich JavaScript library.
- It makes things like
 - HTML document traversal and manipulation
 - event handling
 - Animation
 - Ajax

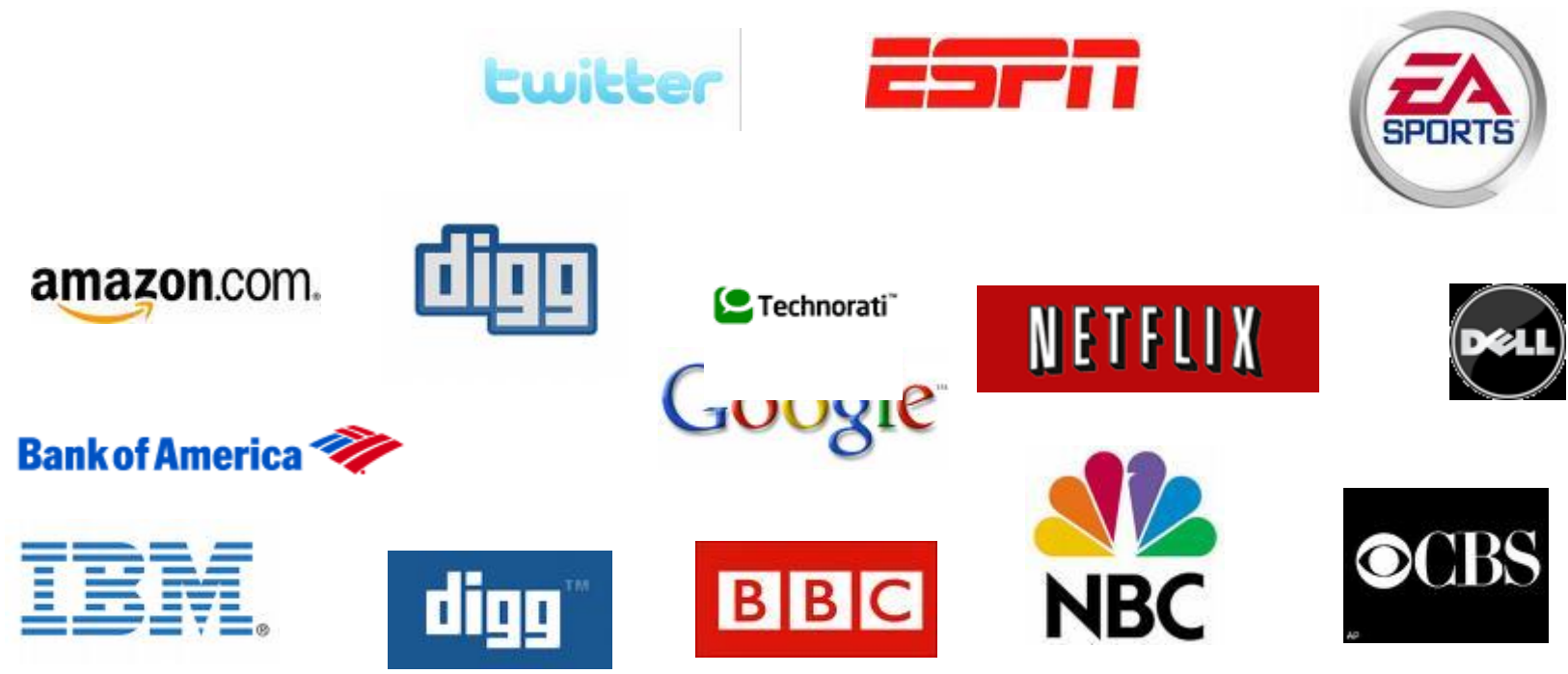
much **simpler** with an easy-to-use API that works across a multitude of browsers.

- With a combination of versatility and extensibility, jQuery has changed the way that millions of people write JavaScript.

Why jQuery?

- Lightweight : 19KB in size (Minified and Gzipped)
- CSS1 - 3 Complaint
- Cross Browser
 - (IE 6.0+, FF 2+, Safari 3.0+, Opera 9.0+, Chrome)

Who's using jQuery?



http://docs.jquery.com/Sites_Using_jQuery

Including jQuery

- Step 1: Include a link to **jquery.min.js** in the HEAD of each HTML file.
- Note the Google URL.

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<title>A Sensible Title for Search Engines to Find</title>
<link rel="stylesheet" href="styles.css">
<script src="//ajax.googleapis.com/ajax/libs/jquery/1.11.0/jquery.min.js">
</script>
</head>
```

Including jQuery

- Step 2: Include a link to your own .js file in the HEAD of the HTML file.
- Place it after the Google jQuery link.

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<title>A Sensible Title for Search Engines to Find</title>
<link rel="stylesheet" href="styles.css">
<script src="//ajax.googleapis.com/ajax/libs/jquery/1.11.0/jquery.min.js">
</script>
<script src="scripts/myjqueryfile.js"></script>
</head>
```

Launching Code on Document Ready

- Unfortunately, the code doesn't start running until after the document has finished downloading, including banner images. When the document is ready to be manipulated, jQuery has a statement known as the [ready event](#).
- To ensure that their code runs after the document is ready, many JavaScript programmers wrap their code in an onload function:
- For example, inside the ready handler to the link:

```
1  
2 window.onload = function() {  
3   alert( "welcome" );  
4 }  
5
```

```
1  
2 $( document ).ready(function() {  
3   // Your code here.  
4 });  
5
```

```
1  
2  
3  
4 $( document ).ready(function() {  
5   $( "a" ).click(function( event ) {  
6     alert( "Thanks for visiting!" );  
7   });  
8  
9
```

Launching Code on Document Ready

- Step 3: Write *your own* .js file correctly (so it works!).

```
// boilerplate for any jQuery
$(document).ready(function(){ // do not delete
// -----

// put all jQuery in here, and it will execute after the HTML page
// has completely loaded

// -----
}); // do not delete; closes (document).ready function
```


jQuery philosophy

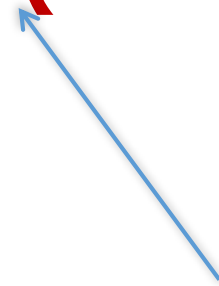
Find Some Elements



```
$("div").addClass("xyz");
```



Do something with them



jQuery Object

Calling jQuery with \$

- Any code statement or instruction that uses jQuery uses the character “\$”
- This will **find** the text enclosed in H1 tags:
`$("#h1").text();`
- This will **change** the text enclosed in H1 tags:
`$("#h1").text("ABC123");`
It will change *all* H1 tags on the page.

A Basic Example

```
<body>  
  <div>  
    <p>I m a paragraph 1</p>  
    <p>I m a paragraph 2</p>  
  </div>  
  <p>I m another paragraph</p>  
</body>
```

A Basic Example

```
<body>
  <div>
    <p>I m a paragraph 1</p>
    <p>I m a paragraph 2</p>
  </div>
  <p>I m another paragraph</p>
</body>
```

Select all paragraphs.
`$("p")`

A Basic Example

```
<body>
  <div>
    <p class="red">I m a paragraph -1</p>
    <p class="red">I m a paragraph -2</p>
    http://tfilm.me/5398-29-palm.html
  </div>
  <p class="red">I m another paragraph</p>
</body>
```

Select all paragraphs. Add a class to them.

```
$("p").addClass("red");
```

Selector Basics

Just pass a **selector** to `$()`

What is **selector**?

Use any CSS selector

Find, change an ID or a class

```
$("#wrapper"); // find the ID wrapper
```

```
$(".item"); // find the class item
```

```
$("#wrapper").append("<p>Hooray!</p>");
```

```
$(".item").append("<p>Hooray!</p>");
```

Quotation marks may be single or double.

Selector Basics

Think about your simplest css file.

```
#header{  
    margin : 0 auto;  
}  
div{  
    margin : 0px;  
    padding : 0px  
}  
ul.menu li{  
    .....  
}
```


Selector Basics

The **red** colored items are **selectors**

```
#header{  
    margin : 0 auto;  
}  
div{  
    margin : 0px;  
    padding : 0px  
}  
ul.menu li{  
    .....  
}
```

Selector Basics

Selecting using selectors

- Selecting By Id
 - \$(" #header")

Selector Basics

Selecting using selectors

- Selecting By Id
 - \$(" #header")
- Selecting By Class
 - \$(".updated")

Selector Basics

Selecting using selectors

- Selecting By Id
 - \$(" #header")
- Selecting By Class
 - \$(".updated")
- Selecting by tag name
 - \$("table")

Selector Basics

Selecting using selectors

- Selecting By Id
 - \$(" #header")
- Selecting By Class
 - \$(".updated")
- Selecting by tag name
 - \$("table")
- Combine them
 - \$("table.user-list")
 - \$(" #footer ul.menu li")

Basic Selector Example

This is my page

```
<body>
  <div id="header">
    <span id="logo">Logo here...</span>
    <ul class="menu">
      <li>user name</li>
      ...
      <li>logout</li>
    </ul>
  </div>
  ....
</body>
```

Basic Selector Example

\$ (" #header")

```
<body>
  <div id="header">
    <span id="logo">Logo here...</span>
    <ul class="menu">
      <li>user name</li>
      ...
      <li>logout</li>
    </ul>
  </div>
  ...
</body>
```

Basic Selector Example

`$ ("ul .menu")`

```
<body>
  <div id="header">
    <span id="logo">Logo here...</span>
    <ul class="menu">
      <li>user name</li>
      ...
      <li>logout</li>
    </ul>
  </div>
  ...
</body>
```


Basic Selector Example

```
$ ("ul.menu li")
```

```
<body>
  <div id="header">
    <span id="logo">Logo here...</span>
    <ul class="menu">
      <li>user name</li>
      ...
      <li>logout</li>
    </ul>
  </div>
  ....
</body>
```

Using filters for selecting

- Basic Filters
 - :first, :last, :even, :odd,

Basic Filters Example

Student list table. Lets make it zebra.

| Name | Class | Roll No. | Comment |
|-------|-------|----------|--------------|
| Raju | XII | 2 | Good |
| Masud | IX | 1 | Good |
| Apu | XII | 3 | |
| Mizan | XII | 5 | |
| Karim | VI | 2 | Satisfactory |

Basic Filters Example

```
$("#students tr:even").css("background-color", "#dde")
```

| Name | Class | Roll No. | Comment |
|-------|-------|----------|--------------|
| Raju | XII | 2 | Good |
| Masud | IX | 1 | Good |
| Apu | XII | 3 | |
| Mizan | XII | 5 | |
| Karim | VI | 2 | Satisfactory |

Using filters for selecting

- Basic Filters
 - :first, :last, :even, :odd,
- Content Filters:
 - :empty , :contains(text), :has(selector),

Content Filters Example

```
$("#students tr:even").css("background-color", "#dde");  
$("#students td.comment:empty").text("No Comment");
```

| Name | Class | Roll No. | Comment |
|-------|-------|----------|--------------|
| Raju | XII | 2 | Good |
| Masud | IX | 1 | Good |
| Apu | XII | 3 | No Comment |
| Mizan | XII | 5 | No Comment |
| Karim | VI | 2 | Satisfactory |

Using filters for selecting

- Basic Filters
 - :first, :last, :even, :odd,
- Content Filters:
 - :empty , :contains(text), :has(selector),
- Attribute Filters:
 - [attribute], [attribute=value], [attribute!=value],
.....

Attribute Filters Example

```
$("#students tr:even").css("background-color", "#dde");  
$("#students td.comment:empty").text("No Comment");  
$("#students td[align='center']").addClass("ocean");
```

| Name | Class | Roll No. | Comment |
|-------|------------|----------|--------------|
| Raju | <i>XII</i> | 2 | Good |
| Masud | <i>IX</i> | 1 | Good |
| Apu | <i>XII</i> | 3 | No Comment |
| Mizan | <i>XII</i> | 5 | No Comment |
| Karim | <i>VI</i> | 2 | Satisfactory |

Using filters for selecting

- Basic Filters
 - :first, :last, :even, :odd,
- Content Filters:
 - :empty , :contains(text), :has(selector),
- Attribute Filters:
 - [attribute], [attribute=value], [attribute!=value],
- Forms
 - :input, :text, :submit, :password,
 - :enabled, :disabled, :checked,

Forms Selector Example

```
$(":submit").click(function(e) { ... });
```

```
$("input:disabled").val("You cannot change me");
```

```
$("#form-id input:checked").addClass("selected");
```

Now we can Select

- Let's perform some **action**

jQuery Methods

- DOM Manipulation
 - before(), after(), append(), appendTo(),

Dom Manipulation Example

Move all paragraphs in div with id "contents"

\$ ("p")

```
<body>
  <h1>jQuery</h1>
  <p>jQuery is good</p>
  <p>jQuery is better</p>
  <div id="contents"></div>
  <p>jQuery is the best</p>

</body>
```

Dom Manipulation Example

Move all paragraphs in div with id "contents"

```
$( "p" ) . appendTo ( "#contents" ) ;
```

```
<body>  
  <h1>jQuery</h1>  
  <div id="contents">  
    <p>jQuery is good</p>  
    <p>jQuery is better</p>  
    <p>jQuery is the best</p>  
  </div>  
</body>
```

Dom Manipulation Example

Move all paragraphs in div with id "contents"

```
$ ("p") .appendTo (" #contents" );  
$ ("h1") .append (" Dom Manipulation" );
```

```
<body>  
  <h1>jQuery Dom Manipulation</h1>  
  <div id="contents">  
    <p>jQuery is good</p>  
    <p>jQuery is better</p>  
    <p>jQuery is the best</p>  
  </div>  
</body>
```

jQuery Methods

- DOM Manipulation
 - before(), after(), append(), appendTo(),
- Attributes
 - css(), addClass(), attr(), html(), val(),

Attributes Example

Make the texts of last paragraph bold

```
$("#contents p:last").css("color", "green");
```

```
<body>
  <h1>jQuery Dom Manipulation</h1>
  <div id="contents">
    <p >jQuery is good</p>
    <p>jQuery is better</p>
    <p style="color:green">jQuery is the
best</p>
  </div>
</body>
```

More Attributes Example

Setting

- `$("#img.logo").attr("align", "left");`
- `$("#p.copyright").html("© 2009 ajaxray");`
- `$("#input#name").val("Spiderman");`

More Attributes Example

Setting

- `$("#img.logo").attr("align", "left");`
- `$("#p.copyright").html("© 2009 ajaxray");`
- `$("#input#name").val("Spiderman");`

Getting

- `var alignment = $("#img.logo").attr("align");`
- `var copyright = $("#p.copyright").html();`
- `var username = $("#input#name").val();`

jQuery Methods

- DOM Manipulation
 - before(), after(), append(), appendTo(),
- Attributes
 - css(), addClass(), attr(), html(), val(),
- Events
 - click(), bind(), unbind(), live(),

Event Example

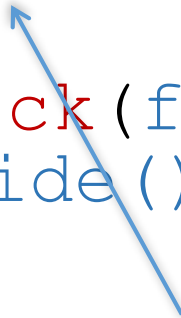
Start when DOM is ready

```
$(document).ready(function() {  
    $(selector).eventName(function() {...});  
});
```

Event Example

Bind all interactions on events.

```
$(document).ready(function() {  
    $("#message").click(function() {  
        $(this).hide();  
    })  
});
```

A blue arrow originates from the word "click" in the second line of the code and points diagonally upwards and to the left towards the word "ready" in the first line, illustrating the relationship between the event and the document ready state.

```
<span id="message" onclick="..."> blah blah  
</span>
```

Event Example

You can fire events manually.

```
$(document).ready(function() {  
  
    $("span#message").click(function() {  
        $(this).hide();  
    })  
  
    $("#form-id:reset").click();  
  
});
```

jQuery Methods

- DOM Manipulation
 - before(), after(), append(), appendTo(),
- Attributes
 - css(), addClass(), attr(), html(), val(),
- Events
 - click(), bind(), unbind(), live(),
- Effects
 - hide(), fadeOut(), toggle(), animate(),

Effects Example

When “**show-cart**” link clicked, slide up/down “**cart**” div.

```
$( "#show-cart" ).click( function() {  
    $( "#cart" ).slideToggle( "slow" );  
})
```

Animations

- `show()`
- `hide()`
- `fadeIn()`
- `fadeOut()`
- `slideUp()`
- `slideDown()`

Effects Example

Build your custom animation

```
$( "#show-cart" ).click( function () {  
    $( "#cart" ).slideToggle( "slow" );  
})
```

```
$( "#showdown" ).click( function () {  
    $( "#my-div" ).animate({  
        width: "70%",  
        opacity: 0.4,  
        fontSize: "3em"  
    }, 1200 );  
});
```

jQuery Methods

- DOM Manipulation
 - `before()`, `after()`, `append()`, `appendTo()`,
- Attributes
 - `css()`, `addClass()`, `attr()`, `html()`, `val()`,
- Events
 - `click()`, `bind()`, `unbind()`, `live()`,
- Effects
 - `hide()`, `fadeOut()`, `toggle()`, `animate()`,
- Ajax
 - `load()`, `get()`, `ajax()`, `getJSON()`,

Ajax Examples

Load a page in a container

```
$("#comments").load("/get_comments.php");
```

```
$("#comments").load("/get_comments.php", {max :  
5});
```

Ajax Examples

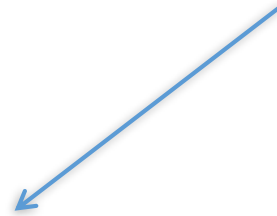
Send ajax request with data

```
$.get("/edit_comment.php",  
      {id: 102, comment: "I m edited"}  
);
```

Ajax Examples

You can send serialized form as data

```
$.get("/edit_comment.php",  
      $("#edit-comment").serialize()  
);
```



id=102&comment=I+m+edited

Ajax Examples

Set a callback function for handling response data

```
$.get("edit_comment.php",  
    $("form#cmm-edit").serialize(),  
    function(data) {  
        if(data == "success")  
            alert("Comment Edited!");  
    }  
);
```


Chaining Methods

- Most jQuery methods return jQuery object
- You can chain them together

Chaining Methods

- Most jQuery methods return jQuery object
- You can chain them together

```
$("#deleted").addClass("red").fadeOut("slow");  
$:button).val("Click  
Me").click(function(){...})
```

Chaining Methods

- Most jQuery methods return jQuery object
- You can chain them together

```
$("#deleted").addClass("red").fadeOut("slow");  
$:button).val("Click Me").click(function() {...})
```

This will not work -

```
$:button).val().click(function() {...})
```

This method will return string

A blue arrow points from the text "This method will return string" to the `val()` method in the code snippet above.

Adding and Removing an HTML Class

Important: You must place the remaining jQuery examples inside the ready event so that your code executes when the document is ready to be worked on.

Another common task is adding or removing a class.

First, add some style information into the <head> of the document, like this:

```
1 <style>
2 a.test {
3   font-weight: bold;
4 }
5 </style>
```

Next, add the [.addClass\(\)](#) call to the script:

```
1 $( "a" ).addClass( "test" );
```

All <a> elements are now bold.

To remove an existing class, use [.removeClass\(\)](#):

```
1 $( "a" ).removeClass( "test" );
```

Special Effects

jQuery also provides some handy [effects](#) to help you make your web sites stand out. For example, if you create a click handler of:

```
1  
2  
3 $( "a" ).click(function( event ) {  
4   event.preventDefault();  
5   $( this ).hide( "slow" );  
6   });  
7
```

Then the link slowly disappears when clicked.

Find, change an ID or a class

```
$("#wrapper"); // find the ID wrapper
```

```
$(".item"); // find the class item
```

```
$("#wrapper").append("<p>Hooray!</p>");
```

```
$(".item").append("<p>Hooray!</p>");
```

Quotation marks may be single or double.

Searching the DOM

```
$("#destinations li");
```

Selects only the LI tags inside the element with the ID “destinations.” (In the example, the element is a UL.) This would also include all LI tags *inside any nested list* that was inside the “destinations” list.

```
$("#destinations > li");
```

Selects only “direct descendants.” The greater-than sign *limits* this to *only* the LIs that are direct children of the ID “destinations.” No LIs in any nested lists will be affected.

Searching the DOM (2)

```
$("#destinations li:first");  
$("#destinations li:last");  
$("#destinations li:odd");  
$("#destinations li:even");
```

These are called “pseudo classes.” They select just what they say. Can be used to style **tables** dynamically, for example.

“Traversing” the DOM

```
$("#tours").find("li").last();
```

```
$("#tours").find("li").first().next();
```

```
$("#tours").find("li").first().parent();
```

This one goes “up” the DOM to find the direct parent of the first LI.

```
$("#tours").children("li");
```

This finds only the LIs that are the direct children of the list with the ID “tours.”

Summary of Levels 1 & 2

- We can use jQuery to target an element that has an **ID** or a **class** in the HTML: "**#wrapper**", "**.photo**", etc.
- We can use jQuery to target any element, using its **tag** or **selector**: "**h1**", "**p**", "**div**", etc.
- We can use jQuery to “walk” up and down the DOM: **next()**, **first()**, **parent()**, etc.

Level 3: Changing things

```
.append(x) // at end, inside  
.prepend(x) // at top, inside  
.before(x) // at top, outside  
.after(x) // at end, outside
```

Example:

```
$( '.vacation' ).append(price);
```

Inside the element with the class *vacation*, at the **bottom**, add the value of *price*.
(Hint: *price* is a variable!)

Changing things (2)

```
.appendTo(x)        // end, inside  
.prependTo(x)       // top, inside  
.insertBefore(x)    // top, outside  
.insertAfter(x)     // end, outside
```

Example:

```
price.appendTo($('.vacation'));
```

Inside the element with the class *vacation*, at the **bottom**, add the value of *price*.

Changing things (3)

```
$( 'button' ).remove();
```

We can use jQuery to completely **remove** an element from the HTML page.

In the example at Code School, when we click the button, the button *disappears* and is *replaced* by a price.

(Note: It will be *gone forever*.)

A function in jQuery

```
$('#button').on('click', function() {  
    // some code here  
});
```

This function doesn't have a name. It *calls itself* when someone clicks the button. Whatever code you write in the middle will run every time the button (*any* button) is clicked.

A function in jQuery (2)

```
$('.choice').on('click', function() {  
    // some code here  
});
```

This function is exactly the same as the previous one, but instead of using the name of the HTML tag (**button**), here we use a *class* assigned to that button (**.choice**).

The keyword *this*

- When you write a function, often you want it to work for *more than one thing*.
- Example: You want to build a slideshow with a thumbnail image for each photo. Click a thumbnail, and the photo changes.
- How can you use *the same function* for every thumbnail and photo in the set?


```
$('.big').hide();
var current = $('#pics').find('.big').first();
$(current).show().addClass('view');
$('#pics').before(current); // see below

$('.thumb').on('click', function() {
    $(current).remove();
    current = $(this).prev();
    $('#pics').before(current); // see below
    $(current).show().addClass('view');
});
```