Simple jQuery

Dr. Stephen Blythe
CSC10022, Fall 2020

The Event Concept

- An *event* is some sort of user interaction:
  - mouse clicked
  - mouse moves over something
  - mouse moves away from (leaves) something
  - mouse hovers over something
  - user text is entered
  - document completes loading
- Code is then written to handle the event
  - code is specified as a function (referred to as “callback” function).

Event Handling in JavaScript

- In JavaScript, event handling is set up by:
    - identifying/selecting the HTML tag(s) to “watch” for events
    - naming the events you want to handle
    - specifying the function to execute when this combination “fires”
- In pure JavaScript, this gets complicated …
- … most web programmers now use library like jQuery instead:
    - jQuery is a set of pre-built functions and utilities
    - eases HTML tag selection
    - simplifies event naming
    - provides built in handlers to do common tasks
    - included through a standard `src` attribute in a `script` tag
        - `src=http://code.jquery.com/jquery.min.js`

Event Handling with jQuery

- jQuery for selecting HTML tag(s):
    - $("#<idname>") // replace <idname> with an HTML tag's ID
    - returns a jQuery "object" containing the tag's contents
    - the # says look for the following ID.
    - other ways to select (which we will not use immediately) include:
        - selecting objects by type
        - selecting objects by class
- Examples:
    - `var myTagObject = $("#myTable");`
    - `var otherObject = $("#currScore");`

Event Handling with jQuery...

- jQuery to use a jQuery object to register a handler:

- append a *.method_name()* to the object
    - place a function name within the ()’s
    - do not call the named function, just list its name
- some *method_name* values
    - click()
    - mouseover()
    - mouseleave() //avoid mouseout unless REALLY careful!!!
- Example:

```
function myHandler() {alert (“Fired!”);}
var myTagObject = $(“#myTable”);
myTagObject.click(myHandler);
```

Accessing or 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
  - val() // extracts contents of input tag (ex. a text box user typed into)
  - append() // adds to the existing tag contents (at the end)

There are MANY more of these, but the above will get you started.

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!!!