Lecture 26

jQuery

Why jQuery

➢ Some aspects of JavaScript grow tiresome to us programmers.

➢ For starters ….

  ➢ Querying the DOM for objects

  ➢ Connecting events to call backs

  ➢ Applying functions to collections

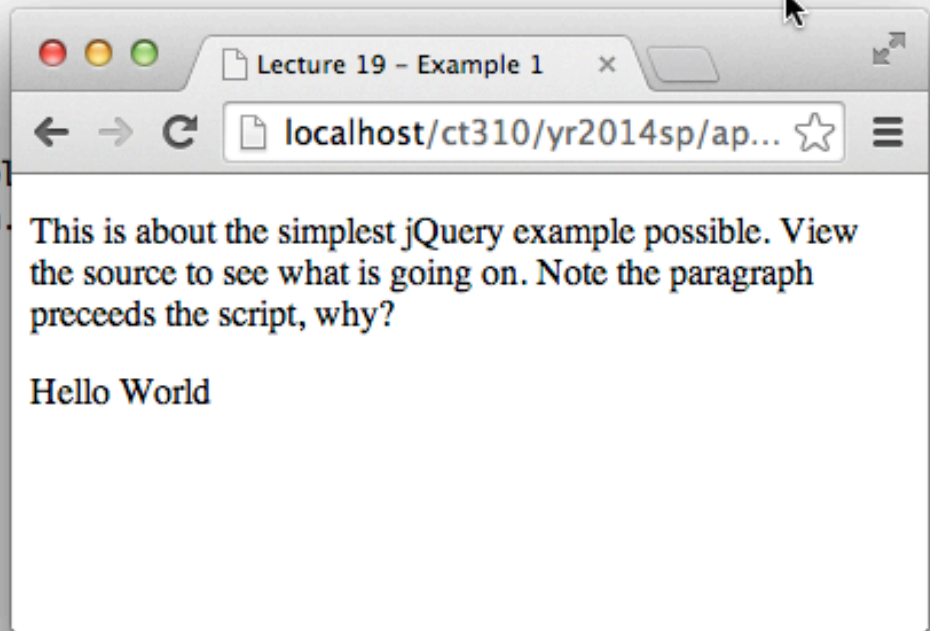 ➢ Asynchronous client-server interaction

What is jQuery?

➢ Too big a question for one slide, but

➢ In practice it acts much like a language extension

   ➢ But not strictly true characterization

➢ Hence, first step is to connect …

   ➢ A new syntax to …

   ➢ A new set of behaviors

Starting by Example

➢ Hello World in jQuery (2014)

```
 3  <head>
 4  <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
 5  <title>Lecture 19 - Example 1</title>
 6  <script src="http://ajax.googleapis.com/ajax/libs/jquery/1.11.0/jquery.min.js"></script>
 7  </head>
 8  <body>
 9  <div id="contents">
10  <p>
11  This is about the simplest jQuery exampl
12  View the source to see what is going on.
13  the script, why?
14  </p>
15  <p id="message">Go Away!</p>
16  <script>
17      $( "#message" ).text("Hello World");
18  </script>
19  </div>
20  <!--  end of the page contents -->
21
22  </body>
23  </html>
```

Lecture 19 – Example 1   ×

← → C   🔒 localhost/ct310/yr2014sp/ap... ☆ ≡

This is about the simplest jQuery example possible. View the source to see what is going on. Note the paragraph preceeds the script, why?

Hello World

But Wait - 2016

Even before we touch jQuery, we need to talk briefly about http and https.

Getting Adjusted to SSL

➢ There are 3 URL forms in this context.

    1. Start with http:// - this failed!

    2. Start with https:// - this will work

    3. Start with // - this is non committal

➢ Which is best? (think about it …)

➢ For example01 they are:
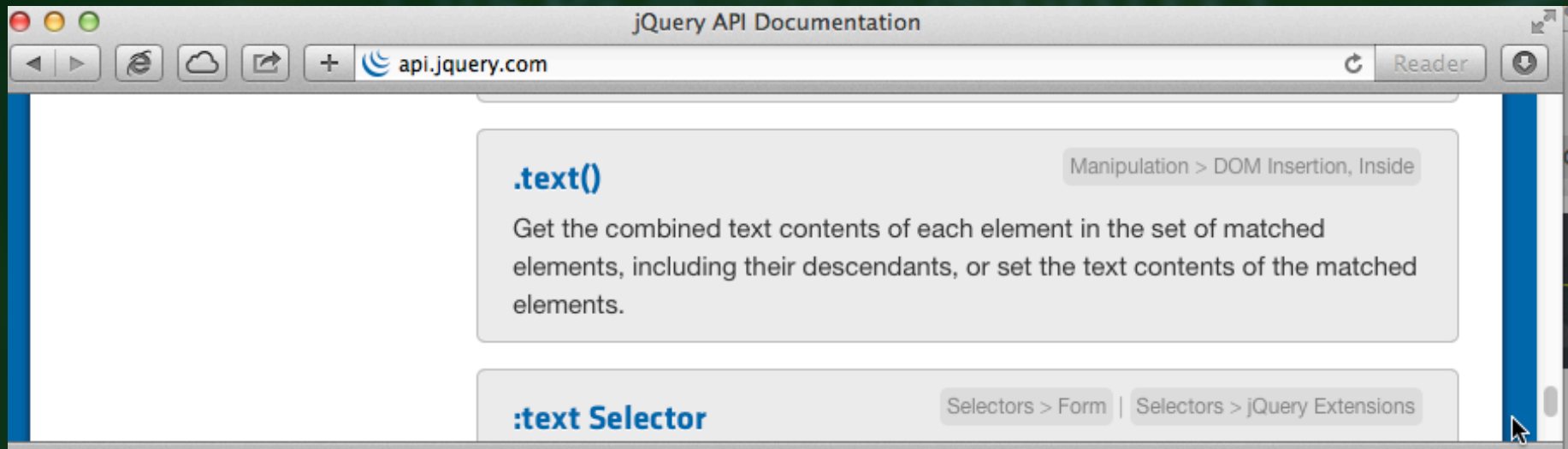
```
<script src="http://ajax.googleapis.com/ajax/libs/jquery/1.11.0/jquery.min.js">
</script>

<script src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.0/jquery.min.js">
</script>

<script src="//ajax.googleapis.com/ajax/libs/jquery/1.11.0/jquery.min.js">
</script>
```

Back to jQuery - Selectors

➢ Here is way to think about syntax
  ➢ $ (selector) . action
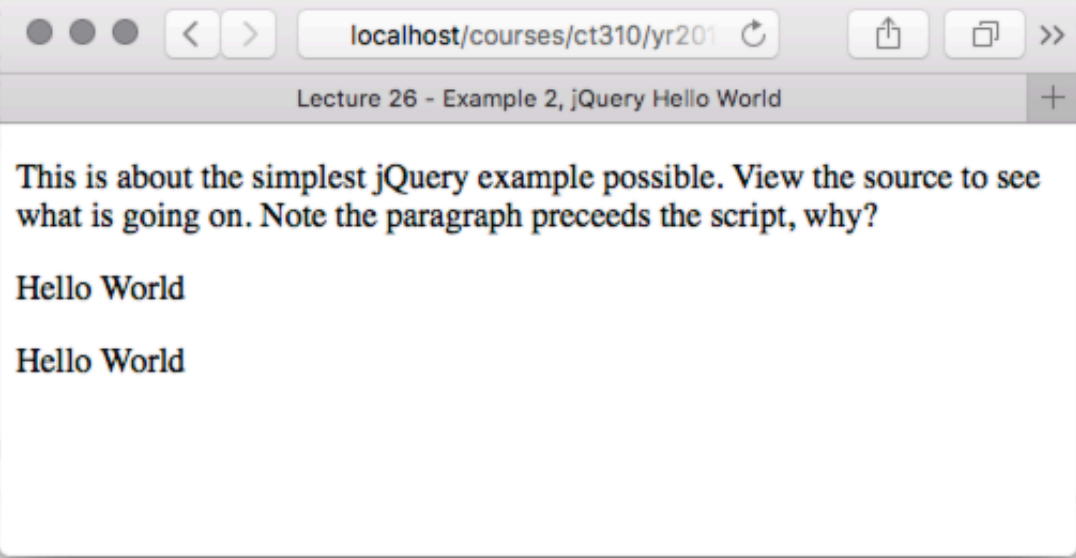➢ Think of selector as DOM query.
➢ Large set of actions/methods defined.

jQuery API Documentation

api.jquery.com   C   Reader

.text()   Manipulation > DOM Insertion, Inside

Get the combined text contents of each element in the set of matched
elements, including their descendants, or set the text contents of the matched
elements.

:text Selector   Selectors > Form | Selectors > jQuery Extensions

The text method

Already Powerful – A Twofer

Pay attention to the fine print on text, it can operate on sets.


```
 4 <meta http-equiv="Co...                              .../>
 5 <title>Lecture 26 - ...
 6 <script src="//ajax.g...                          ...in.js">
 7 </script>
 8 </head>
 9 <body>
10 <div id="contents">
11 <p>
12 This is about the sin...
13 View the source to se...                              ...s
14 the script, why?
15 </p>
16 <p class="message">Seriously Folks</p>
17 <p class="message">Seriously Folks Again</p>
18 <script>
19    $( ".message" ).text("Hello World");
20 </script>
```

Browser window showing:

localhost/courses/ct310/yr201

Lecture 26 - Example 2, jQuery Hello World

This is about the simplest jQuery example possible. View the source to see what is going on. Note the paragraph preceeds the script, why?
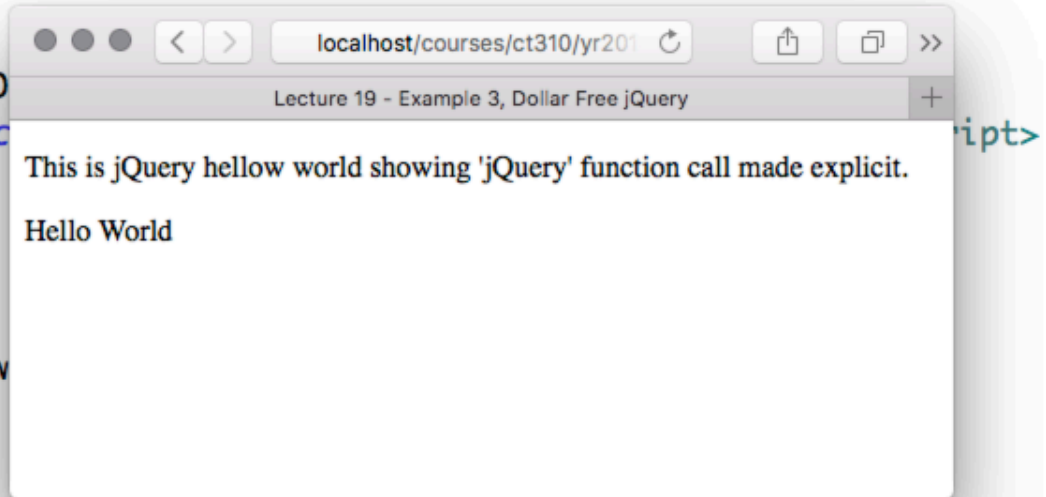
Hello World

Hello World

Detail – Dollars = Conflict

➢ The concise syntax has many virtues

➢ But, sometimes library conflicts arise over '$' and so, use 'jQuery' instead.

➢ This also is helpful in terms of understanding.

➢ jQuery is a library and it defines a single very powerful function 'jQuery'.

Dollar Free jQuery

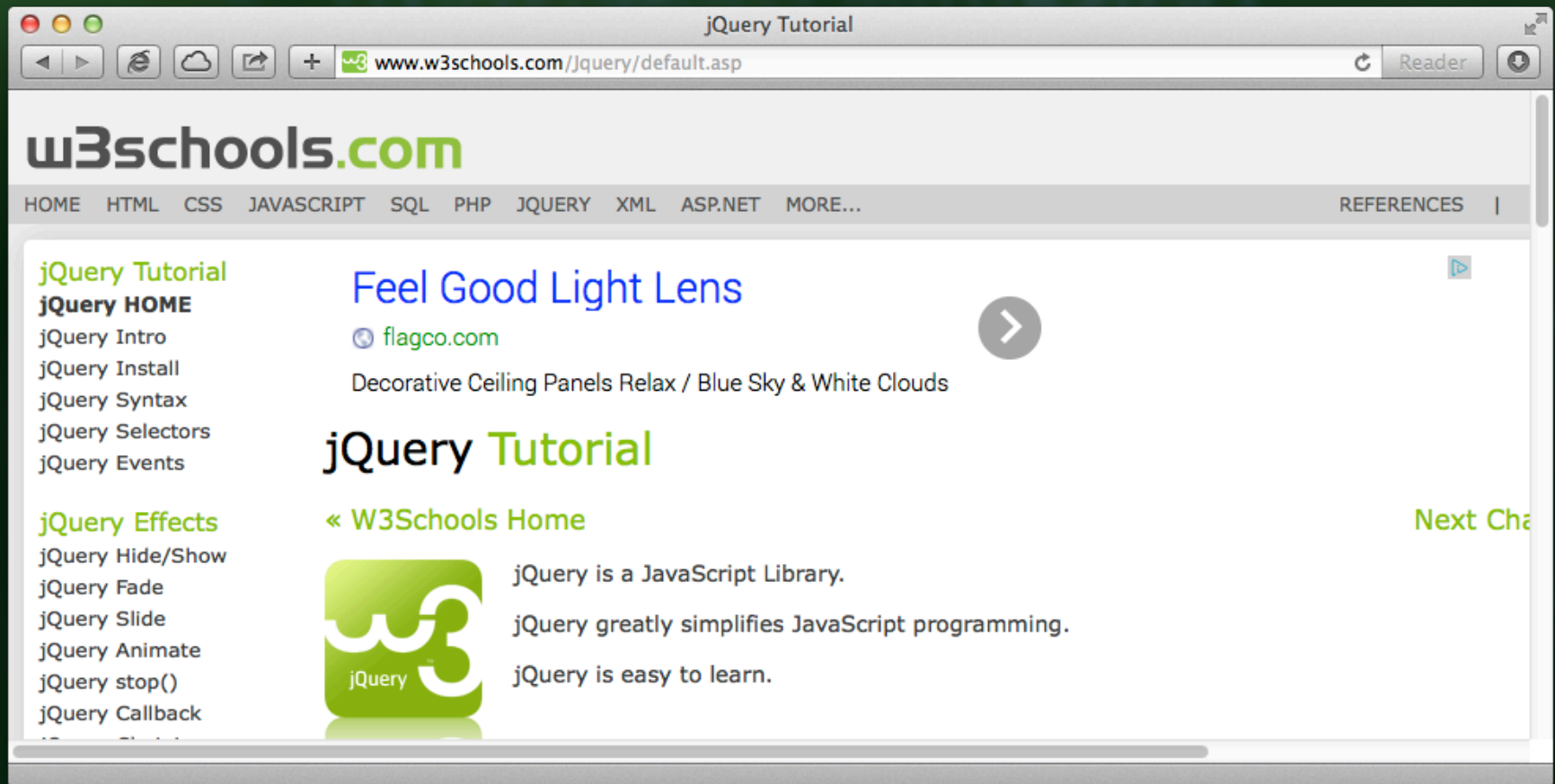➢ For compatibility with other libraries that may want prior claim on '$'

```
 3 <head>
 4 <meta http-equiv="Content-Type"
 5 <title>Lecture 19 - Example 3, D
 6 <script src="//ajax.googleapis.c                    ipt>
 7 </head>
 8 <body>
 9 <div id="contents">
10 <p>
11 This is jQuery hellow world show
12 </p>
13 <p id="message">Go Away!</p>
14 <script>
15    jQuery( "#message" ).text("Hello World");
16 </script>
```

localhost/courses/ct310/yr20

Lecture 19 - Example 3, Dollar Free jQuery

This is jQuery hellow world showing 'jQuery' function call made explicit.

Hello World

Credit Where Credit …

➢ w3schools jQuery tutorial is good.

Hide, Show & Buttons

➢ Pay attention to ready function.

```
 7  <script>
 8  $(document).ready(function(){
 9      $("#lassie_button").click(function(){
10        $("#snoopy").hide();
11        $("#lassie").show();
12      });
13      $("#snoopy_button").click(function(){
14        $("#snoopy").show();
15        $("#lassie").hide();
16      });
17    });
18  </script>
19
20  </head>
21  <body>
22  <div id="contents">
23  <p>
24  Everyone knows my favorite dog is
25  <span id="snoopy">Snoopy</span><span id="lassie">Lassie</span>.
26  </p>
27  <p>
28  <button id="snoopy_button">Pick Snoopy</button>
29  <button id="lassie_button">Pick Lassie</button>
30  </p>
31  </div>
32  <!--  end of the page contents -->
```

Lecture 19 – Example 3, S... ×
localhost/ct310/yr2014sp/ap...

Everyone knows my favorite dog is SnoopyLassie.

Pick Snoopy   Pick Lassie

Lecture 19 – Example 3, S... ×
localhost/ct310/yr2014sp/ap...

Everyone knows my favorite dog is Lassie.

Pick Snoopy   Pick Lassie
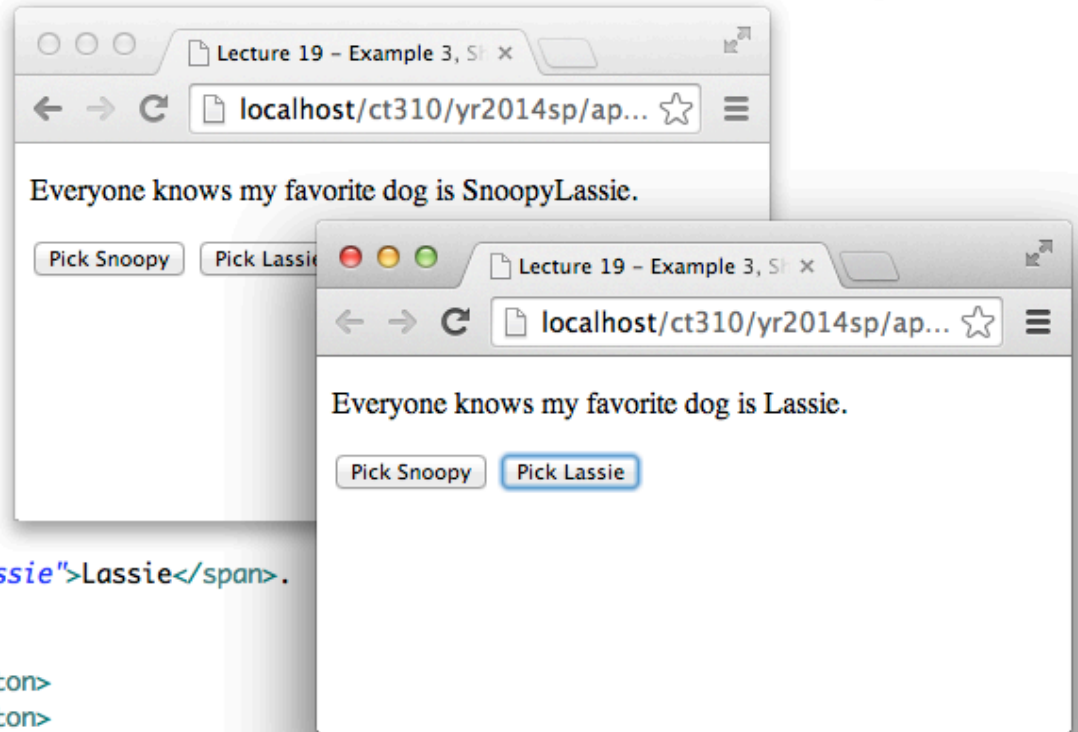
Example 5 – Table Cells

➢ Define a table on server – SQLite

➢ Edit cells on client

➢ Update table whenever a cell changes

➢ Puts a lot of pieces together:

    ➢ HTML, CSS, PHP,

    ➢ SQL,

    ➢ JavaScript, AJAX through jQuery

Changes Persistent

Pay Attention To …

➢ PHP Code to create HTML for table

➢ Note a CSS id encodes table cell

```
14  <table id="dogtable">
15     <tr> <th>Name</th> <th>Creator</th> <th>Person</th></tr>
16     <tr>
17     <td><input type="text" id="dogs0001name" value="Snoopy"
18     <td><input type="text" id="dogs0001creator" value="Charl
19     <td><input type="text" id="dogs0001person" value="Charli
20     </tr>
21     <tr>
22     <td><input type="text" id="dogs0002name" value="Gromit"
```

➢ Probably better ways in complex systems, but this is clear.

Also, onchange call back

➢ There is only one call back function

➢ Two arguments
  ➢ The id of the input text field
  ➢ The value of the field after change.

```
value="Snoopy" onchange="newval(this.id, this.value)"></td>
or" value="Charles Schultz" onchange="newval(this.id, this.valu
n" value="Charlie Brown" onchange="newval(this.id, this.value)"


value="Gromit" onchange="newval(this.id, this.value)"></td>
or" value="Nick Parks" onchange="newval(this.id, this.value)"><
n" value="Wallace" onchange="newval(this.id, this.value)"></td>
```

CSU CT 310, Web Development, ©Ross Beveridge

Simpler using jQuery

➤ Post functionality of jQuery.

➤ Cell input field id (cid) sent to the URL responsible for making change.

➤ So too is the new value.

➤ Bit tricky – think about "cid:cid"

```
1
2  function newval(cid, val) {
3      $("#outp1").text("Cell id " + cid + " now has value " + val);
4      $.post("lec19ex04up.php", { cid:cid, val:val },
5          function(data,status){
6              $("#outp2").html("Return from AJAX <br/>\nData: " + data + "<br/>\nStatus: " + status);
7      });
8  }
```

URL Handling Update

```php
1  <?php
2  if (! isset ( $_REQUEST ['cid'] )) {
3     die ( "Wow, that is not right." );
4  }
5  $cid = $_REQUEST ['cid'];
6  if (isset ( $_REQUEST ['val'] )) {
7     $val = strip_tags($_REQUEST ['val']);
8  }
9  try {
10    $dbh = new PDO ( "sqlite:./lec19ex04.db" );
11 } catch ( PDOException $e ) {
12    echo $e->getMessage ();
13 }
14 $row = intval(substr($cid, 4, 4));
15 $fld = substr($cid, 8);
16 $cmd = "UPDATE dogs SET $fld=\"$val\" WHERE dog_id = $row";
17 $dbh->exec($cmd);
18 echo $cmd;
19 ?>
```

Comments

- ➢ Security
  - ➢ Terrible! Simple example is not tied to prior user authentication.

- ➢ Virtue, hints at …
  - ➢ Convenience of jQuery
  - ➢ Immediacy of moden web apps

Load URL /

Submit Form Data /

Refresh Page