Aside: Should check out a few chapter of the older jQuery tutorial, in particular the parts about the DOM and cookies but probably one or two other chapters as well that weren't covered adequately in Allerdice's course, though likely they won't be covered in substantially greater depth in the Dori Smith class either.

1. OVERVIEW OF JQUERY

Works across all modern browsers...abstracts browser specific features allowing you to concentrate on design.

Focuses on manipulating page content (the DOM)

Simplifies working with modern browser event model

Adding sophisticated effects like animations and transitions

Common patterns

page loads: setup on response to load event

event -> retrieve content -> manipulate it -> put the content back on the page

Leverages you existing knowledge of CSS

Works with sets of elements

Performs multiple operations on a set of elements with one line of code

(known as statement chaining)

Hides browser quirks (so you can concentrate on end results).

Is extensible so you can use 3rd party plugins to perform specialized tasks or you write your own.

Compatible with modern browsers but there are/were a few known issues

DOWNLOADING AND INSTALLING JQUERY

During development use the development version if you need to debug. When you deploy use the min version for fast downloads. Functionality is same in both versions but size of min version is much smaller.

download at jquery.com

CREATING A SIMPLE JQUERY ENABLED PAGE

```
<!DOCTYPE html>
<html>

<script>
/* without jquery we might do this to run after load is finished */
/*
function runOnLoad() {
       alert("the page just loaded!");
}
window.onload = runOnLoad;
*/
</script>

<script src="js/jquery-1.9.1.js"></script>
<script>
/* we can also pass a named function
   this is considered more succinct and concise */
       $("document").ready( function () {
               alert("the page just loaded!");
       });
</script>

<head>
<meta content="text/html; charset=utf-8" http-equiv="Content-Type" />
</head>
<body>
</body>
</html>
```

ANONYMOUS FUNCTIONS

See lynda.com class: Practical and Effective JavaScript

the $("document").ready executes when the dom of the page has loaded and is ready for use rather than waiting for all page content to finish loading. Can call this same function multiple times without problem.

OVERVIEW OF JQUERY FEATURES

* core functionality - core functions and utilities
* selection and traversal - finding content and navigating the content
  this is the query in the jQuery name!
* Manipulation and CSS - functions for editing and changing content and working
  with CSS data such as positioning
* Events - simplifies working with modern DOM events & provides common event helper functions
* Effects - functions for creating basic animations and effects such as hiding and showing
  elements and moving objects around
* Ajax category - utilities for working with Ajax such as loading content from pages
  and dealing with JSON data
* User Interface plugin for commonly used interface widgets like slider controls, progress
  bars, accordions, etc.
* Extensibility

Wont' be covering Ajax or Extensibility

2. RETRIEVING PAGE CONTENT

OVERVIEW OF SELECTORS AND FILTERS

Selectors and filters:

jq selectors will return an array of objects that match the criteria

filters will refine the results array that the selector returns

The array that comes back is not a set of DOM elements. It is a collection of jq objects that provide a large number of predefined functions for further operating on the objects.

You can get access to underlying DOM objects if you want but the idea is to use jq objects and functions.

USING BASIC JQUERY SELECTORS

CSS selectors and filters are based on familiar CSS syntax, and work pretty much the same was as CSS does.

SELECTOR	PURPOSE
tagname	Finds all elements that are named tagname
#id	All elements with the given id
.className	All elements with the given class attribute
tag.className	
tag#id.className	
*	All elements on the page

Using jq selectors vs plain browser DOM

PLAIN	JQUERY
document.getElementsByTagName("p")	$("p"); // returns a list
document.getElementById("list1")	$("#list1");
requires a loop	$("li.a"); // all li tags with class a
requires a loop	$("ul .b"); // all class b within a ul element

The Hierarchy and Combination Selectors allow you to get more advanced

Select elements based on hierarchical relationships or on a series of common criteria

`selector, selector` - a comma delimited list, finds all specified selectors

`.class1.class2` - all elements with both .class1 and .class2 (space delimited list)

`parent>child` - all child elements that are direct children of elements of type parent `ancestor descendent` - all descendent elements that are contained within elements of type ancestor

`prev + next` - all next elements that are next to a prev element

`prev ~ siblings` - all siblings that come after prev and match the siblings selector

see HierCombo.html and default.html for examples of using selectors

USING BASIC JQUERY FILTERS

Filters work together with selectors and provide even more fine-grained control over element selections.

Six categories of filters:

1. Basic - first, last, even numbered, odd numbered
2. Content - whether an element contains a particular string
3. Visibility - tests visibility settings
4. Attribute - examines attributes of an element
5. Child filters - selects based on relationship to a parent
6. Form - special filters that operator on forms elements

FILTER	PURPOSE
:first	Only first instance in the returned set
:last	Only last instance in the returned set
:even	Only even numbered in the returned set
:odd	Only odd
:eq(n)	Filters out elements not positioned at given index
:gt(n)	Includes elements past the given index
:lt(n)	Includes elements before the given index
:header	All header elements (H1, H2, etc)
:animated	All elements being currently animated in some way
:not(selector)	All elements that do not match the given selector

see BasicFilters.html for examples

USING JQUERY ATTRIBUTE FILTERS

Further filter results of a selector statement based on attribute content

Note: substitute the term attribute with the actual attribute, for example $("a[href]")

```
FILTER                       PURPOSE
[attribute]                  Include if element has the attr
[attribute=value]            Include if element has the attr and value
[attribute!=value]           Include if element has the attr but not the value
[attribute^=value]           Include if element has attr and value that starts with value (RE!)
[attribute$=value]           Include if element has attr and ends with value (RE!)
[attribute*=value]           Include if element has attr and contains value (match)
[attrFilter1][attrFilterN]   Include elements that match all specified attr filters in the
                             specified list of attr filters
```

see AttrFilters.html for examples

CHILD, VISIBILITY, AND CONTENT FILTERS

```
CONTENT FILTER        PURPOSE
:contains(text)       Filters the selection to only include elements that contain the text string
:empty                Only empty elements
:has(selector)        Matches elements that contain at least one element that has
                      the specified selector
:parent               Matches all elements that are parents (elements with children)

VISIBILITY FILTER
:visible              Filters the selection to include only visible elements
:hidden               Only hidden elements

CHILD FILTER
:nth-child(index)     Matches elements at index, or even or odd increments, or who
:nth-child(even)      match an equation of the form Xn+M (for example, 2n or 3n + )
:nth-child(odd)
:nth-child(equation)
                      Note: the nth-child filters are not zero indexed,
                      they start at 1 for the first element

:first-child          Matches elements who are the first child of their parent
:last-child           Matches elements who are the last child of their parent
:only-child           Matches elements who are the only child of their parent
```

see ChildVisCount.html for examples of child, visibility, and content filters

FORM SELECTORS AND FILTERS

You can use form selectors to deal with form elements

They work like other selectors but start with a colon (:) like a regular filter

SELECTOR	PURPOSE
:input	Finds all input, select, textarea, and button elements
:text	Finds all text element
:password	Finds all password elements
:radio	Finds all radio button elements
:checkbox	Finds all checkbox elements
:submit	Finds all submit elements
:reset	Finds all reset elements
:image	Finds all image elements
:button	Finds all button elements
:file	Finds all file upload elements

You can perform addition filtering of form elements, such as whether items

are checked, selected, or enabled

SELECTOR	PURPOSE
:enabled	Matches all form elements that are enabled
:disabled	Matches all form elements that are disabled
:checked	Matches all form elements that are checked (radio buttons & check boxes)
:selected	Matches all elements that are selected

The above are convenience filters that help you select form elements that are in a certain state.

see FormSelectors.html for examples.

TRAVERSING DOCUMENT INFORMATION

FUNCTION/PROPERTY	PURPOSE
size(), length	The number of elements in the jQuery result set
get()	Returns an array of all matched DOM elements. Useful if you need to operate on the DOM elements themselves instead of using built-in jQuery functions.
get(index)	Access a single matched DOM element at a specified indiex in the matched set
find(expression)	Searches for descendent elements that match the specified expression
each(fn)	Execute a function within the context of every matched element

see traversing.html for examples.

JQUERY STATEMENT CHAINING

One of jQuery's most powerful features in its ability to chain multiple functions together to perform several operations in one line of code.

```
$(selector).fn1().fn2().fn3();
```

PRACTICAL EXAMPLE 1: ANNOTATING PAGE LINKS

We used jquery to add an icon to anchor tags whose href ends in .pdf

```
$("a[href$='.pdf']").after("<img src='images/pdf_icon_small.gif' align='absbottom' />");
```

see AutoPDFIcons.html for complete code

3. MANIPULATING PAGE CONTENT

CREATING, GETTING, AND SETTING CONTENT

jQuery has functions for creating, copying, deleting, and moving content around, as well as wrapping page content in other content.

jQuery provides cross-browser support for working with CSS, including positioning and sizing information.

* To create new HTML content, you simply pass a string containing new HTML to the $() function:

```
        var newHeader = $("<h1>My New Header</h1>");
        var myStr = "<h1>My New Header</h1>";
        var newHeader = $(myStr);
```

* You can use the html() and text() methods to get and set content on elements

FUNCTION	PURPOSE
html()	Returns the HTML content of the first matched element
html(newcontent)	Sets the HTML content of every matched element
text()	Returns the text content of the first matched element
text(newtext)	Sets the text content for all matched elements If you pass html as an argument to the text function, it will automatically escape the html so that it won't work as functional html

This section seems a bit weak but maybe it's all there is to know about creating new elements without adding them to the DOM.

The more important function introduced here is the $("") function. This section did not show many good ways to add a newly created element to the dom.

see creating.html for examples

MANIPULATING ATTRIBUTES

* To inspect or change the value of attributes on elements, use jQuery's attr functions

FUNCTION	PURPOSE
attr(name)	Accesses property on the first match element. This method makes it easy to retrieve a property value from the first matched element. If the element does not have an attribute with such a name, undefined is returned.
attr(properties)	Sets a series of attributes on all matched elements using an object notation syntax. This is best used for setting large numbers of properties at once. `$("img").attr({ src: "/image/hat.gif", title: "jQuery", alt: "jQuery Logo" });`
attr(key,value)	Sets a single property to a value on all matched elements
attr(key, fn)	Sets a single property to a computed value, on all matched elements Instead of supply a string value, a function is provided that computes the value of the attribute
removeAttr(name)	Removes the named attribute from all matched elements

see attributes.html for examples

INSERTING CONTENT

jQuery provides several functions for inserting content into the document both before and after existing page elements.

FUNCTION	PURPOSE
append(element)	Appends content to the inside of every matched element
appendTo(selector)	Appends all of the matched elements to another, specified, set of elements
prepend(content)	Prepends content to the inside of every matched element
prependTo(selector)	Prepends all the matched elements to another, specified, set of elements
after(content)	Inserts contents after each of the matched elements
before(content)	Inserts contents before each of the matched elements
insertAfter(selector)	Inserts all of the matched elements after another, specified, set of elements
insertBefore(selector)	Inserts all the matched elements before another, specified, set of elements

see inserting.html for examples

WRAPPING, REPLACING, REMOVING CONTENT

JQuery can wrap existing content in the page, replace content, copy content, and remove it.

FUNCTION	PURPOSE
wrap(html)	Wraps each matched element with the specified HTML content
wrap(element)	Wraps each matched element with the specified element
wrapAll(html)	Wraps all elements in the matched set with the specified HTML content seems to actual move all matched set elements into the same wrapper
wrapAll(element)	Wraps all the elements in the matched set into a single wrapper element
wrapInner(html)	Wraps the inner child content of each matched element (including text nodes) with an HTML structure
wrapInner(element)	Wraps the inner child contents of each matched element (including text nodes) with a DOM structure
replaceWith(content)	Replaces all matched elements with the specified HTML or DOM elements
replaceAll(selector)	Replaces the elements matched by the specified selector with the matched elements
empty()	Removes all child nodes from the set of matched elements
remove()	Removes all matched elements from the DOM
clone()	Clone matched DOM elements and selects the clones
clone(bool)	Clone matched DOM elements, and all their event handlers, and select the clones

When we wrap with html, it looks like we should be specifying only the complete start tag.

WORKING WITH CSS INFORMATION

jQuery's CSS functions provide easy, cross-browser access for setting properties and working with positioning and sizing information.

The css() function allows you to retrieve and set styles for a set of matched elements

FILTER	PURPOSE
css(name)	Returns the value for the named CSS property for the first matched element
css(properties)	Sets the CSS properties of every matched element using an object-notation syntax `var cssObj = { 'background-color' : '#ddd', 'font-weight' : ' ', 'color' : 'rgb(0,40,244)' }` `$(this).css(cssObj);`
css(property, value)	Sets a single style property to a value on all matched elements. If a number is provided, it is automatically converted into a pixel value, with the following exceptions: z-index, font-weight, opacity, zoom, and line-height

jQuery provides a set of functions for working with css classes on page elements. Classes can be easily added, removed, toggled, and detected

CSS FUNCTIONS	PURPOSE
addClass(class)	Adds the specified classes to each of the set of matched elements
hasClass(class)	Returns true if the specified class is present on at least one of the set of matched elements
removeClass(class)	Removes all the specified class(es) from the set of matched elements
toggleClass(class)	Adds the specified class if it is not present, removes the specified class if it is present
toggleClass(class,switch)	Adds the specified class if the switch is true, removes the specified class if the switch is false

WORKING WITH CSS POSITIONING

CSS FUNCTIONS	PURPOSE
offset()	Gets the current offset of the first matched elements, in pixels, relative to the document
offsetParent()	Returns a jQuery collection with the positioned parent of the first matched element
position()	Gets the top and left position of an element relative to its offset parent
scrollTop()	Gets the scroll top offset of the first matched element
scrollTop(val)	Sets the scroll top offset to the given value on all matched elements
scrollLeft()	Gets the scroll left offset of the first matched element
scrollLeft(val)	Sets the scroll left offset to the given value on all matched elements

WORKING WITH CSS SIZING INFORMATION

To retrieve cross-browser sizing information for elements, use the jQuery size-related functions.

CSS FUNCTIONS	PURPOSE
`height()`	Gets the current computed, pixel, height of the first matched element
`height(val)`	Sets the CSS height of every matched element
`width()`	Gets the current computed pixel, width of the first matched element
`width(val)`	Sets the CSS width of every matched element
`innerHeight()`	Gets the inner height (excluding the order and including the padding for the first matched element
`innerWidth()`	Gets the inner width (excluding the border and including the padding) for the first matched element
`outerHeight(margin)`	Gets the outer height (includes the border and padding by default) for the first matched element. If the margin argument is true, then the margin values are also included.
`outerWidth(margin)`	Gets the outer width (includes the border and padding by default) for the first matched element. If the margin argument is true, then the margin values are also included.

see css_sizing.html for examples.

4. WORKING WITH EVENTS

UNDERSTANDING THE JQUERY EVENT HANDLING FEATURES

Provides mechanism for working with events that is simpler than relying on the DOM.

Abstracts away the differences between browser implementations

Makes it easy to assign event handlers to groups of elements by using selectors and filters

Breaks down into a couple of categories

* Binding/Unbinding

  Allows events to be wired up and torn down in a cross browser way.

*	Unified Event Object

Provides an event object that exposes the most common properties in a cross-browser way

*	Convenience features

Provides functions that encapsulate common event features and cross-browser helper routines

BINDING AND UNBINDING EVENTS

*	Events are connected to and disconnected from elements using the bind() and unbind() functions

```
$(selector).bind(event, data, handler)
$(selector).unbind(event, handler)
```

BIND() PARAMETER	PURPOSE
event	Defines the event that you want to be bound to for each element in the selector's result set. Possible values are blur, focus, load, resize, scroll, unload, beforeunload, click, dblclick, mousedown, mouseup, mousemove, mouseover, mouseout, mouseenter, mouseleave, change, select, submit, keydown, keypress, keyup, error
data	Optional. Defines a piece of data that will be passed to the handler function when the event happens and the handler function is called.
handler	Specifies the function that will handle the event. If you plan to unbind you need to use a named function.

UNBIND() PARAMETER

event	Defines the event that you want to be disconnected for each element in the selector's result set. If you are unbinding, you need to use named function for the handler.
handler	Specifies the handler function that was defined to handle the event

see binding.html for examples

CONVENIENT HELPER METHODS

Several "helper" functions can perform common event-related tasks

```
$(selector).click(fn)
$(selector).hover(fnOver, fnOut)
$(selector).toggle(fn1, fn2, fn3, fn4...)
```

```
FUNCTION                      PURPOSE
```

`click(fn)`	Shortcut for click function handler. There are also shortcuts for: blur, change, dblclick, error, focus, keydown, keypress, keyup, load, mousedown, mouseenter, mouseleave, mouseout, mouseover, mouseup, resize, scroll, select, submit, unload
`hover(fnOver, fnOut)`	Help function for hover behavior. fnOver is the function to call when the mouse enters, fnOut for when the mouse leaves

Note: toggle method was removed from jQuery 1.9

`toggle(fn1, fn2, fn3,...)`	Helper function for implementing toggling behavior. jQuery will call each function on every other click, starting with fn1, then fn2, then fn3, etc.

see helpers.html for examples

USING THE JQUERY EVENT OBJECT

* Writing event-handling code is frustrating when it differs across browser

* The jQuery event object smoothes these differences and provides a single object with the most important properties

Most common functions:

PROPERTY	PURPOSE
type	Type of event ("click", e.g.)
target	Element that issued the event
data	Data passed to bind function
pageX, pageY	Coordinates of mouse when event happended, relative to document
result	Value returned by the last handler function
timestamp	Time when event occurred

METHOD	PURPOSE
preventDefault()	Prevents the browser from executing the default action
isDefaultPrevented()	Returns whether preventDefault() was ever called on this object
stopPropagation()	Stops the bubbling of an event to parent elements
isPropagationStopped()	Returns whether stopPropagation() was ever called on this object

see eventobj.html for examples

MISCELLANEOUS JQUERY EVENT FUNCTIONS

* For a couple of specialized tasks, jQuery provides some miscellaneous functions

```
$(selector).one(type, data, handler)
$(selector).trigger(event, data)
$(selector).triggerHandler(event, data)
```

FUNCTION	PURPOSE
one(type, data, handler)	Works the same as bind(), but the event handler is only ever executed one time for each matched element
trigger(event, data)	Triggers an event on every matched element. This will also cause the default action of the browser to be executed. For example, passing 'click' to the trigger function will also cause the browser to act as though the item were clicked
triggerHandler(event, data)	Triggers all bound event handlers on an element (for a specific event type) without executing the browser's default actions, bubbling, or live events. Only works on the first matched element in the result set for selector.

see miscevents.html for example

PRACTICAL EXAMPLE 3: TABLE STRIPING AND HIGHLIGHTING

see miscevents.html. I'm pretty sure this could have been done just with css.

5. JQUERY ANIMATION AND EFFECTS

HIDING AND SHOWING ELEMENTS

* jQuery library supplies some basic animation and effects functions that perform common visual effects

    Showing and hiding elements

    Fading elements in and out

    Moving elements around on the screen

* You can use the basic animation function to easily build your own animation effects

Showing/hiding elements is simple and can be done immediately or over a specified duration of time

FUNCTION	PURPOSE
show()	Displays each of the set of matched elements, if they are hidden
show(speed, callback)	Shows all matched elements using a graceful animation. Fires an optional callback after completion.
hide()	Hides each of the set of matched elements if they are shown.
hide(speed, callback)	Hides all matched elements using graceful animation. Fires an optional callback after completion.
toggle()	Toggles displaying each of the set of matched elements
toggle(switch)	Toggles displaying each of the set of matched elements based upon the switch (true shows all elements, false hides all elements)
toggle(speed, callback)	Toggles displaying each of the set of matched elements using a graceful animation and firing an optional callback after completion

speed can be slow, normal, fast or a number in milliseconds.

FADING ELEMENTS IN AND OUT

* Elements can be faded in or out completely or to a predetermined opacity level.
* The speed of the face can be specified as either a string ("slow", "normal", "fast") or a millisecond duration

```
fadeIn(speed, callback) - fades all matched elements by adjusting their opacity and firing an
                          optional callback when finished

fadeOut(speed, callback)

fadeTo(speed, opacity, callback) - opacity 0 = invisble, 1 = totally opaque (not faded)
```

see faceeffect.html for examples

SLIDING PAGE ELEMENTS

* The sliding effects is another way to reveal page elements in jQuery
* jQuery provides functions for sliding elements up or down, as well as toggling the slide animation

FUNCTION	PURPOSE
slideDown(speed, callback)	Reveals all matched elements by adjusting their height and firing an optional callback after completion
slideUp(speed, callback)	Hides all (as above)

```
slideToggle(speed, callback) Toggles all (as above)

"slow", "normal", "fast", number of milliseconds
```

see sliding.html for example

CREATING CUSTOM ANIMATIONS

* To create custom animation for many properties on page elements, call the aminate() functon

* To stop animations in progress, call the stop() function

```
FUNCTION                        PURPOSE
animate(params, duration,       Creates a custom animation
easing, callback)               params:  The properties on the elements to animate
                                duration: "slow" "normal" "fast"
                                easing: The type of easing: linear or swing
                                callback: optional callback function on completion
animate(params, options)        Creates a custom animation
                                params: The properties to animate
                                options: Set of options for the animation to take
                                Note:  Options were not explained and I couldn't find them
                                in the jQuery online documentation for the animate function

stop()                          Stops all the currently running animations on all the specified elements
```

see animating.html for examples

PRACTICAL EXAMPLE 4: IMAGE ROTATOR

Uses a timer function to cycle through images. Pile images one on top of another. Displays each one for 2 seconds, fades the top one for one second. Pulls the next image to the top by changing its class and its z-index. Uses a class selector to find the current image. Uses the next() method to find the next image. Checks to see if at end of child divs by getting the length of the next() image. If the length is zero, begins at the top of the list of divs by using the div:first selector. Fades the image by animating the opacity

see imagerotator.html for complete code.

6. USING THE JQUERY UI PLUG-IN

INTRODUCTION TO JQUERY UI

INTERACTIONS
Draggable
Droppable
Resizable
Selectable
Sortable

WIDGETS
Accordion
Datapicker
Progressbar
Dialog
Slider
Tabs

EFFECTS
Add Class
Remove Class
Toggle Class
Switch Class
Hide
Show
Toggle
Color Animation

EXPLORING THE UI WIDGETS

Accordion
- collapse content - can click the open panel to collapse it without opening another panel
- customize icons - use custom icons on accordion tabs
- fill space - resizes to fit size of outer container
- no auto height - sizes each tab to fit content
- open on hoverintent - opens tabs on hover
- sortable - drag and drop reordering of tabs

Datepicker
- Animation
- Dates in other months (first and last weeks)
- Display button bar
- Display inline
- Display month & year menus
- Display multiple months
- Format date
- Icon trigger

- Localize calendar
- Populate alternate field
- Restrict date range
- select a date range
- show week of the year

Dialog
- Animation
- Basic Model
- Modal confirmation
- Modal form
- Modal message

Progress Bar
- Custom label
- Indeterminate value

Slider Bar
- Colorpicker
- Multiple sliders
- Range slider
- Range with fixed maximum
- Range with fixed minimum
- Slider bound to select
- Slider scrollbar
- Snap to increments
- Vertical range slider
- Vertical slider

Tabs
- Collapse content (similar to Jerome's registration and login dropdowns)
- Content via Ajax
- Open on mouseover
- Simple manipulation
- Sortable
- Tabs at bottom
- Vertical Tabs functionality

EXPLORING THE JQUERY UI EFFECTS

jQueryUI offers the following UI effect categories that supplement jQuery's basic functionality.

Add Class
Adds class(es) to elements while animating all style changes.

Color Animation
Animate the properties of elements between colors.

Effect
Apply an animation effect to an element
Blind, Bounce, Clip, Drop, etc.

Hide
Hide elements using custom effects.
Blind, Bounce, Clip, Drop, etc.

Remove Class
Removes class(es) from elements while animating all style changes.

Show
Display elements using custom effects.

Switch Class
Add and remove class(es) to elements while animating all style changes.

Toggle
Display or hide elements using custom effects.

Toggle Class
Toggle class(es) on elements while animating all style changes.

Many of the effects allow a setting for easing. There are quite a few choices for an easing effect

linear
swing
easeOutBounce
easeInOutQuad
easeOutExpo
and more... http://api.jqueryui.com/easings/

USING THE JQUERY UI THEMEROLLER

Build a theme

Roll your own: uses a starting point based on your last selection from the gallery

Gallery - prebuilt themes

Download CSS styles & images by clicking on a button

DOWNLOADING AND INSTALLING JQUERY UI

Can build a custom download

PUTTING IT ALL TOGETHER

To illustrate how several of the features we've learned work in a real site, we'll take a site that was built without jquery and update it with some new features.

The accordion control

By default, uses link tag as it's header and paragraph as its content. We can change it by passing an argument to the constructor(?). Other options are available
$("element").accordion({ header: "h4" });

The image selector:

```
$("a:has(img.gallery)").click( function () {
      var largePath = $(this).attr("href");
      var caption = $(this).attr("title");
      $("#photo_large").attr( { src: largePath } );
      $("#caption1").text(caption);
      return false;
});
```

We're assigning the click on the a:has(img.gallery) elements - I don't know why he used :has

(:has matches elements that contain at least one element that has the specified selector) -

to do the following:

```
assign largePath value from the link's href attribute which contains the pathname of the image
assign caption the value from the link's title attribute which contains a description
of the image
change the src attribute value of the large display image to the value in var largePath
change the text of the paragraph element that we are using as a caption for the large image
to the value in var caption
return false to prevent the link from performing it's default function, which would
be opening the link.
```

A big lesson here is to return false when you want to prevent the link from performing it's default action after executing the function.

Good code, I think.

RESIZABLE

This part was weird. Maybe textarea elements weren't resizable when this movie was made.

I tried to fix it a little bit to compensate.

```
$(function() {
	var maxw = $("#commentsSection").width();
	var minw = maxw;
	var minh = $("#comments").height();
	$("#commentsSection").resizable({maxWidth: maxw});
	$("#comments").resizable( {maxWidth: maxw-10, minHeight: minh, minWidth: minw} );
});
```

There were no references to $("#comments") in the code presented in the movie, just to $("#commentsSection").

The code makes both the container div and the textarea inside it resizable with constraints. Some of the constraints were read from the commentsSection div element's underlying css.