

More jQuery

Dr. Stephen Blythe
CSCI0022, Fall 2020

Modifying Selected jQuery Objects

- built in jQuery functions allow for access/modification of a jQuery object:
 - `text()` // returns a string holding the tag's text contents
 - `html()` // returns a string holding the tag's HTML contents
 - `text("new contents")` // replaces tag's contents with specified text
 - `html("<i>new</i> HTML")` // replaces tag's contents with specified HTML
 - `append("new contents")` //adds content inside of the end of the tag
 - `prepend("new contents")` //adds content inside of the beginning of the tag
 - `after("new contents")` //adds content just after the end of the tag
 - `before("new contents")` //adds content just before the beginning of the tag
 - `remove()` //removes tag from document
 - `replaceWith("new contents")` //replaces tag with specified content
 - `val()` // extracts contents of input tag (ex. a text box user typed into)

Don't Panic!! we'll see use(s) of these in class examples soon.

- built in jQuery functions allow for hiding/showing jQuery objects:
 - `hide()` // makes a (visible) jQuery object disappear from the document
 - it is still in the document, just no longer “visible”
 - `show()` // makes a hidden jQuery object visible again
 - `fadeOut()` // “fades” a jQuery object away.
 - `fadeIn()` // “fades” a jQuery object into visibility
 - `slideUp()` // “slides” a jQuery object up (away).
 - `slideDown()` // “slides” a jQuery object down (back to visibility)

Don't Panic!! we'll see use(s) of these in class examples soon.

Some Built-in jQuery “Variables”

- `$(document)` // refers to the HTML document

```
function doStart()
{
    alert("Page has loaded completely!");
}

$(document).ready(doStart); //ensures doStart() is called after all HTML has loaded
```

- `$(this)` //refers to object associated with event

```
function showText()
{
    alert("The text you clicked is:" + $(this).text());
}

$("#otherlabel").click(showText);

$("#mylabel").click(showText);
```

```
<!-- some useful HTML for this example-->
<b id="mylabel">The first line </b>
<i id="otherlabel">Another line </i>
```

WARNING!!! : `$(this)` has scope only in callback function!!!

Selection of Multiple Tags

- You may also select tag objects by naming their type:
 - `$("table")` // selects *all* table tags
 - `$("b")` // selects *all* bold tags
- So, consider these examples:
 - `$("b").click(myHandler);` //all bold text calls myHandler when clicked!!
 - `$("div").click(myHandler);` //any clicked div tag calls myHandler !!!
- You can do things to many tags at one time with this:
 - `$("b").hide();` // hides ***all*** bold tag contents!! -- hide() discussed later

The class selector

- any HTML tag can have a `class` attribute
 - For example: `<tr class=homeInning> ... </tr>`
- You can select all tags of one class with “`.<class_name>`” (note the period)
 - For the above example:
`$(".homeInning")`
 - would select all tags of class `homeInning`
- Of course, such selections usually match more than one tag
 - ... so you have to make sure to process the results carefully.
 - we'll discuss this more later ...

The attribute value selector

- any HTML tag can have (possibly user defined) attributes
 - For example: `<tr inningNum=9> ... </tr>`
- You can select all tags matching an attribute value with
 - “[`<attrName>=<value>`]”
- For the above example:
`$ (“[inningNum = 9]”)`
 - would select all tags with attribute `inningNum` having value 9
- Of course, such selections often match more than one tag
 - ... so you have to make sure to process the results carefully.
 - we'll discuss this more later ...

Anonymous Functions

- You can write a “temporary” function without a name!
- jQuery coders use this when a “callback” function is very simple
 - ... or when they want to pass a parameter to a handler function

```
$("#otherlabel").click( function(){alert("very simply:" + $(this).text() ); } );
$("#mylabel").click( function(){ showText( $(this) ); } );
function showText(obj)
{
    alert("The text you clicked is:" + obj.text() );
}
```

Anonymous Functions

- Remember, spacing of code doesn't matter:

```
$(" #myID").click(  
    function() // no name!  
    {  
        alert("This is an alert!");  
    }  
);
```

- So, the following is identical (without the in-line comment):

```
$(" #myID").click(function(){alert("This is an alert!");});
```

- Many JavaScript programmers do this for “short” callbacks!

each()

- Sometimes you will want to code something specific for many elements
 - ex. when a selector picked many tags ...
- You need a way to tell JavaScript this ... the each() function does this
- Example:

```
var count=0;  
$("b").each(function()  
{  
    var num = Number($(this).text()); // use <b> tag contents as a number  
    if (num>0)  
        count++;  
}  
);  
alert("There were " + count + "positives in bold!");
```

Navigating the DOM

- DOM is short for Document Object Model
- It describes the structure of any HTML code
- JavaScript can find tags relative to current objects in the DOM:
 - `.find(<selector>)` // inside of current tag only, look for this selector
 - `.next()` // find next tag after current one
 - `.siblings(<selector>)` // get tags at same level as current that match selector
- Note that these are likely to return many tag objects!

Unbinding a callback

- Sometimes, you want to remove a callback
- This can be done with the unbind() call

```
<strong id=clickMe> Some Text </strong>

...
function secondClick()
{
    alert("Not the first click!");
}
function firstClick()
{
    alert("Ouch - this is the first click!");
    $(this).unbind();
    $(this).click(secondClick);
}
$("#clickMe").click(firstClick);
```

The .css() jQuery Function

- CSS is a powerful method of specifying the format of web pages.
- CSS is a collection of properties about each tag in a web page.
- Each CSS property has both:
 - a name (for example: “color”)
 - a value (for example: “red”)
- jQuery can tell you the value for a CSS property of a given tag:
 - `var myColor = $("#someTag").css("color");`
- jQuery can also change the value of a CSS property for a given tag:
 - `$("#someTag").css("color", "red"); // changes text color to red.`
- There is much more that CSS can change: font, location, size, ...
 - CSC34400, AAD36100, AAD36200, ... all discuss more about CSS.