Web and Mobile Application Development

jQuery

Material created by:
David Augenblick, Bill Mongan, Dan Ziegler, Samantha Bewley, and Matt Burlick

What is jQuery?

- jQuery is a popular browser manipulation tool.
    - Created in 2006
- It’s built on top of JavaScript, to help users quickly find elements of interest in the HTML code and either get or set various attributes.
    - Along with making various other things simpler.

jQuery

- For example, maybe we want all paragraphs within `<p>` tags to hide themselves when clicked upon.
- Here's how easy that can be with jQuery!

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

- About `$(document).ready`
  - Ensures that code cannot run until the DOM for that page has been created
  - It is a jQuery handler akin to the `windows.onload` event handler

jQuery

- How do we get/use/install jQuery?
- There's two ways:
  1. Download it from jquery.com then include it locally:
  2. Or include it remotely from someone/somewhere (Google in this case):
- For simplicity let's do the second way

```
<head>
   <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.2.1/jquery.min.js">
   </script>
</head>
```

jQuery – some common selectors

- Here’s some more examples of using jQuery is to be able to access parts of a page/DOM easily
  - `$(this)` Selects the current HTML element
  - `$("p")` Selects all HTML elements with the <p> tag.
  - `$(".test")` Selects all elements with class="test".
  - `$("#test")` Selects all elements with id="test".
  - `$("*")` Selects all elements

jQuery – some common selectors

- And there’s lots more…
  - `$("p.intro")` Selects all `<p>` elements with class="intro"
  - `$("p:first")` Selects the first `<p>` element
  - `$("ul li:first")` Selects the first `<li>` element of the first `<ul>`
  - `$("ul li:first-child")` Selects the first `<li>` element of every `<ul>`
  - `$("[href]")` Selects all elements with an `href` attribute
  - `$("a[target='_blank']")` Selects all `<a>` elements with a `target` attribute value equal to "_blank"
  - `$("a[target!='_blank']")` Selects all `<a>` elements with a `target` attribute value NOT equal to "_blank"
  - `$(":button")` Selects all `<button>` elements and `<input>` elements of `type="button‟`
  - `$("tr:even")` Selects all even `<tr>` elements
  - `$("tr:odd")` Selects all odd `<tr>` elements
- We can then use the `.html` method of the object to change the innerHTML

```
$(“#testID”).html(“Here we go!”);
```

Selecting Items with jQuery (Cont.)

- There are many other selectors that can be used. For example, you can look for items that are only visible or checkboxes that are checked.

- W3 provides a list of all of the available selectors. Take a look at them and consider using them to make your queries more precise!

```
// visible div’s
$(“div:visible”);

// checked checkboxes
$(“#input:checked”);

// odd rows of a table
$(“tr:odd”);

// even rows of a table
$(“tr:even”);

// all empty input elements
$(“input:empty”);

// all elements with an HREF attribute
$("[href]");

// all input elements with a type “password”
$(":password");
```

http://www.w3schools.com/jquery/jquery_ref_selectors.asp

Example

```
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.2.1/jquery.min.js">
</script>
</head>
<body>
<div bgcolor="WHITE">
<button id=”collapseAll”> Collapse All Text </button>
<h1 class=”funSize”>This is a size "1" heading</h1>
<h2>This is a size "2" heading</h2>
<h3>This is a size "3" heading</h3>
<h4>This is a size "4" heading</h4>
<h5>This is a size "5" heading</h5>
<h6>This is a size "6" heading</h6>
</div>
<script>
    $(“h2”).html(“This is NOT a size ‘2’ header.”);
    $(“.funSize”).html(“You’ve been jQueried!”);
    $(“#collapseAll”).html(“Collapse NO Text”);
</script>
</body>
</html>
```

Common jQuery Functions

- jQuery has some commonly used functions and methods that you'll probably use to edit your web content. Here are some common ones:

- **html()** - gets/sets the HTML contents (like `divs, spans`, etc..)
- **val()** – gets/sets the content of a input element (like `input, textarea`, etc..)
- **prop()** - gets/sets an HTML tag property
- **click()** - sets a click event for the search results
- **on()** - sets an event for the search results
- **hide()** – hides the object
- **show()** – shows the object

.each()

- We’ve seen that jQuery selection can allow for getting back several things
- Often we’ll want to go through *each* of them.
- To do this we’ll use the `.each()` function
- Example:

```
// for each element that has a class “google”, alert us of the
// inner contents of the HTML
$.each($(“.google”), function() {
        alert($(this).html());
});
```

  - The bolded and blue portion of the code is the handler function. Notice that **this** is representing each element that has a “google” class in it.

.each()

- This can be particularly useful when going through JSON data we got that contains arrays of things:

```
<script>
var js={"data": [0, 1, 3, 5]};
$.each(js.data, function (i,item){
    document.write(item);
});
</script>
```

jQuery – handling events

- As covered previously, events are web page activities such as:
  - Button click
  - Mouse click
  - Hover over an image
  - Page loads (process startups)
- And we’ve seen how we can create/set event handlers to respond to any of these events using regular JavaScript

Handlers (Cont.)

- In jQuery, one way to set a handler for an object is to use the method

```
.on(eventType, function)
```

- Example:

```
// when the clearAllCategories button is clicked, clear all
// of the input elements with a google class
$(“#clearAllCategories”).on(“click”, function() {
    $(“input.google”).html(“”);
});
```

Handlers (Cont.)

- A second (overloaded) version of this method allows us to pack data into a JSON object to be passed to an outside function

```
.on(eventType, data, function)
```

- Example:

```
function myfunc(event){
   alert(event.data.val);
}
// when the clearAllCategories button is clicked, create a JSON object
// and pass it to the function myfunc as an attribute of the
// input parameter
$(“#clearAllCategories”).on(“click”,{val:”uh oh”}, myfunc});
```

Events in jQuery

- Or we can just call the event method on the object and pass it a function:

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

- Other supported event methods include:
  - `dblclick()`
  - `mousedown()`
  - `mouseup()`
  - `hover()`
  - `focus()`
  - `blur()`
  - And many more…

Animation in jQuery

- One cool thing that you can do with jQuery is animation!
  - The `.hide()` method is overloaded so you include *duration*
- Example: Sliding panels for information bars

```
$("#flip").click(function(){
    $("#panel").slideDown();
});
```

- There's also a `slideUp()` function, and, perhaps more useful, a `slideToggle()` function!

JSON Parsing

- When we were talking about JSON we said often web services return a textual representation of JSON data.
- Therefore we need to *parse* it into an actual JSON object.
- Fortunately jQuery gives us a function for that!

```
var jsonObject = $.parseJSON(jsonString);
```

- Note: As with most parsers, this is very finicky about things being in a proper/correct JSON format…

References

http://www.w3schools.com/jquery/default.asp