COMP335
Web Application Development

JQuery

Tentative Schedule

- Group Project
  - group name and short (one paragraph) description
  - **http://goo.gl/forms/eo5wnbMtXf**

- Today
  - jQuery
- Next class
  - AJAX

  - Assignment#2-CSS
    - **by Friday 2/26**
  - Assignment#3-JavaScript
    - **TBD**
  - In-class programming exam (tentative)
    - **3/14 (M)**

- Group project prototyping and presentation
  - 2/29 (M), 3/4(F)

---

jQuery History

- A library or framework is software that you can utilize in your own software, which provides some common implementations of standard ideas.

- In August 2005, jQuery founder John Resig was looking into how to better combine CSS selectors with succinct JavaScript notation.

- jQuery is now the most popular JavaScript library currently in use as supported by the statistics from BuiltWith.com.

---

example: basic.html

Including jQuery

- You must either:
  - link to a locally hosted version of the library
  - Use an approved third-party host, such as Google, Microsoft, or jQuery itself

- Benefits using a third-party content delivery network (CDN)?
  - The bandwidth of the file is offloaded
  - The user may already have cached the third-party file

- Disadvantage to the third-party CDN?
  - your jQuery will fail if the third-party host fails (unlikely)

---

jQuery Selectors

- The relationship between DOM objects and selectors is important in JavaScript programming.
- A shortcut notation for jQuery() : $()
- The four basic selectors are:
  - `$("*")` Universal selector matches all elements (and is slow).
  - `$("tag")` Element selector matches all elements with the given element name.
  - `$(".class")` Class selector matches all elements with the given CSS class.
  - `$("#id")` Id selector matches all elements with a given HTML id attribute.
- These selectors replace the use of getElementById() and getElementsByTagName().

---

example: basic.html

The Document Ready Event

```
$(document).ready(function(){

  // jQuery methods go here...

});
```

Anonymous (lambda, \, λ) function: you can write local functions that can be passed as arguments or returned as the value of function calls.

- This is to prevent any jQuery code from running before the document is finished loading (is ready). For example,
  - Trying to hide an element that is not created yet
  - Trying to get the size of an image that is not loaded yet

Slide 1

```
descendant (space)
child (>)
adjacent sibling (+)
and general sibling (~)      <body>
                               <nav>
                                 <ul>
                                   <li><a href="#">Canada</a></li>
                                   <li><a href="#">Germany</a></li>
                                   <li><a href="#">United States</a></li>      $("#main time")
                                 </ul>
                               </nav>
                               <div id="main">
                                 Comments as of <time>November 15, 2012</time>
        $("#main>time")        <div>
                                   <p>By Ricardo on <time>September 15, 2012</time></p>
                                   <p>Easy on the HDR buddy.</p>
        $("#main div p:first-child")  </div>
                                 <hr/>
                                 <div>
                                   <p>By Susan on <time>October 1, 2012</time></p>
                                   <p>I love Central Park.</p>
                                 </div>
                                 <hr/>
                               </div>
                               <footer>
                                 <ul>
                                   <li><a href="#">Home</a> | </li>
                                   <li><a href="#">Browse</a> | </li>
                                 </ul>
                               </footer>
                             </body>
```

Slide 2

jQuery Attributes and Properties

- The set of attributes related to DOM elements are the ones specified in the HTML tags.
  - The *href* attribute of an <a> tag
  - The *src* attribute of an <img>
  - The *class* attribute of most elements

- In jQuery we can both **set** and **get** an attribute value by using the attr() method on any element from a selector.

Slide 3

Changing CSS

- jQuery provides the extremely intuitive css() methods.
- To get a css value, use css() method with 1 parameter:
  ```
  // get the color
  var color = $("#colourBox").css("background-color");
  ```

- To set a CSS variable use css() with two parameters: the first being the CSS attribute, and the second the value.
  ```
  // set color to red
  $("#colourBox").css("background-color", "#FF0000");
  ```

Slide 4

A comparison of node creation in JS and jQuery

```
// pure JavaScript way
var jsLink = document.createElement("a");
jsLink.href = "http://www.funwebdev.com";
jsLink.innerHTML = "Visit Us";
jsLink.title = "JS";

// jQuery way
var jQueryLink = $("<a href='http://funwebdev.com'
title = 'jQuery'>Visit Us</a>");

// jQuery long-form way
var jQueryVerboseLink = $("<a></a>");
jQueryVerboseLink.attr("href",'http://funwebdev.com');
jQueryVerboseLink.attr("title","jQuery verbose");
jQueryVerboseLink.html("Visit Us");
```

Slide 5

Modifying the DOM

- The append() method takes as a parameter an HTML string, a DOM object, or a jQuery object. That object is then added as the last child to the element(s) being selected.

HTML Before
```
<div class="external-links">
    <div class="linkOut">
        funwebdev.com
    </div>
    <div class="linkIn">
        /localpage.html
    </div>
    <div class="linkOut">
        pearson.com
    </div>
<div>
```

jQuery append
```
$(".linkOut").append(jsLink);
```

HTML After
```
<div class="external-links">
    <div class="linkOut">
        funwebdev.com
    <a href='http://funwebdev.com'
    title='jQuery'>Visit Us</a>
    </div>
    <div class="linkIn">
        /localpage.html
    </div>
    <div class="linkOut">
        pearson.com
    <a href='http://funwebdev.com'
    title='jQuery'>Visit Us</a>
    </div>
<div>
```

Slide 6

- The prepend() and prependTo() methods operate in a similar manner except that they add the new element as the first child rather than the last.

HTML Before
```
<div class="external-links">
    <div class="linkOut">
        funwebdev.com
    </div>
    <div class="linkIn">
        /localpage.html
    </div>
    <div class="linkOut">
        pearson.com
    </div>
<div>
```

jQuery append
```
$(".linkOut").prepend(jsLink);
```

HTML After
```
<div class="external-links">
    <div class="linkOut">
    <a href='http://funwebdev.com'
    title='jQuery'>Visit Us</a>
        funwebdev.com
    </div>
    <div class="linkIn">
        /localpage.html
    </div>
    <div class="linkOut">
    <a href='http://funwebdev.com'
    title='jQuery'>Visit Us</a>
        pearson.com
    </div>
<div>
```

```
$("#custName").change(function() {          example: JQueryForm.html
  if (!validName())
    $("#error").html("error message here....");
});
$("#phone").change(function() {
  if (!validPhone())
    $("#error").html("error message here....");
});
$("form").submit(function(){
  if (!validName() || !validPhone())
    $("#error").html("<br> Fix your inputs.");
  return validName() && validPhone();
});
```

```
function validPhone(){
  var pos = $("#phone").val().search(/^\d{3}-\d{3}-\d{4}$/);
  return (pos != -1);
}
function validName(){
  var pos=$("#custName").val().search(/^[A-Z][a-z]+, [A-Z][a-z]+$/);
  return (pos != -1);
}
```
13

In-class exercise

- Modify "JQuery-ex1.html". You must use JQuery. Refer to the example JQueryForms.html

Form validation Example

| ••• | Re-enter Password | Reset | Submit |

Enter both passwords

Form validation Example

| ••• | ••• | Reset | Submit |

Your two passwords are not same

14

3