



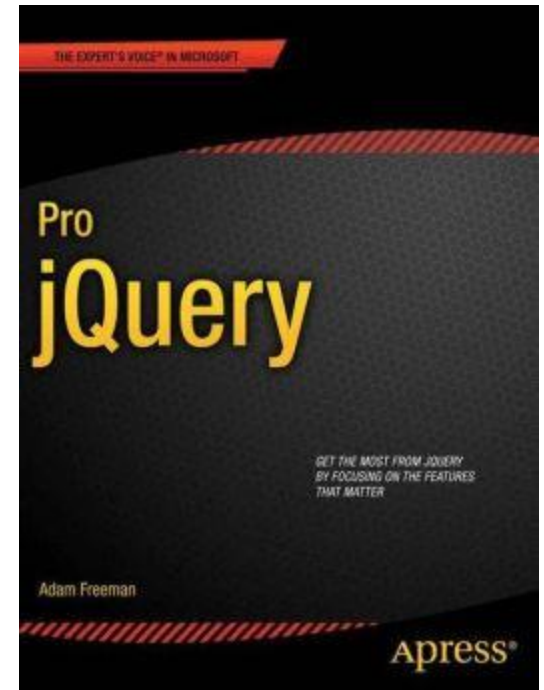
**An introduction to the JavaScript language through the original innovative language making JavaScript better**

# Topics

- Covered
  - How to set up and use jQuery
  - How to manage basic web design needs
- Not covered
  - How to program
  - How to use JavaScript
  - How to build a web app
  - jQuery UI
  - jQuery Mobile

# Course resources

- Book
  - Pro jQuery 2.0  
Adam Freeman  
Apress  
February 24, 2012
- Files
  - <http://github.com/doughoff/wd-505>





# jQuery intro

# DOM



- Document Object Model
- The html collection of nodes represented by an XML tree in memory of the browser created by parsing and fixing the html sent by the server.
- Any change to the DOM immediately fires a redraw to the browser screen.

# What is jQuery?

- a JavaScript library created in 2006
  - standards compliant
  - light-weight (92k)
  - cross-browser compatible
  - adopted by market leaders
- UI widgets became jQueryUI
- Mobile solution became jQuery Mobile
  - not an additional jQuery library, a new scripting language

# Why use jQuery?

- Easier to read/write than JavaScript
  - Compatibility, DOM effects, AJAX
  - Uses simple CSS selectors (Sizzle)
- Manages cross-browser incompatibilities
  - Becoming less important
- Extensible by plugins
- Industry adoption – Oracle, Google, Microsoft, IBM...

# Simpler syntax for DOM

## jQuery

## JavaScript

```
$(function( ) { // code...});
```

```
document.addEventListener('DOMContentLoaded', function() { // code... });
```

```
var divs = $('div');
```

```
var divs = document.querySelectorAll('div');
```

```
var aDiv = $('<div/>');
```

```
var aDiv = document.createElement('div');
```





# jQuery source code

- Current versions (7/2017) (min: 85k)
  - Core: **3.2.1** Slim (no Ajax and effects) (min: 68k)
  - UI: **1.12.1**
- Sources
  - local copy
  - **CDNs**
    - jQuery, Google
    - <http://cdnjs.com/> (for open source libraries)

# jQuery Migrate

- Use in development and testing
  - Restores deprecated features and behaviors
  - Two versions
    - Migrating older code since 1.9
    - Migrating code from 1.9 to 3.0
- Displays warning messages
  - Only the non-minified version.
- `<script  
src="https://cdnjs.cloudflare.com/ajax/lib  
s/jquery-migrate/3.0.0/jquery-  
migrate.js"></script>`

# Downloading jQuery

- Local

- `<script src='/script/jquery-1.11.1.min.js'></script>`
- `<script src='/script/jquery-ui-1.11.1.custom.min.js'></script>`

- Google CDN

- `<script  
src='http://ajax.googleapis.com  
/ajax/libs/jquery/3.1.0/jquery.  
min.js'></script>`

- cdnjs.com

- Search for jquery and click on link. Paste in file.

/jquery.min.js

Copy

Copy Url  
Copy Script Tag

# Versions

- Always use latest version?
  - Not best practice for production
    - <http://code.jquery.com/jquery-latest.js>
    - <http://ajax.googleapis.com/ajax/libs/jquery/1/jquery.min.js>
- Use 1.9?
  - If you need IE 6, 7 & 8 support stay with 1.12.\*
  - 2+ versions are about 20% smaller by dropping IE's old version compatibility

# Versions

- Non-minified
  - for study or debugging
- minified (.min.js)
  - no need to read the code
  - production
  - faster to download
    - for better user experience
    - no line breaks, single spaces, short variable names
- Slim – no ajax and effects

# Using plugins

- Always look for a plugin
  - will likely fix any browser problems
  - will add features
  - will look state-of-the-art
  - will speed up development
  - makes code more readable
- Look for the most recent plugin in a Google search
- Look at most recent overviews of plugins



# Writing jQuery

# On the web page

- jQuery loaded first
- Then JavaScript/jQuery code is written in a `<script>` tag below it
  - `<script src='/script/jquery-#.#.#.min.js'></script>`
  - `<script>`
    - `//this is where code goes (not DOM related)`
  - `</script>`
  - DOM (HTML)
  - `<script>` this is DOM related code `</script>`



# On the web page

- Sequence of events
  - browser requests URL and receives html file
  - browser loads html
    - Supporting files are requested and loaded async when read
    - JavaScript at the beginning is executed.
    - HTML5 provides a defer attribute to delay loading scripts
    - jQuery can defer script execution with a 'document ready' function.
    - DOM is loaded and rendered
    - document.load event is fired (blue? line in Network tab)
    - JavaScript at end is executed and can operate on DOM
  - changes to DOM are rendered immediately

# On the web page

- Bottom of page
  - Traditional JavaScript
- Top of page
  - Put code in a jQuery `document.load` (ready) function that delays execution until the page loads
  - Use the HTML5 **defer** attribute with a `src` attribute
    - Opera does not support
    - Buggy in <IE10 – check <http://caniuse.com>

# jQuery and JavaScript

- JavaScript code can be used at any point in jQuery code.
  - String and Math functions are popular to use.

# jQuery function

- jQuery( )
  - more common is the alias \$( )
- Four overloads
  - Depending on what is put in the parentheses, the function will do different things.

# jQuery namespace

- `jQuery.someFunction()`
  - Or `$.someFunction()`
- These are jQuery namespaced functions
- Many utility functions
  - see also Underscore and Lo-Dash (merging)
- Protects from name collisions

# jQuery function – select from DOM

- `$(‘a CSS selector’)`
  - returns a jQuery object (array-like) of matched elements
- `$(‘a CSS selector’, context element)`
  - Looks in a portion of the page
- Examples
  - `$(‘div’)`
  - `$(‘div’, ‘#rightColumn’)`

# jQuery function – JS to jQuery

- `$(element(s) or object(s))`
  - Returns a jQuery object (array-like element(s) or object(s))
  - Used when you use JavaScript usually to retrieve something from the page
- Examples
  - `$(this)`
  - `$(document)`

# jQuery function – new HTML

- `$('html we write', object with attributes)`
  - Returns a jQuery object (array) of one new element
  - Used when adding html the DOM
- Examples
  - `$('<div>This is my div</div>')`
  - `$('<div/>')` or `$('<div></div>')`



# jQuery function - new HTML

- \$('<img/>', **map**)
- **map** = a JavaScript object = JSON format
- {**src**: 'http://www.camp-4-paws.com/wp-content/uploads/2013/12/slider-dog3.png',
- **alt**: 'a dog',
- ...
- }

# jQuery function – defer execution

- `$(function( ){ ...deferred code... } );`
  - executes the callback function of deferred code when document.load event is fired
  - Called the document ready or **ready function**
  - Used to defer execution of code placed before the DOM to the end of the page
  - Use for DOM manipulation code when placed before the DOM it works on.

# jQuery function – defer execution

- The document.load event callback
- The short (anonymous) version (used in course)
  - **`$( function( ) {`**
  - `alert('Testing...testing...');`
  - `});`
- The long version
  - **`$(document).ready(function( ) {`**
  - `alert('Testing...testing...');`
  - `});`

# jQuery ready function template

- This will be your template for DOM manipulation jQuery code at the top of the file
  - `<script>`
  - `$( function( ) { // ready function`  
`code here`  
`}); // end ready function`
  - `</script>`
- Keep the comments to ensure you don't delete the line

# Chaining statements

- Reuse selector and create terser code
  - `$(...)`
    - `.functionName1 ( arg, arg, ... )`
    - `.functionName2 ( arg, arg, ... )`
  - also called fluent style

# JavaScript comments

- 2 styles
  - `//` ignored until the end of line
  - `/*`  
ignored  
until  
`*/`

# Exercise



- Follow instructions in Exercises handout for
  - Create an html template
  - (optional) Add a script template without anonymous functions



The jQuery function used to select nodes/elements/attributes

## Selecting



# Basic XML terms

- XML **document** – the entire text
- **element** – anything between and including the open tag and matching closing tag
  - `<a>ajdfkjhad<b>hklha lj alskfjh ld</b><c>hjad jf hajldf</c></a>`
  - `<a></a>`
  - `<a />`
- **tag** – just the open or closing part, or empty element
  - `<a>ajdfkjhad<b>hklha lj alskfjh ld<b/><c>hjad jf hajldf</c></a>`

# Basic XML terms

- body – what's in between the open and closing tags
  - `<a>ajdfkjhad<b>hklha lj alskfjh Id</b><c>hjad jf hajldf</c></a>`
  - `<a>ajdfkjhad<b>hklha lj alskfjh Id</b><c>hjad jf hajldf</c></a>`
- attribute – meta data for the element
  - `<x id='the unknown'>Dean Jagger as Dr. Adam Royston</x>`
- XHTML – HTML - XML compliant
  - DTD – document that gives rules for XML language to validate data
  - DOCTYPE – instruction for browser to recognize data

# HTML tree traversing

- **child**
  - the immediate nested level of an element
- **parent**
  - the immediate level of an element that an element is contained by
- **descendant**
  - any nested level of an element
- **ancestor**
  - any nested level that an element is in

# HTML tree traversing

- parent of **b1** is **a**
  - ancestor of **c** is **a** is but is not a parent
- child of **a** is **b1**
  - descendant of **a** is **c** but is not a child
- sibling of **b1** is **b2**
  - `<a>`
    - aaaaaaaaaaaaaaaaaa
    - `<b1>bbbbbbbbbbbbbbbbbbbb</b1>`
    - `<b2>BBBBBBBBBB<c>B</c>BBBBBBBBBB</b2>`
    - aaaaaaaaaaaaaa
  - `</a>`

# DOM



- DOM – document object model
    - JavaScript talks to the html text as an object called the **document**
    - subsets of the DOM are called fragments
- ```
▼ document: HTMLDocument
  URL: "http://jqueryui.com/"
  ► activeElement: HTMLBodyElement
  alinkColor: ""
  ► all: HTMLAllCollection[156]
  ► anchors: HTMLCollection[0]
  ▼ applets: HTMLCollection[0]
    length: 0
  ► __proto__: HTMLCollection
  attributes: null
  baseURI: "http://jqueryui.com/"
  bgColor: ""
```

# DOM manipulation

- DOM – Document Object Model
  - the in-memory OO version of the HTML page
- Using JavaScript to change the DOM which then updates the rendered view of the browser
- The user sees a very responsive interactive page.

# HTML

- **id** attribute
  - the unique name given to one element
  - CSS selects this with a # (pound sign) prefix
- **class** attribute
  - multiple elements can have the same class
  - CSS selects this with the . (period) prefix

# Selecting by CSS

- Use CSS3 selectors
  - element - `$('table')`
  - class - `$('table.data')`
  - id (best) - `$('table#celebs')`
  - descendant - `$('table tr td')`
  - union - `$('p,div,h1,input')`
- Select as narrowly as you can
- Use matched quotes



# Demo

- On a page that has loaded jQuery, run in the console:
  - `$('p')`
  - `$('p').length` – how many paragraphs are there?
  - `$('div').`
  - Etc.

# Selecting by CSS

- CSS3 attribute selectors
  - any value- `$( '[title]' )` or `$( 'a[title]' )`
  - matches - `$( '[title = "value"]' )`
  - does not match - `$( '[title != 'value']' )`
  - starts with - `$( '[title |= 'value']' )`
  - ends with - `$( '[title $= 'value']' )`
  - contains all - `$( '[title *= 'value']' )`
  - contains word - `$( '[title ~= 'value']' )`
  - union - `$( '[title = 'value'] [alt = 'value']' )`

# Selecting by CSS

- Pseudo classes
  - :even, :odd, tr:eq(#)
  - :animated
  - :first, :last
  - :focus
  - :gt(#), lt(#)
  - :header
  - :not()

# Selecting by CSS

- text nodes (body of element) are children too!
- children
  - :first-child, :last-child, nth-child(#) - one-based, only-child
- content
  - :contains('text'), :empty, :has( **selector** ), :parent
- hierarchy
  - child - ('parent > child')
  - next adjacent - ('prev + next')
  - next sibling - ('prev ~ siblings')

# Selecting by CSS

- Forms
  - :focus
  - :checked, :disabled, :enabled, :selected
- Visibility
  - :hidden, :visible
- Deprecated
  - :button, :file, :image, :text
  - :input, :password, :checkbox, :radio
  - :reset, :submit

# Selecting with jQuery functions

- `.is( selector )`
  - `if ($(this).is(':hidden')) ...`
- `.not ( selector )`
  - `$('p').not('#selected')`
- `.eq(#), .first( ), .last( ), has( )`
- `.slice( ) - $object`  
→ `$objectSubsetByIndicesRange`

# Subselecting

- find by selector - includes only children
  - `$('li').find('a')`
  - `$('li').find('.highlight')`
    - `$('li .highlight')` or `$('.highlight', 'li')` is slower
- find by jQuery object
  - `$('li').find($link)`
- `$( selector ).find( tagName )` // faster
  - = `$( tagName, selector )`
  - = `$( 'selector tagName' )` // my choice

# Selecting by tree traversing

- `.closest()` – find the best parental match or itself
- `.children()`, `.siblings()`
- `.parent()`, `.parents()`, `.parentsUntil()`, `.offsetParent()`
- `.prev()`, `.prevAll()`, `.prevUntil()`,
- `.next()`, `.nextAll()`, `.nextUntil()`



# Selector optimization

- Fast to slow
  - ID
  - element name
  - CSS class
  - attribute value
  - CSS pseudo selector
- Moral
  - always use an ID for single selection

| Testing in Chrome 16.0.912.63 32-bit on Windows Server 2008 R2 / 7 64-bit |                                  |                                 |
|---------------------------------------------------------------------------|----------------------------------|---------------------------------|
|                                                                           | Test                             | Ops/sec                         |
| jQuery ID                                                                 | <code>\$("#foo");</code>         | 436,062<br>±0.00%<br>fastest    |
| jQuery Tag                                                                | <code>\$("blockquote");</code>   | 141,837<br>±7.14%<br>67% slower |
| jQuery Class                                                              | <code>\$(".bar");</code>         | 161,948<br>±1.06%<br>63% slower |
| jQuery attribute                                                          | <code>\$("[name='baz']");</code> | 31,788<br>±1.19%<br>93% slower  |
| jQuery Pseudo-selector                                                    | <code>\$(":hidden");</code>      | 557<br>±0.71%<br>100% slower    |



# Selector optimization

- Don't use CSS descendants
  - `$('#id p');`
- Use chained function calls
  - `$('#id').find('p');`
  - can be more than 2x as fast as the first



# Exercise

- Follow instructions in Exercises handout for
  - Show an alert when any link is clicked
  - Show an alert when a unique link is clicked
  - Show an alert when a link in a special div is clicked

# Debugging workflow

1. Check console errors - F12
2. Check unchecked comment syntax
  1. CSS - `/* */`
  2. HTML - `<!-- -->`
  3. JavaScript - `/* */`, `//`
3. Load order errors
  1. jQuery, jQueryUI, DOM, DOM manip.
  2. jQuery, Migrate
  3. jQuery, plug-in
  4. normalize, theme, site CSS

# Debugging workflow

4. DOM selection errors
  1. # for ids, period for classes, space for descendant
5. Logic errors - insert `console.log( )`
  1. to assert every known value as it is known
  2. to verify conditional blocks are running
6. Debugger
  1. debugger;
  2. Set a breakpoint

# Exercise



- Follow instructions in Exercises handout for
  - Show output from a debug script and in the DOM



# Events

# Events

- The triggers that run code (event handlers)
- Pre-defined and managed by the browser
- Custom



# Old event binding

- HTML binds the event handlers with attributes
  - `<button`  
**`onclick='document.getElementById('field2').value`**  
**`=`**  
**`document.getElementById('field1').value'>`**
- Poor coding practice
  - sprinkles code throughout document
  - better to combine code in one place

# Event binding

- Add/register/**bind** your code/functions to the element on the browser's events
  - when the browser 'hears' a **click** event on the button it fires/does my **function**
- `$('#aButton').on('click', function( ) {  
    // custom button action  
});`

# Event binding

- Click handler for **all** links
  - `$('#a').on ('click', function() {`
  - `alert('Nooooooooo. Don't leave!!!!');`
  - `});`
- shortcut
  - `$('#a').click( function() {`
  - `alert('Nooooooooo. Don't leave!!!!');`
  - `});`

## on( ), off( ) - 1.7

- **on(** event [, selector] [, data] , handler );
- **off(** [events] [, selector] [, handler] );
  - **events** – event names e.g. 'click', 'click hover resize'
  - **handler** – the function to fire on the event(s)
- New methods unify event handling
  - no need for bind( ), delegate( ), or live( )
  - stop using bind, live, and delegate **now!**

# Event handlers

shortcuts for on( 'event', callback function )



- .change()
- .click(), .dblclick()
- .error()
- .blur(), .focus(), .focusin(), .focusout()
- .hover()
- .keydown(), .keypress(), .keyup()
- .load()
- .mouseenter(), .mouseleave()
- .mousemove(), .mouseout(), .mouseover(), .mousedown(), .mouseup()
- .ready()
- .resize()
- .scroll()
- .select()
- .submit()
- .toggle() – different than hide/show
- .unload()

# Event object

- jQuery always passes the event object as the first argument to the handler.
  - `$('#html')`  
    `.click(function(eventObject){`  
        `console.log(eventObject);`  
        `// inspect the object in the browser`  
    `});`

# Event handlers - deprecated

- **Deprecated** in 1.8 (check with Migrate)
- **.toggle(function1( ) { }, function2( ) { }, ...)**
  - 1<sup>st</sup> click, 2<sup>nd</sup> click, 3<sup>rd</sup> click...
- same function name is still used for hide and show animation
  - .toggle(5000) – show/fade element over 5 seconds
  - .toggle(true) – show element

# Event handlers - map

- If a map property matches the event registration method, it's an event handler.
- `$('<img/>', {  
 src: 'https://www.petfinder.com/wp-content/uploads/2012/11/142025007-petside-dog-dry-nose-sick-632x475.jpg',  
 alt: 'dog',  
 class: 'translucentBox',  
 click: function( ) { $(this).css('opacity', '.50'); } }));`







# Demo

- Add 8 dog pictures 250px wide. Click each one to test.
- ```
$( '<img/>', {  
  src: 'find one on Google',  
  alt: 'dog',  
  title: 'good dog',  
  style: 'float: right; width: 250px',  
  click: function(e) { $(this).css('opacity', '.50'); }  
})  
  .appendTo($('body'))
```

# Event propagation

- Event objects generated by an event will 'bubble up' to the parent element
  - event objects are not consumed
- A parent div can handle any child element's event object.
  - ask the event object for what element got the click
  - `$('#parent').click(function(e) {`
    - `console.log(e.target.id);`
  - `});`

# Event propagation

- HTML

- `<div id='parent'>`
  - `<div id='child1'> Click child 1! </div>`
  - `<div id='child2'> Click child 2! </div>`
- `</div>`

- jQuery

- `$('#div').click(function(e) {`
  - `alert('Hello from ' + $(this).attr('id'));`
  - `e.stopPropagation( ); // will not send event to parent`
- `});`

# Event delegation optimization

- A normal 1.7 bind -
  - `$('table').find('td').on('click',function( ) {  
$(this).toggleClass('active'); });`
- If table has 100 cells, we have 100 event listeners!
- Better is one event listener – for any td of the table
  - `$('table').on('click','td',function( ) {  
$(this).toggleClass('active'); });`

no delegation	<pre><code>\$('table').find('td').on('click', function() {     \$(this).toggleClass('active'); });</code></pre>	1,421 ±9.53% 93% slower
with delegation	<pre><code>\$('table').on('click', 'td', function() {     \$(this).toggleClass('active'); });</code></pre>	19,438 ±8.81% fastest

# Prevent default event actions

- Clicking on a link creates a click event.
- Stop the click event with
  - `$('#a').click(function(e) {`
    - `e.preventDefault(); // will send event to parent`
    - `// do our event action stuff`
    - `url = $(this).attr('href');`
    - `window.location = url;`
  - `});`

# return false;

- Put at end of function
- does 3 things:
  - event.preventDefault()
    - all you need to cancel behavior
  - event.stopPropagation()
    - stop bubbling
  - stopImmediatePropagation( )
    - stops any further event execution, even to other handlers bound on the same object

```
/events/*.html  
/migrate/*.html
```



# Exercises

- Show an alert for chained selectors
- On click, add a class to span elements
- On hover, change a class
- Check out migrate examples if you have time.



# DOM manipulation



# Attributes

- **.addClass( )** – creates a class attribute
  - .removeClass( )
  - .toggleClass( ) – adds and removes
- **.attr( )** – creates/reads any attribute
- **.css( )** – creates a style attribute, used for testing

# .addClass( )

- addClass ( **className ...** )
  - \$('div').addClass('className');
  - \$('div').addClass('className1 className2 className3');
- Table with alternating row colors
  - **CSS**
    - .contrastingGrays { background-color: #dddddd; color: #666666; }
  - **jQuery:**
    - \$('#celebs tr:even').addClass('contrastingGrays');

# **.removeClass( )**

- removeClass( **className** )
  - \$('div').removeClass('className');
- check for class with .hasClass( **className** )

# .toggleClass( )

- same as
  - `$el.hasClass(className)`  
  ? `$el.removeClass( className )`  
  : `$el.addClass ( className )`
- CSS
  - `.bgLemonchiffon { background-color: #FFFACD; }`
- jQuery
  - `$('#celebs tr').hover( function() {  
    $(this).toggleClass('bgLemonchiffon');  
});`

# **.text( )**

- **Get** text node value
  - `$('h2:first').text( )`
  - Returns string of concatenated body results
  - Strips all tags
- **Set** the text node
  - `$('p').text('set text to this');`

# .text( ) vs. .html( )

- Two methods to **set** body of element
  - \$(**selection**).html( **text** ) – text will be rendered as if source
    - Use to remove current element body and add marked up text
  - \$(**selection**).text( **text** ) – text will be rendered as if in a <pre> (as-is)
    - Use to remove current body and add text
- html( ) is best overall

# .attr( )

- get or set any attribute of an element
  - HTML
    - `<p>`  
Once there was a `<em title='huge, gigantic'>large</em>`  
dinosaur...  
`</p>`  
The title of the emphasis is:`<div id='titleText'></div>`
  - jQuery
    - `var title = $('em').attr('title');`  
`$('#titleText').html(title);`

# .attr( )

- add
  - \$('#greatphoto').attr('title', 'Photo by Kelly Clark');
- change
  - \$('#greatphoto').attr('alt', 'Beijing Brush Seller');
- multiple attributes using map
  - \$('#greatphoto').attr(  
    { alt: 'Beijing Brush Seller',  
      title: 'photo by Kelly Clark' }  
);



# id



- The attribute id can be retrieved in JavaScript by the field id
  - `<section id='topOfPage'>`
  - `$('section').attr('id')`
  - `$('section')[0].id`
  - `document.querySelector('section').id`

## **.css( property ) - get**

- `<table id='celebs'><tbody>  
    <tr style='font-size:12px'>`
- `var fontSize = $('#celebs tr:first')  
    .css('font-size');`

## **.css( property, value ) - set**

- `$('#celebs tr:even')`  
`.css('background-color', '#dddddd');`
- Second argument causes an assignment
  - `.css( { } )` – use a map for multiple key-values

# Create an element

- Create and select a fragment with `$( )`
  - `var $p = $('`
- Add a class to the fragment
  - `$('#<p>A new paragraph!</p>').addClass('new');`
  - or
  - `$p.addClass('new');`
  - or
  - `var $p = $('#<p/>').html('A new paragraph').addClass('new');`

# Add/move element – as child

- inside of target ← content
  - `< div id='disclaimer' >`
  - `$('#disclaimer ').prepend('... content ...');`
    - text and elements inside div
  - `$('#disclaimer ').append('... content ...')`
    - `</ div>`
- content → inside of target
  - `< div id='disclaimer' >`
  - `$('#... content ...').prependTo('#disclaimer');`
    - text and elements inside div
  - `$('#... content ...').appendTo('#disclaimer');`
    - `</ div>`

# Add/move element – as sibling

- no need to remove before inserting if relocating
- outside of target ← content
  - `$(' #disclaimer ').before('... content ...');`
  - `< div id='disclaimer' />`
  - `$(' #disclaimer ').after('... content ...');`
- content → outside of target
  - `$('... content ...').insertBefore('#disclaimer');`
  - `< div id='disclaimer' />`
  - `$('... content ...').insertAfter('#disclaimer');`

# Add/move

- Child or sibling
- jQuery object or html text

# clone()

- `var $p = $('<p/>');`
  - `var $template = $('template#a');`
- `$p.appendTo('body')` - inserts \$p
- `$p.after('h1')` - moves \$p after inserted
- `$p.clone().after('h1')` - adds a copy, no events
- `$p.clone(true).after('h1')` - adds a copy with events - best
- `$p.clone(true).html('new headline').after('h1')`



## **.wrap( ), unwrap( )**

- `wrapAll('<p id='wrap'>')` - wraps each selected element
- `wrapInner('<p id='wrap'>')` - wraps contents of each selected element
- `unwrap( )` removes parent

# .remove()

- HTML

- `<p id='no-script'> We recommend that you have JavaScript enabled! </p>`

- jQuery

- `$('#no-script').remove();`
- `$('#celebs tr').remove(':contains('Singer')');`
  - where the row contains the text 'Singer'

## **.empty( ), .detach( )**

- **empty( )** removes children elements and text nodes but leaves element
- **detach( )** method works just like `remove( )` but does not remove event handlers and custom data
  - useful for a temporary removal

# Demo

- jQuery.com
- Add menu items to the page
  - `var $menuLink = $('<a/>').attr('id','link').attr('href','http://google.com').html('Google')`
  - `var $menuitem = $('<li/>').addClass('menu-item')`
  - `$menuitem.append($menuLink)`
  - `var $menu = $('ul#menu-top')`
  - `$menu.append($menuitem) // prepend`
  - `var $menuitem2 = $('#menu-top>li:nth-child(2)')`
  - `$menuitem2.after($menuitem) // before`
  - `var $apiMenu = $menuitem2.clone(true)`
  - `$menuitem2.after($apiMenu)`

# DOM manipulation optimization

- DOM change = repaint + reflow
- Append an li to the DOM in a loop (bad)
  - `var arr = [reallyLongArrayOfImageURLs];`
  - `$.each(arr, function(count, item) {`  
    `var newImg = '<li><img src=' + item + ' ></li>';`  
    `$('#imgList').append(newImg);`  
    `// render again`  
});

# DOM manipulation optimization

- This code appends once outside the loop (good)
  - `var arr = [reallyLongArrayOfImageURLs], tmp = '';`
  - `$.each(arr, function(count, item) {`  
    `tmp += '<li><img src=' + item + ' ></li>';`  
    `});`
  - `$('#imgList').append(tmp); // render once`

```
/dom/*.html  
/migrate/*.html
```



# Exercises

- Append text to an element
- Create a striped table with hover highlighting
- Run and review the migrate examples
  - Read the console messages
  - Also, run without the migrate code



# JavaScript syntax



# Operators

- Is equal to `==` and `===`
  - JavaScript has two equality operators
  - Best practice: use `===` always

# Operators

- the modulus operator
  - 5 mod 2 is written `5 % 2`
  - it means subtract as many 2s as possible and return the remainder
  - helpful for paging and loops
- compound operators
  - `var x = 1;`
  - `x += 5` means add 5 to the value of x
  - `x = x + 5;`
  - `x -= 5` means subtract 5 from the value of x
  - `x = x - 5;`

# JSON format

- How to create an object
- {  
    key: value,  
    key: value,  
    key: value
- }
- also called a map
- `var properties = {'alt':'dog', style:'width:250px'};`



# Animation effects

# .hide( )

- Create a button and text to hide
  - `<input id='hideButton' type='button' value='hide text' />`
  - `<div id='disclaimer'>...</div>`
- `$('#hideButton').on('click',function( ) {  
    $('#disclaimer').hide( );  
});`
- Or hide the selection itself with **this**
  - `$('#hideButton').on('click',function( ) {  
    $( this ).hide( );  
});`

# .show( )

- to show hidden elements
  - `<input type='button' id='showButton' value='show' />`
- `$('#showButton')`
  - `.on('click',function( ) {`
    - `$('#disclaimer').show( );`
- `});`

# **.hide( ) / .show( ) options**

- `show( [duration][, easing][, callback] )`
  - Duration
    - Default – 400ms
    - 'slow' – 600ms
    - 'fast' – 200ms
    - # of ms – 1000, 2000, 8000, etc.
  - `easing = 'linear', 'swing'`
  - `callback` – function to run when done (synchronous call)

# .toggle( )

- animated insert, animated remove
- hide( ) & show( )
  - Old school – use **if** to see if it's visible, then modify
- HTML
  - `<input type='button' id=' toggleButton' value='show' />`
- jQuery
  - `$('#toggleButton ').on('click', function( ) { $('#disclaimer').toggle( ); } );`



# **.toggle( )**

- `hide( ) = toggle(false)`
- `show( ) = toggle(true)`

# Fade methods

- `fadeIn([duration][,easing][, callback])`
- `fadeOut([duration][,easing][, callback])`
- `fadeTo(duration, opacity [,easing][, callback])`
  - opacity - between 0 and 1
- `fadeTo(0, opacity) = css('opacity', opacity).`
- fade with easing, collapse is sudden

# Hide & fade in with prompt

- Hide any element with a spoiler class
  - `$('.spoiler').hide( );`
- Put a teaser element before any spoiler element
  - `$('<span class='revealer'>Tell me!</span>').insertBefore('.spoiler');`

# Hide & fade in with prompt

- Bind any teaser on click to hide, and then fade the next sibling in (spoiler)
  - `$('.revealer').click(function() {  
 $(this).hide();  
 $(this).next().fadeIn();  
});`

# Arguments in map

- `.fadeIn({`
  - **duration**: 'fast',
  - **easing**: 'swing',
  - **complete**: function( ){  
    \$(this).text('Hello World');  
    }
- `});`

## **.fadeToggle( )** 1.4.4

- abrupt insert, fade, abrupt remove
- .fadeToggle( [duration] [, easing] [, callback] )
- ```
$('#button:last').click(function ( ) {  
    $('#p:last').fadeToggle('fast', function ( ) {  
        $('#log').append('<div>finished</div>');  
    });  
});
```

# Slide effects

- `slideDown( [speed][,callback] );`
- `slideUp( [speed][,callback] );`
- `slideToggle( [speed][,callback] );`
  - `$(selection).slideToggle('slow', function( ) { alert('Done!'); });`
  - `$(selection).slideToggle( 'slow', doAfterToggle);`

# Chaining effects

- Unchained and concurrent (asynch)
  - `$('#p:first').show( 5000 );`
  - `$('#p:first').slideDown('slow');`
  - `$('#p:first').fadeOut( );`
- Chained and queued (synch)
  - `$('#p:first')`
    - `.show( 5000 )`
    - `.slideDown('slow')`
    - `.fadeOut( );`



# Chaining effects

- `$('#blinker')`  
    `.fadeIn(100).fadeOut(100)`  
    `.fadeIn(100).fadeOut(100) .fadeIn( );`

# delay( )

- delay(duration [, queueName] )
- Adds a pause
  - `$('#p:first').hide().slideDown('slow').delay(2000).fadeOut( );`
- queueName defaults to fx, the standard effects queue

# True cross-fading

- Use CSS' z-index or the ordered stack of elements
  - requires same size images

# Plugins - effects

- **InnerFade -**

- <http://medienfreunde.com/lab/innerfade/>

- allows multiple images or blocks to cycle cross-fade or slide

- **Zoom & WheelZoom**

- <http://www.jacklmoore.com>

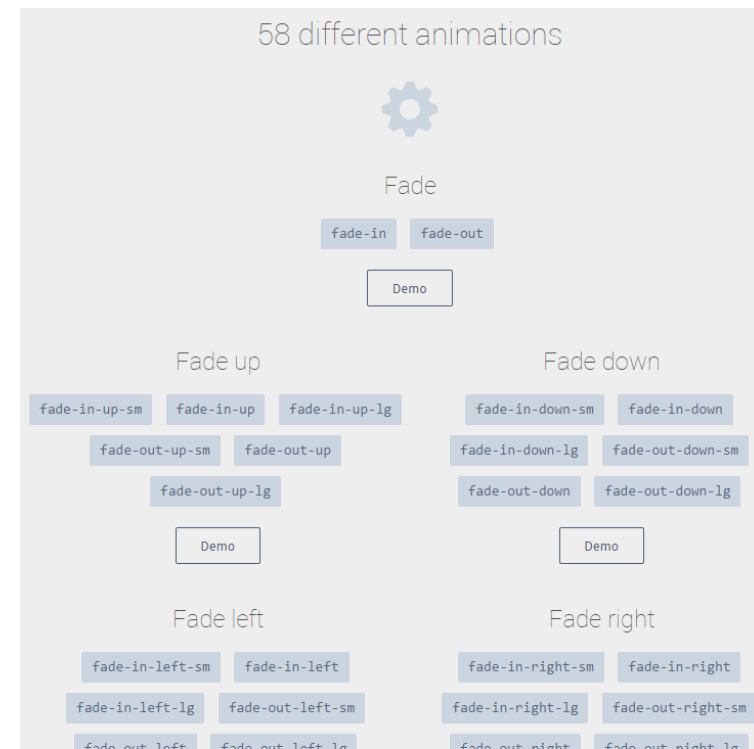
- hover / mousedown / mouse wheel to zoom image

- **Transit - use CSS effects instead (smoother)**

- <http://ricostacruz.com/jquery.transit>

# Page transitions - plugins

- Animsition
  - <http://git.blivesta.com/animsition/>





# Exercises

- Toggle text visibility
- InnerFade with images



# More Animation

# Easing

- Acceleration and deceleration like under gravity or using rubber band stretching.
- Typically follows duration as an option.
- Default = 'swing'
  - 'linear' is boring.
- Show difference
  - `$('#p:first').toggle(function() { $(this).animate({height :'+=150px'}, 1000, 'linear'); }, function() { $(this).animate({height :'-=150px'}, 1000, 'swing'); });`



# .animate( )

- gradual transition to CSS properties defined in an object literal

- Object literal

```
.animate( properties, options )  
properties A map of CSS properties that the animation will move toward.
```

- { **key: value** , **key: value**, ... }
- { opacity: 0.25, left: '+=50', height: 'toggle' }
- { padding: '20px', 'border-bottom': '3px solid #8f8f8f', borderRight: '3px solid #bfbfbf' }

# animate( )

- Two syntaxes
  - `$(this).animate({ height: '-=350px' }, 3000, 'swing');`
  - `$(this).animate({ height: '-=350px' },
    - { duration: 3000, easing: 'swing' }`
  - `);`
- Options allows for more keys such as queue
  - **queue:** false
    - doesn't wait for previous animation effect to complete

# animate( )

- `$(selection).animate(  
map,  
[duration], [easing]);`
- `$('#p').animate(  
{ padding: '20px',  
borderBottom: '3px solid #8f8f8f',  
borderRight: '3px solid #bfbfbf'  
},  
2000);`

# Color animation

- due to color complexity, requires jQuery UI
  - `$('.block').toggle(`
  - `function( ) {`  
    `$(this).animate({ backgroundColor: 'black' },`  
    `1000); },`
  - `function() {`  
    `$(this).animate({ backgroundColor: '#68BFEF' }`  
    `, 500); });`

# stop( )

- **.stop([clearQueue][,goToEnd])**
  - clearQueue when true stops any currently running animations before moving on
  - goToEnd when true will cause animation to immediately complete calling callback
- typical use is menu dropdowns

# effect() - jQuery UI

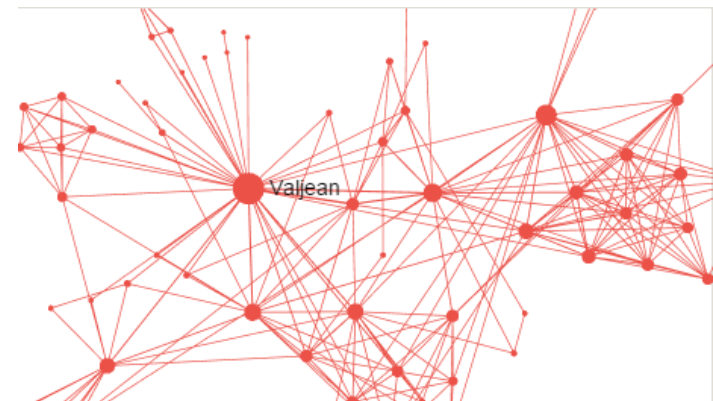
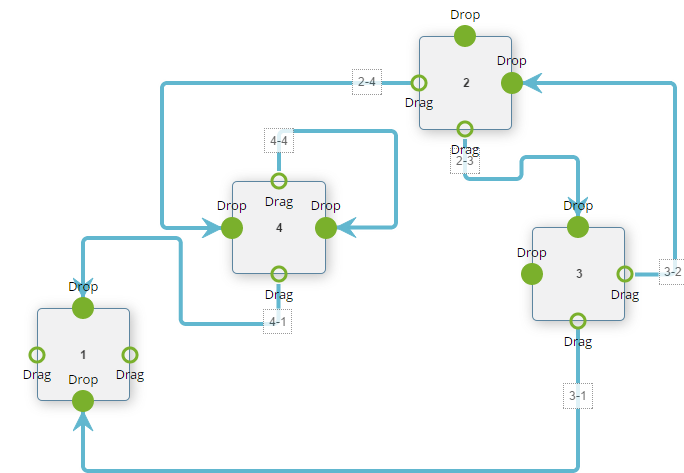
- Used with Show/Hide/Toggle
  - [Blind](#), [Clip](#), [Drop](#), [Explode](#), [Fade](#), [Fold](#), [Puff](#), [Slide](#), [Scale](#)
- Standalone effects
  - [Bounce](#), [Highlight](#), [Pulsate](#), [Shake](#), [Size](#), [Transfer](#)
- `$('.p:first')`
  - `.effect('shake', {times:3}, 300)`
  - `.effect('highlight', { }, 3000)`
  - `.hide('explode', { }, 1000);`

# Plugins - animation

- Keyframes
  - <http://farukat.es/journal/2011/02/514-new-creation-jquery-runloop-plugin>
  - **Sequence** - <http://www.sequencejs.com/>
- Smart3D
  - <http://4coder.info/proekty/jquery-plaginy/jquery-smart3d/>
- Transit - animate replacement
  - <http://ricostacruz.com/jquery.transit/>
- Adipoli - hover transitions
  - <http://jobyj.in/adipoli/>
- **Velocity.js** - animate( ) replacement
  - <http://julian.com/research/velocity/>

# Data display - plugins

- Elycharts - <http://elycharts.com>
- Flot – <http://flotcharts.org>
- jsPlumb
  - <https://jsplumbtoolkit.com/>
- Sigma.js - networks
  - <http://sigmajavascript.org/>





# Exercises



- Animate the addition of CSS on load
- Find and set CSS based on window size



# Scrolling and resizing

# Scrolling

- Detect user's scrolling with **scroll**( ) event
  - divs or windows
- CSS to simulate a scrolling div
  - `#news { height: 100px; width: 300px; overflow: scroll; }`
- jQuery event handler
  - `$('#news').on('scroll', function( e ) { ... });`
  - `$('#news').scroll(function( e ) { ... });`

# Floating navigation

- Movable menu CSS
  - position: relative
  - top: adjusted by jQuery based on how much is scrolled
    - `$(window).scroll(function( ) { $('#navigation').css('top', $(document).scrollTop( ... ) ); });`
- `.scrollTop( )` – vertical position of 1<sup>st</sup> scrollbar (how much is hidden in pixels)
- position: fixed will do same thing without easing

# Floating navigation

- Final version
  - `$(window).on('scroll', function( ) {`
  - `$('#navigation')`
  - `.stop()`
  - `.animate( { top: $(document).scrollTop( ) },`  
`'slow' ,`  
`'easeOutBack');`
  - `});`

# Scrolling the document

- Adding animation to `<a href='#>Go to top</a>`
  - ```
$( 'a[href=#] ' ).on( 'click', function( e ) {  
    e.preventDefault();  
    $( 'html' ).animate( { scrollTop: 0 }, 'slow' );  
}
```
- `'a[href=#]'` – attribute selector
  - select all links where the href attribute equals #
- **preventDefault()** or **return false;** at end of function for event handler to stop event's default behavior

# Demo

- `$(window).scroll(function(){`
  - `var newColor = 'hsl(' + $(this).scrollTop() % 360 + ', 50%, 50%)';`  
`console.log(newColor);`
  - `$('#body')`  
`.css('background-color', newColor);`
- `});`

# Plugins - scrolling

- **ScrollTo**

- <http://flesler.blogspot.com/2007/10/jqueryscrollto.html>

- **Other**

- <https://www.pixxelfactory.net/jInvertScroll/>
- **\*ScrollMagic**– supports parallax effects  
<http://janpaepke.github.io/ScrollMagic/>
- **jParallax** - <http://stephband.info/jparallax/>
- **(Almost) Static Background** -  
<http://johnpolacek.github.com/scrolldeck.js/decks/parallax>
- **Scrollpath** - <http://joelb.me/scrollpath/>



# Plugins - scrolling

- Mouse wheel support
  - left out of jQuery on purpose, add it back in if needed
  - <http://plugins.jquery.com/project/mousewheel>
- Infinite / Endless Scroll
  - <https://github.com/fredwu/jquery-endless-scroll/tree/2.0-rewrite>
  - Waypoints (show on bottom, detect when seen, dial controls) <http://imakewebthings.github.com/jquery-waypoints/>
  - <http://www.infinite-scroll.com/infinite-scroll-jquery-plugin/>

# Resizing

- Detect user's resizing with **resize**( ) event
  - for document views
- Demo code
  - ```
$(window).on('resize', function( ) {  
    alert('You resized the window!');  
});
```
- Dangerous
  - like scrolling, resizing fires events for every little change
  - used for monitoring

# Resizing

- Fluid or fixed width?
- Responsive design – adaptive layouts
  - <http://www.lifeonlars.com/design/web-design/responsive-web-design-adaptive-layouts-with-media-queries>
- Check current width
  - ```
if ($('#body').width( ) > 900) {  
    $('<link rel='stylesheet' href='wide.css' type='text/css'  
/>')      .appendTo('head');  
} else {  
    $('link[href=wide.css]').remove( );  
}
```

# Resizing – current state

- Browsers are confused about the visual viewport.
- Browsers agree the resize event should be fired when the dimensions of the visual viewport, the layout viewport, or the HTML element change.
- Slightly more than half of browsers fire a resize event when the HTML element's dimensions change.
  - Only Android up to 2.3 and BlackBerry fire the event in all tested cases.
  - Android 3.2 is worse in firing event than 2.3.
  - Does not work at all in Opera Mobile.

# Plugins

- Responsive design
  - Restive - <https://github.com/obihill/restive-plugin>
- Centering
  - Centr - <http://boeddo.com/centr>



# Exercises

- Shake and slide
- Floating navigation



# jQuery UI

# Check if loaded

- in the console type
  - `jQuery.ui`
  - `$.ui`
- undefined = no
- Object { ... } = yes



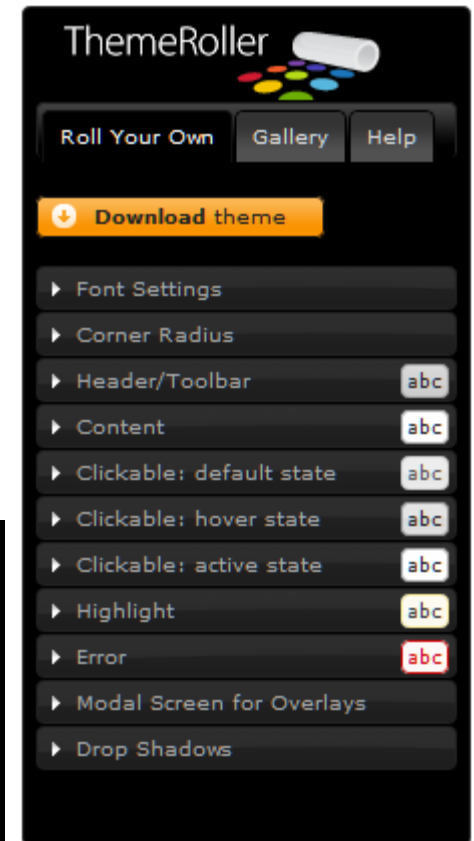
# ThemeRoller

- jQuery UI rewrites the all UI elements with classes using ThemeRoller CSS
- Choose Themes on the jQuery UI menu
  - Select a theme from the Gallery
  - Use the CSS



# Theme rolling

- <http://jqueryui.com/themeroller/>
- Gallery
  - Default – lightness theme



# Using themes

- Wrapper div: `class='ui-widget'`
  - class `ui-helper-reset` can be replaced with another reset
- Docs - <http://jqueryui.com/docs/Theming/API>
- Class categories
  - layout helpers
  - widget containers – header, content
  - interaction states
  - icons
  - corner radius helpers
  - overlay & shadow

# Exercise



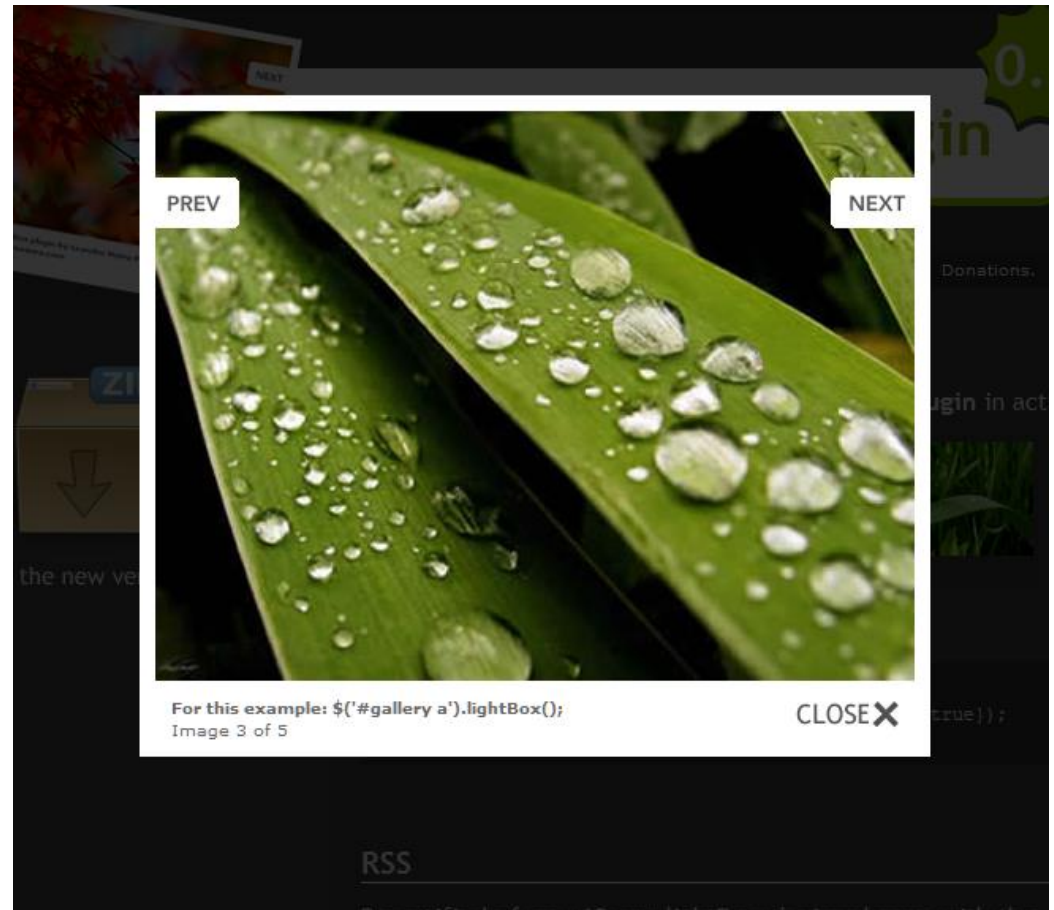
- Follow instructions in Exercises handout for
  - Create an html template for ThemeRoller



# Image plugins

# Lightboxes

- Click thumbnail image, show full-size image in modal dialog
- Linkable text in captions.
- Slideshows are automatic advancing.



# Carousels

- Lightbox with features
- Filmstrip of coming images at bottom



# Sliders

- Not the same as a slider control
- Large image, right & left navigation buttons, extra CSS
- Most popular plugin on home pages



*Chrome*



*Safari*



*Opera*





# Plugins - images

- **Flexslider** – slider / carousel
  - <http://flexslider.woothemes.com/>
- **Cycle** - slider
  - <http://malsup.com/jquery/cycle/>
  - effects, speed, timeout, random, pause on hover, easing, delay, syncing, event handling, paging, callbacks



# Exercise

- Follow instructions in Exercises handout for
  - Colorbox plugin
  - Rollover images
  - Cycle plugin



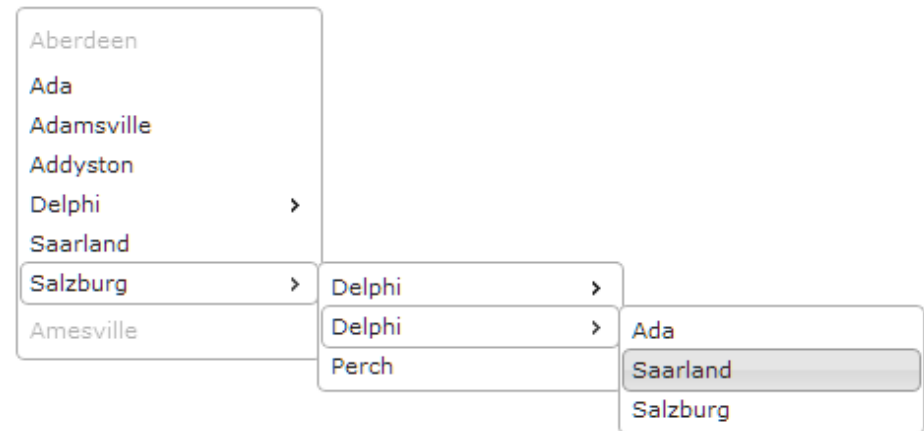
# **Menus, tabs, panels, dialogs & tooltips**

# Menus

- Expandable / collapsible
  - Use <nav> as a container
  - typically unordered lists with CSS styles
- CSS only
  - Listamatic 1 & 2, Listutorial –
  - <http://css.maxdesign.com.au/listamatic/>
  - <http://css.maxdesign.com.au/listutorial/>
- jQueryUI widget is basic
- needs delay for mouseout event

# Menus - html

- `$( '#menu' ).menu();`
- `<ul id='menu'>`
  - `<li class='ui-state-disabled'><a href='#'>Aberdeen</a></li>`
  - `<li><a href='#'>Delphi</a>`
    - `<ul>... for nested menu ...`
  - `</li>`

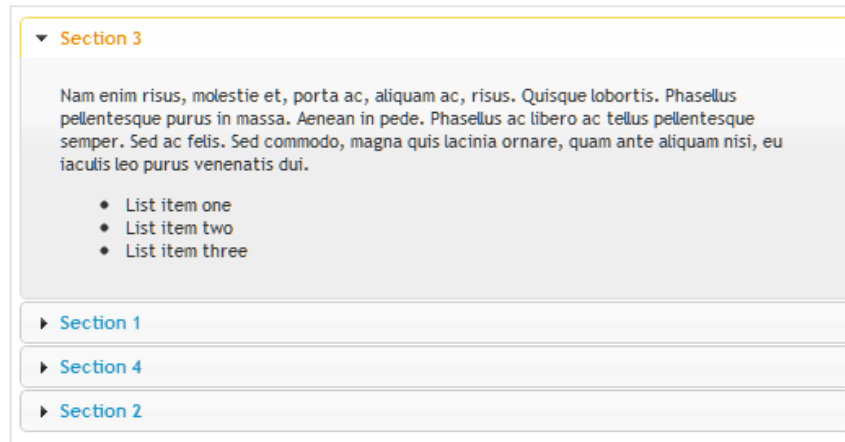


# Menus - plugins

- **Superfish** - Suckerfish style
  - [http://users.tpg.com.au/j\\_birch/plugins/superfish](http://users.tpg.com.au/j_birch/plugins/superfish)
- **Hover Intent** - UI for menus
  - <http://cherne.net/brian/resources/jquery.hoverIntent.html>
  - notes the speed of the cursor and only fires a hover if the cursor has slowed down enough

# Accordion

- Can use any containers
  - hide contents of all containers
  - show contents of active container
- Multiple levels
  - selectors use child selectors



Default functionality

Fill space

No auto height

Collapse content

Open on mouseover

Open on hoverintent

Customize icons

Sortable

# Tabs

- Panels with buttons on top
- One panel visible at a time

Nunc tincidunt

Proin dolor

Aenean lacinia

Proin elit arcu, rutrum commodo, vehicula tempus, commodo a, risus. Curabitur nec arcu. Donec sollicitudin mi sit amet mauris. Nam elementum quam ullamcorper ante. Etiam aliquet massa et lorem. Mauris dapibus lacus auctor risus. Aenean tempor ullamcorper leo. Vivamus sed magna quis ligula eleifend adipiscing. Duis orci. Aliquam sodales tortor vitae ipsum. Aliquam nulla. Duis aliquam molestie erat. Ut et mauris vel pede varius sollicitudin. Sed ut dolor nec orci tincidunt interdum. Phasellus ipsum. Nunc tristique tempus lectus.

## Default functionality

Content via Ajax

Open on mouseover

Collapse content

Sortable

Simple manipulation

Tabs below content

Cookie persistence



# Tabs - plugins

- EasyTabs.js
  - <http://os.alfajango.com/easytabs/>

# Panels and panes

- Container visually on top of another container
  - log-in form
  - email collection
  - quick survey
  - disclaimer
- Container under panel is usually visually subdued (faded, blurred, darkened)

# Panels and panes

- Notifications - Temporary small panels
- Dialogs – panels with buttons
- Modal – does not allow you to work with container under this panel

# Dialog - jQuery UI

- HTML button
  - `<button id='btn_openDialog'>Hated it!</button>`
- HTML dialog
  - `<div id='div_dialog' title='Are you sure?'>  
    <p>You've assigned the current celebrity a bad  
rating.</p>  
    <p>Perhaps you'd like to think this over./p>  
</div>`

# Dialog - jQuery UI

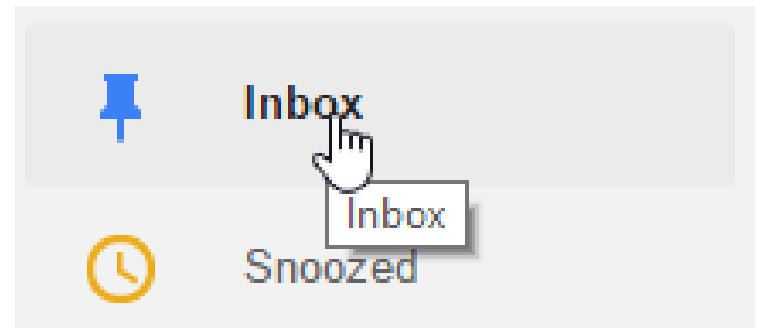
- `$('#div_dialog').dialog( {  
    autoOpen: false, height: 350,  
    modal: true, resizable: true,  
    buttons: {  
        'Continue': function ( ) {  
            $(this).dialog('close'); submitRating( ); },  
        'Change Rating': function ( ) {  
            $(this).dialog('close'); updateRating( ); }  
    }  
});`

# Dialog - jQuery UI

- jQuery code to open dialog
  - `$('#btn_openDialog').button( )  
 .on('click',function ( ) {  
 $('#div_dialog').dialog('open');  
 });`

# Tooltips

- Small temporary non-modal dialog triggered when hovering over element
  - Default browser behavior on links with **title** attributes



# Tooltips - jQueryUI

- `$( document ).tooltip({`
  - `show: { effect: 'slideDown', delay: 250 }`
- `});`
- `title='We ask for your age only for statistical purposes.'`

Your age:

Hover the field to see the tooltip

We ask for your age only for statistical purposes.



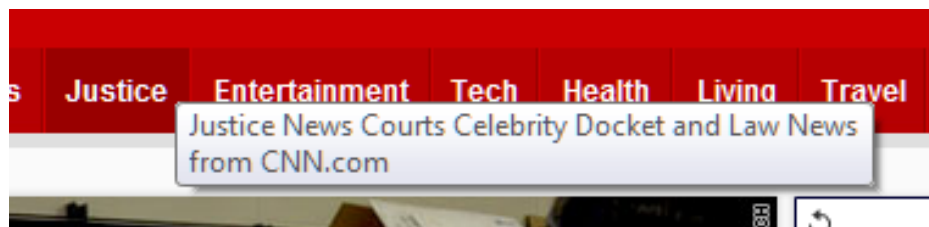
# Tooltips - plugins

- <http://www.1stwebdesigner.com/css/stylish-jquery-tooltip-plugins-webdesign/>
- Grumble - <http://jamescryer.github.com/grum>
- <http://craigsworks.com/projects/qtip2/>

BUBBLE-TASTIC!

Approx. Location	Price
<a href="#">Los Angeles</a>	\$39.95
<a href="#">New York</a>	\$39.95

Check out all the Celebs in LA this week!



A must have tool for designing better layouts and more intuitive user-interfaces.



# Panels - plugins

- Growl-style notifications
  - Gritter
    - <http://boedesign.com/demos/gritter/>
  - Toastr
    - <http://codeseven.github.io/toastr/demo.htm>
  - Toastr with FontAwesome icons
    - <http://plnkr.co/edit/6W9URNyyp2ltO4aUWzBB?p=preview>





# Exercises

- Follow instructions in Exercises handout for
  - Change a link style when on its own page



# Forms

# Forms

- Simple form validation
  - Don't trust your users
  - Don't only use jQuery to validate the data
- jQuery with forms should be used for
  - assisting input
  - validation of simple data input mistakes and revalidated in server code when submitted

# Form selectors

- `:input`
  - `$('#myForm:input').css('background-color', 'lemonchiffon')`
- `:text`, `:password`, `:radio`, `:checkbox`, `:submit`, `:button`,  
`:image`, `:file`
- `:enabled`, `:disabled`
- `:checked`, `:selected`

# .val( )

- `val( )`, `val(string)`
  - Gets the value or
  - Sets the value of an input, select or textarea field
- `$('select.foo option:selected').val( )`
  - `$('select.foo').val( ); // easier`
- `$('input:checkbox:checked').val( );`
- `$('input:radio[name=bar]:checked').val( )`

# .val( )

- val(**function(index, value)**)
- Use to calculate the value to set
- value = old value
- index = position of element in the array
- Always shows current value
  - html( ) will show original value even after user enters in new value



# Simple validation scripts

- Highlight required text fields

- `$(':submit').on('click',function (e) {`
- `$(':text').each(function () {`
- `if (! $(this).val().trim() ) { // not inputField's value`  
`withoutSpaces`
- `$(this).css('border', '2px solid red');`
- `e.preventDefault();`
- `} else {`
- `$(this).css('border', ' ');`
- `}`
- `});`
- `});`

# Simple validation scripts

- Tell user when leaving empty required field
  - `$(':input').blur(function() {`
  - `if (! $(this).val().trim()) {`
  - `$(this).addClass('error')`
  - `.after('<span class='error'>This field must ... </span>');`
  - `}`
  - `});`
  - `$(':input').focus(function() {`
  - `$(this).removeClass('error')`
  - `.next('span')`
  - `.remove();`
  - `});`

# Designing form validation

- Keep it simple.
  - Unless you plan it out well before
- Offer sample input and guidance
  - (mm-dd-yyyy)
  - a strong password uses a-z, A-Z and special characters

# The submit event

- Better than the click event, don't forget the Enter key
- `$('#form').on('submit', function( ) {  
 var error = false;`
- `$(this).find(':text').each(function( ) {`
- `if (! $(this).val().trim()) {`
- `alert('Textboxes must have a value!');`
- `$(this).focus( );`
- `error = true;`
- `return false; // Exit this function`
- `}`
- `});`
- `if (error) { return false; }`
- `});`

# Form validation with a plugin

- Best - <http://jqueryvalidation.org/>
- Workflow
  - Download and/or declare script element
  - Build the form with ids & names for each input
    - name binds to success message
    - id binds to selectors if used
  - Write the `$(selector).validate(...)` function body
  - Write CSS for `.error` and `.success`

# Form validation with a plugin

- `<div id='signup'><h2>Sign up</h2>`
- `<form action=''>`
- `<div><label for='name'>Name:</label>`  
`<input name='name' id='name' type='text' /></div>`
- `<div><label for='email'>Email:</label>`  
`<input name='email' id='email' type='text' /></div>`
- ...
- `<input type='submit' value='Submit!' />`
- `</form>`

# Form validation with a plugin

- `$('#signup form').validate({`
- **rules:** {
- `name: {required: true,},`
- `email: {required: true, email: true }, ...`
- **success:** `function(label) {`
- `label.text('OK!').addClass('valid');`
- `}`
- `});`

# Form validation with a plugin

- `input.error`
  - for highlighting the field to correct
- `label.error`
  - for the error message and/or graphic
- `input.valid`
  - for validated data in the field
- `label.goodjob`
  - for a 'good job!' message when adding the text & class in a success field of the `validate( )` arguments



# Validation predefined rules

- required, email, url, minlength, equalTo
- remote – for server side validation
- maxlength, rangelength, min, max, range
- date, dateISO, DateDE
- number, numberDE, digits
- creditcard, accept (file extension)
- minWords, maxWords, rangeWords
- phoneUS

# Maximum length indicator

- Problem: show remaining characters in textarea
- keypress event handler
  - e.which - which key was pressed for event
  - backspace = 8
- Plugin
  - <http://tpgblog.com/noblecount/>

# Checkbox state

- `.attr('checked')` is not correct
- `.prop('checked')`
  - or `.is(':checked')`
  - or `:checked` on element

# Check all checkboxes

- When any checkbox.check-all is checked, change all check boxes.
- `$('.check-all:checkbox').change(function( ) {  
    var group = ':checkbox[name='  
        + $(this).attr('name')  
        + '];'  
    $(group).prop('checked', $(this).  
prop('checked'));  
});`

# Inline editing

- Two workflows
  - Fake the style of the field
    - make fields look like label/p/div
    - when user clicks on one, style it back as a field
  - Replacing text with input element
    - when user click on label/p/div
    - replace field with input fields
- Expectations of spreadsheet style entry
- <http://www.appelsiini.net/projects/jeditable/default.html>

# Star ratings - plugins

- Rate Yo!

- <http://prrashigithubio/rateYo/>



- SVG, no images

- Star Rating plugin

- <http://www.fyneworks.com/jquery/star-rating/>

- Raty

- <http://www.wbotelhos.com/raty/>

# Slider - jQuery UI

- Based on **select + options** elements
- `var max = $('#max').val( ); var min = $('#min').val( );`
- `var rangeSlider = $('<div></div>')`
- `.slider({ min: 0, max: 100, step: 10, values: [min, max], range: true,`  
`animate: true,`
- `slide: function(e,ui) {`
- `$('#min').val(ui.values[0]);`
- `$('#max').val(ui.values[1]);`
- `showCelebs();`
- `} })`
- `.before('<h3>Drag the slider to filter by price:</h3>');`
- `$('#price-range').after(rangeSlider).hide( );`

Drag the slider to filter by price:



# Select box - jQuery UI

- Autocomplete
- `<input id='tags'>`
- ```
var availableTags = [ 'ActionScript',  
  'AppleScript', 'Asp', 'BASIC', 'C', 'C++',  
  'Clojure', 'COBOL', 'ColdFusion', 'Erlang',  
  'Fortran', 'Groovy', 'Haskell', 'Java',  
  'JavaScript', 'Lisp', 'Perl', 'PHP', 'Python',  
  'Ruby', 'Scala', 'Scheme' ];
```
- `$( '#tags' ).autocomplete({ source:  
 availableTags });`

Tags:

- AppleScript
- Clojure
- Erlang
- Haskell
- Perl
- Scheme



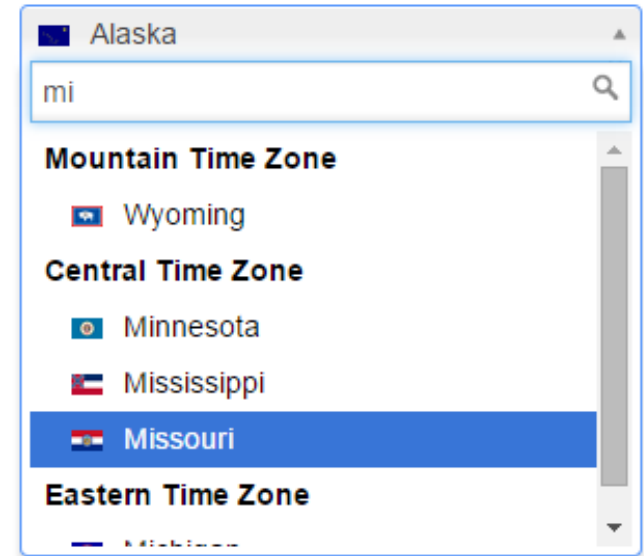
# Select box - plugin

- **\*Select2 v4**

- <https://select2.github.io/>
- Searchable options
- Multiple options
- Autocomplete
- Infinite scrolling
- Tagging

- [Chosen](#)

- <http://harvesthq.github.io/chosen/>



# Select box with common options

- Quick[select]

- <http://eggboxio.github.io/quick-select/>

|       |       |       |       |                   |
|-------|-------|-------|-------|-------------------|
| 06:30 | 07:00 | 07:30 | 08:00 | -- Pick a time -- |
|-------|-------|-------|-------|-------------------|

Quick[select] uses the browser's native select box on mobile devices.

-- Pick a time --

00:00  
00:15  
00:30  
00:45  
01:00  
01:15  
01:30  
01:45



INDIAN PIZZA CHINESE OTHER ▾

INDIAN PIZZA CHINESE Select one ▾

```
<select id="mdl-example"
  <option value=""
  <option value="I
  <option value="C
  <option value="P
```

Select one...  
Indian  
Chinese  
Pizza  
Fish and Chips  
Thai  
Fast Food

# Tagging - plugin

- [Tags Input](#) , [Selectize](#)
- \*Bootstrap tags input
  - <http://timschlechter.github.io/bootstrap-tagsinput/examples/>
  - Type and delimit, requires some autocomplete
- Tag-it 
  - Type and delimit, with autocomplete (requires jQuery UI)
  - <http://aehlke.github.io/tag-it/>
- TagBox 
  - <http://www.geektantra.com/2011/05/jquery-tagbox-plugin/>
  - Type and confirm, no autocomplete


# Autocomplete

- Typeahead.js
  - <http://twitter.github.io/typeahead.js/>
  - Better than the bootstrap typeahead (deprecated)
- Ajax AutoComplete
  - <https://www.devbridge.com/sourcery/components/jquery-autocomplete/>

# Button - jQuery UI

- `$( 'input:submit, a, button, input[type=button]',  
' .demo' ).button();`
- `<div class='demo'>`
  - `<button>A button element</button>`
  - `<input type='submit' value='A submit button'>`
  - `<a href='#>An anchor</a>`
- `</div>`

A button element

A submit button

An anchor

# Plugins - forms

- Chosen (better drop down menus)
  - <http://harvesthq.github.com/chosen/>
- Drop downs (select boxes) - searching, remote data sets, and infinite scrolling
  - <http://ivaynberg.github.com/select2/>
- Ideal Forms
  - <http://jqidealforms.com/>

# Plugins - forms

- <http://fuelyourcoding.com/scripts/infield/>
- <http://digitalbush.com/projects/masked-input-plugin/>
- <http://www.jquery4u.com/plugins/8-jquery-form-label-html5-placeholder-plugins/>
- <http://jquery.kuzemchak.net/togglevel.php>

# Plugins - forms

- Textarea
  - Autosize - <http://www.jacklmoore.com/autosize>
    - adjusts size as you type
- Clone
  - Cloneya - <http://yapapaya.github.io/jquery-cloneya>



# Plugins - forms

- float label pattern
  - Flylabel.js (above)
    - <http://athaeryn.github.io/flyLabel.js>
  - Floatlabels.js (inside)
    - <http://clubdesign.github.io/floatlabels.js/>
  - JvFloat.js
    - <http://bootsnipp.com/iframe/Ov>

Destroy all humans?

DESTROY ALL HUMANS?

Not today.

This is a floated Label. It stays there until the field is cleared.

Text text text

Name

My name

Email



# Exercise

- Form event testing
- Look at **props-vs-attr.html** for a visual display of why you don't use attributes for checkbox state testing
- Look at **checkboxes.html** for a generated set of checkboxes from data.



The common term for XHR

**AJAX**

# AJAX crash course

- old school request for some new data
  - dump the current page and ask the HTTP server for a new page
  - get rid of any user entered data on the current page unless you program it to stay
- AJAX request for some new data
  - sneak off while user is interacting with the page to ask any server via JSON/JSONP to get a little piece of data for you
  - update the part of the page you want to change

# AJAX



- Talk via browser to update current page data
- Talk via browser to update server data
- Talk via browser to load scripts or other resources
- Talk via server side language to add to composed page

# JSON

- a text data format
- JavaScript object declaration
- Converts into objects without mapping
  - { key: value, key2: value2, etc... }
  - values are data **or** functions

# JSON conversions

- convert an object to text (for screen)
  - `var JSONtext = JSON.stringify(anObject);`
  - `var partOfObject =  
anObject.bigpart.smallpart.data`
- convert JSON text back to an object
  - `var o = JSON.parse(JSONtext)`

# Load remote HTML

- Replace the contents of a container with a page body
  - `$('div:first').load('test.html');`
  - `$('#temp').load('weather_report.html#temperature');`
- From origin server only



# Load with GET/POST

- GET request
  - uses a query string
  - `$('#target').load('remotePage selector ', 'query string');`
  - `$('#temp').load('us_weather.html', 'zipcode=02134');`
- POST request
  - `$('#target').load('remotePage selector ', object );`
  - `$('#temp').load('us_weather.html',
    - { zipcode:02134, units:'F' });`

# Load – filter by selector

- load page and filter with selector
  - `$('#target').load(' remotePage selector ');`
  - `$('#biography').load('computadors.html div:first');`
  - Use for pseudo-database access
- load and show message
  - `$('#target').html('loading...').load(' remotePage selector ');`

# Fetching data

- When 3<sup>rd</sup> parties provide JSON methods (services), use
  - \$.getJSON( **serviceURL** , **callback function** )
  - or
  - jQuery.getJSON( **serviceURL** , **map for GET** , **callback function** )

# JSONP



- COR – Cross Origin Request being used more.
- JSONP (with padding) format allows cross-domain requests a
  - callback=? appended to URL... but not always that key
- JSON sites without P
  - Set up server –side proxy

# Delicious

- Base URL call
  - <http://feeds.delicious.com/v2/json>
  - retrieves bookmarks from hotlist
- Other calls
  - **base**/recent
  - **base**/tag
  - **base**/tag/**tag+tag+...**
  - **base**/popular
  - **base**/popular/**tag**
  - **base**/**username**

# Delicious

- `$.getJSON('http://feeds.delicious.com/v2/json/tag/centriq?callback=?',`
- `function (data) {`
- `console.log(data); // so you can drill down and see the data with`
- `F12`
- `$('#body').append('<h2>Fetched ' + data.length + ' items!</h2>');`
- `$(data).each(function (index) {`
- `$('#<a>', {`
- `href: this.u,`
- `text: this.d`
- `}).hide( )`
- `.appendTo('body').fadeIn( ) ;`
- `$('#body').append('<br/>');`
- `});`
- `});`

# Flickr

- Base URL call
  - [http://api.flickr.com/services/feeds/photos\\_public.gne?jsoncallback=?](http://api.flickr.com/services/feeds/photos_public.gne?jsoncallback=?)
- Object map for GET
  - { tags: 'cat, grumpy', format: 'json' }

# Flickr



```
var commaSeparatedSearchTags = 'cat, grumpy';
var flickrBase =
'http://api.flickr.com/services/feeds/photos_public.gne?jsoncallback=?';
$.getJSON(flickrBase, { tags: commaSeparatedSearchTags, format: 'json' },
function (data) {
    console.log(data);
    $('<h1>').text(data.title)
    .appendTo('body');
    $(data.items).each(function ( ) {
        $('<a>', {href: this.link}).append(
            $('<img>', { src: this.media.m, style: 'height: 200px; padding:2px'
        )))
        .appendTo('body');
    });
});
```



# The jQuery Ajax workhorse

- `$.ajax( { } );`
- very customizable, all other Ajax functions wrap around this one

# Common Ajax settings

- Reusable settings for multiple `$.ajax( )` calls
- `$.ajaxSetup ( { } );`

# **.ajax( ) for JSON**

- \$.ajax({ url: *url*, dataType: 'json', data: *data*, success: *callback* });
  - \$.getJSON( ) is a shortcut
- Datatypes  
include xml, html, json, jsonp, script, or text.
- If the URL includes 'callback=?' requests are treated as JSONP

# **.ajax( ) error handling**

- \$.ajax({  
    url: *url*,  
    dataType: 'jsonp',  
    data: *data*,  
    timeout: 15000,  
    success: *callback*,  
    error: function( ) {  
        \$response.html(failed); }  
});

# Loading external scripts

- `$.getScript( script URL )`
- delay loading code until necessary
- loads scripts from server as needed

# GET and POST methods

- Shorthand methods of `$.ajax( )`
- `$.get( url, data, callback, dataType );`
- `$.post( url, data, callback, dataType );`

# Links - Web services

- Get a Google API key
  - <https://console.developers.google.com>

# Links - APIs

- <http://www.programmableweb.com/apis/directory>
- TwitStream plugin
  - <http://iamnoahcooper.com/TwitStream/>



# Debugging AJAX

- <http://fixingthesejquery.com/#slide33> and do rest of deck



# Exercise

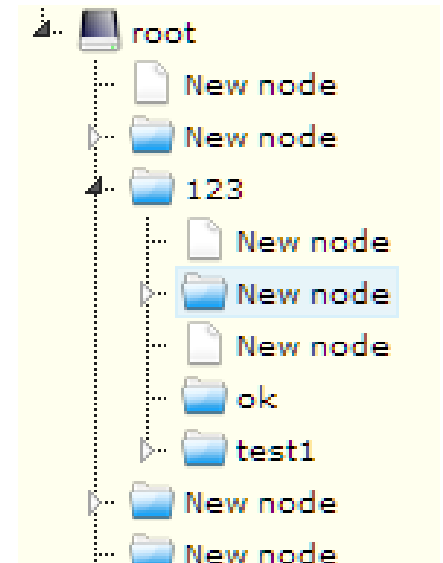
- Copy and play with
  - flickr-grumpy-cats.html



# **Lists, Trees, and Tables/Grids**

# Plugins - trees

- jsTree - <http://www.jstree.com/>
  - theme support, search, AJAX, drag & drop...
- <https://github.com/pioz/jquery-tree>



# Plugins - tables

- Front-end managed data display components
  - requires data feed from back end, usually a REST service
  - not needed if you are doing back-end extraction and sorting
- Targeted towards the desktop
  - detail and small components pack in functionality such as search boxes, pagination, sort icons, etc.
  - not appropriate for touch interface

[Live example](#)

Show  entries

Search:

Rendering engine ▲	Browser	Platform(s)	Engine version	CSS grade
Gecko	Firefox 1.0	Win 98+ / OSX.2+	1.7	A
Gecko	Firefox 1.5	Win 98+ / OSX.2+	1.8	A
Gecko	Firefox 2.0	Win 98+ / OSX.2+	1.8	A
Gecko	Firefox 3.0	Win 2k+ / OSX.3+	1.9	A
Gecko	Camino 1.0	OSX.2+	1.8	A
Gecko	Camino 1.5	OSX.3+	1.8	A
Gecko	Netscape 7.2	Win 95+ / Mac OS 8.6-9.2	1.7	A
Gecko	Netscape Browser 8	Win 98SE+	1.7	A
Gecko	Netscape Navigator 9	Win 98+ / OSX.2+	1.8	A
Gecko	Mozilla 1.0	Win 95+ / OSX.1+	1	A

Showing 1 to 10 of 57 entries

Previous Next

# Plugins - tables

- DynaTable
  - More JSON compatibility
  - <http://www.dynatable.com/>
- Handsontable (non jQuery)
  - minimal Excel inline style editing
  - <http://handsontable.com>

# Plugins – tables

- \*DataTables
- <http://www.datatables.net/>
  - Check cdnjs.com for associated modules

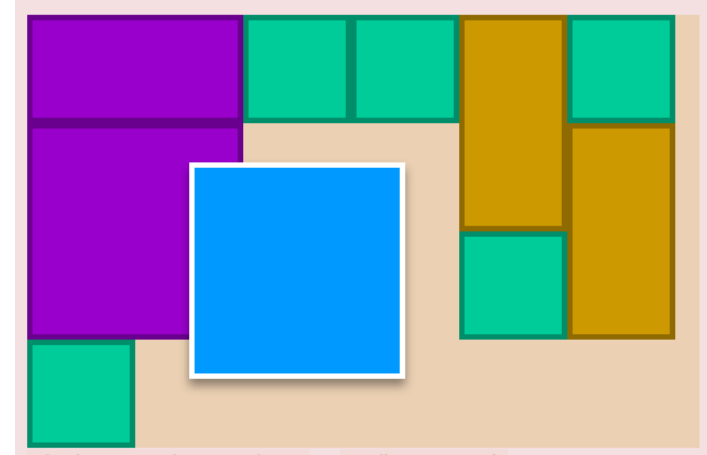
# Plugins – pagination with MVC

- PagedList (.NET)
  - <https://www.nuget.org/packages/PagedList.Mvc>



# Plugins – layout

- Pinterest style
- \*Packery – no gaps
  - <http://packery.metafizzy.co/>
- Minigrid – simple
  - <http://alves.im/minigrid/>
  -



# Plugins – layout

- Magnet
  - <http://codecanyon.net/item/magnet-jquery-plugin-for-filterable-layouts>
- Windows8 look
  - <http://sarasoueidan.com/blog/windows8-animations/demo/index.html>



# Exercises

- DataTables



# jQuery UI Interactions

# Drag and drop - JQueryUI

- in CSS set **cursor: move** on draggable elements
- workflow:
  - set up html image elements
  - set up jQuery draggable & droppable behaviors on those images
  - set up jQuery special animations - optional
  - set up CSS visual hints for cursor, activeClass, hoverClass

# Drag and drop - JQueryUI

- draggable
  - events - start, stop, drag
- droppable
  - activeClass:, hoverClass:
  - accept: - a list of valid draggables
  - events – over, out, drop

# Drag and drop - JQueryUI

- CSS

- #paper { cursor:move; }

- jQuery

- ```
$('#paper, #paper2').draggable({
    revert: 'invalid', cursor: 'move' , containment: 'parent' });
$('#trash').droppable({
    drop: function (event, ui) {
        $(ui.draggable).hide('slow');
        console.log($(event));
        console.log($(ui));
    }
});
```

# Drag and drop - JQueryUI

- Use Google images to find trash and crumpled paper images
- HTML
  - ```
<div id='dragAndDrop1' style='height:300px'>  
  <img id='trash' src='?' style='height:200px;  
float:right' />  
  <img id='paper' src='?' style='height:100px;' />  
  <img id='paper2' src='?' style='height:100px;' />  
</div>
```



# Drag and drop - JQueryUI

- Other options
  - axis – to drag on X or Y axis only
  - containment – to drag within a restricted area
  - handle – to define an area to drag for a larger element
  - opacity / helper – show a shadowed version when dragging

# Drag and drop - plugins

- Draggabilly
  - <http://draggabilly.desandro.com/>
  - <https://github.com/chienhungchen/droppabilly>
- Pep
  - <http://pep.briangonzalez.org/>

# Resizing - JQueryUI

- <http://jqueryui.com/demos/resizable/>
- Resizable <textarea> with options
  - `$('#textarea').resizable({  
 grid : [20, 20],  
 minWidth : 153,  
 minHeight : 30,  
 maxHeight : 220,  
 containment: 'parent'  
});`



# Sorting - JQueryUI

- A special kind of droppable
  - all children are draggable and will reorder
  - no permanent sorting involved
- HTML
  - ```
<ul id='fruit-list'>  
  <li id='appleID'>Apple</li>  
  <li id='bananaID'>Banana</li>  
  <li id='cherryID'>Cherry</li> </ul>
```
- CSS
  - ```
li { padding:.5em; margin-bottom: 1em; border: solid; width:  
100px; }
```

# Sorting - JQueryUI

- jQuery
  - `$('#fruit-list').sortable({`
  - `update: function (event, ui) {`
  - `var fruitOrder =`
  - `$(this).sortable('toArray').toString();`
  - `$('#fruit-`
  - `list').parent().append($('`
  - `}`
  - `});`

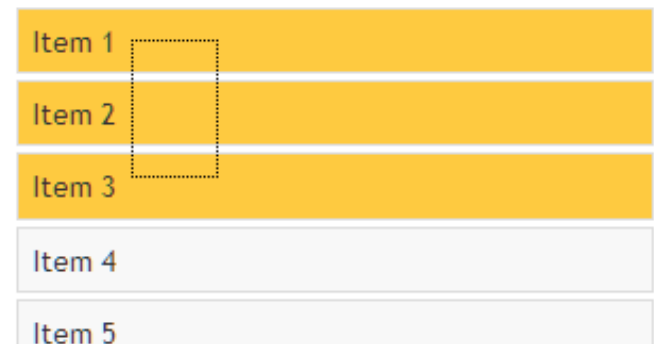
# Sorting - plugins

- jQuery Sortable
  - <http://johnny.github.io/jquery-sortable/>

# Selectable

- Select by dragging a box (lasso) over elements
- `selectable( )` function
  - apply to the `<ul>` container
  - applies classes `ui-selecting` and `ui-selected`
  - event sequences
    - start, selecting, ... , selected, stop
    - start, unselecting, ... , unselected, stop

You've selected: #2 #3 #4 .



# Libraries/Plugins - Touch

- \*hammer.js
- jQuery.event.swipe
  - from jQuery Mobile custom download
- TouchPunch
- Touchy.js
  - <https://github.com/jairajs89/Touchy.js>





# Exercise

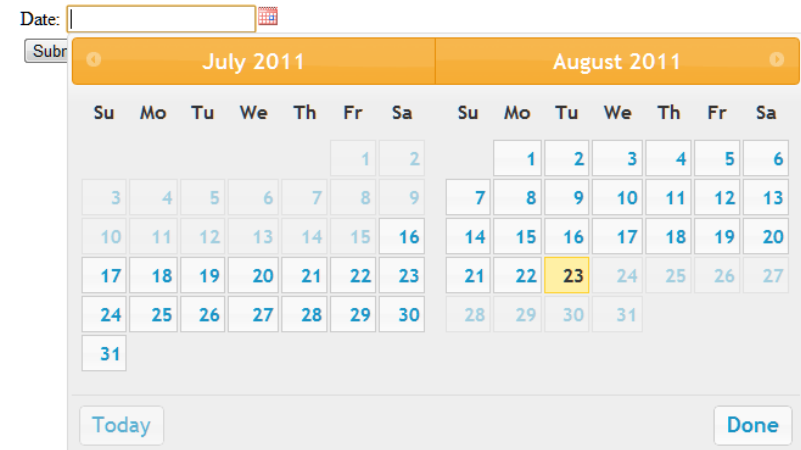
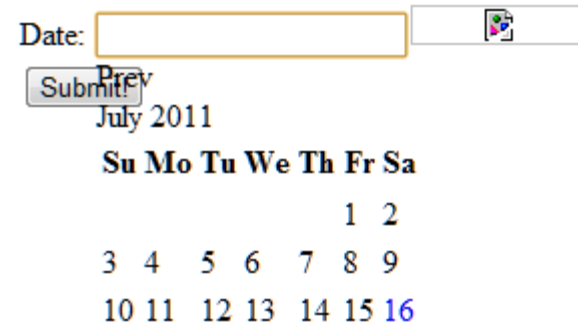
- Follow instructions in Exercises handout for
  - 25. Drag & drop



# jQuery UI Controls

# Datepicker

- `$('#date').datepicker({  
   showOn: 'button',  
   buttonText: 'Choose a date',  
   buttonImage: 'calendar.png',  
   buttonImageOnly: true,  
   numberOfMonths: 2,  
   maxDate: '0d',  
   minDate: '-1m -1w',  
   showButtonPanel: true  
 });`
- Needs a theme!
- `setDefaults( )`, `parseDate( )`, `formatDate( )`



# Plugins - controls


- Datepicker
  - <http://www.kelvinluck.com/assets/jquery/datePicker/v2/demo/>

Date 1:

Date 2:


Test select:

**Page sourcecode**



August 2011						
M	T	W	T	F	S	S
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31	1	2	3	4

# Progress bar

- One block element to show
  - `<div id='bar'></div>`
- jQuery 
  - `$('#bar').progressbar( { value: 10 } );`
- jQuery updating the bar
  - ```
var i = 0;
var intervalID = setInterval(moveBar, 100);
function moveBar() {
    $('#bar').progressbar('value', i +=5 );
    if (i == 100) { clearInterval(intervalID); }
}
```

# 1.9 – 1.12 Controls

- Spinner -  
<http://wiki.jqueryui.com/w/page/121380777/Spinner>
- Tooltip -  
<http://wiki.jqueryui.com/w/page/12138112/Tooltip>
- Menu -  
<http://view.jqueryui.com/master/tests/visual/menu/menu.html>
- [Controlgroup](#) replaces buttonset and adds support for selectmenu
- [CheckboxRadio](#)

# Exercise



- Follow instructions in Exercises handout for
  - 35. Dialogs and a datepicker



**End matter**





# Plugin aggregators

- The jQuery Plugin Registry
  - <http://plugins.jquery.com/>
- jQuer.in
  - <http://jqquer.in/>

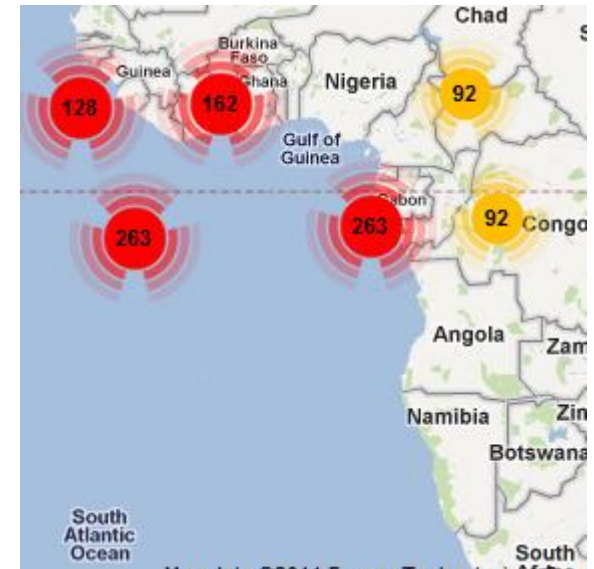
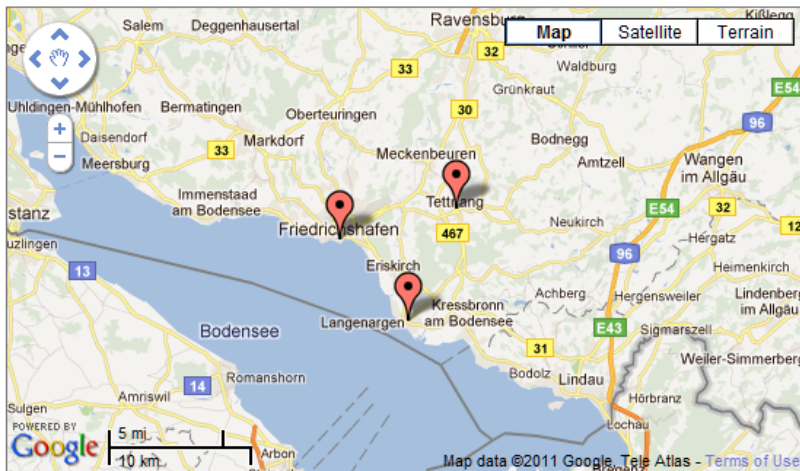
# Plugins – Files

- HTML5 Uploader (file upload)
  - <http://www.igloolab.com/jquery-html5-uploader/>
- Uploads
  - <http://tutorialzine.com/2011/09/html5-file-upload-jquery-php/>
- Filesystem API (Webkit and future browsers)
  - filer - <https://github.com/ebidel/filer.js>

# Plugins - Maps

- <http://gmap.nurtext.de/>
- <http://code.google.com/p/jquery-ui-map/>
- [http://www.mosne.it/playground/mosne\\_map/](http://www.mosne.it/playground/mosne_map/)

```
$("#map4").gMap({ markers: [{ latitude: 47.651968,
                           longitude: 9.478485,
                           html: "_latlng" },
                          { address: "Tett nang, Germany",
                           html: "The place I live" },
                          { address: "Langenargen, Germany",
                           html: "_address" }],
  address: "Braitenrain, Germany",
  zoom: 10 });
```



# Plugins - Maps

- Mapit - <http://vdw.github.io/MapIt/>

# Frameworks

- jQuery UI
  - no touch event support
    - Use simple plugin - <http://touchpunch.furf.com/>
    - Use better Hammer.js - <http://hammerjs.github.io/>
- Wijmo – full UI library to extend jQuery
  - <http://wijmo.com/>
  - no touch events
  - \$495/795/895/1195

# Sorting - packages

- List.js (3k)
  - <http://listjs.com/>
  - lists/tables searchable, sortable and filterable

# Directories

- Uibox - <http://www.uibox.in/>
- The Badass List of Essential jQuery plugins - <http://hasinhayder.github.io/essential-jquery-plugins/>

# Libraries - routing & history

- BBQ - Back Button and Query  
<http://benalman.com/projects/jquery-bbq-plugin/>
- Crossroads  
<http://millermedeiros.github.com/crossroads.js/> and Hasher (history)



# Libraries - dependency script loading



- require.js
  - <http://requirejs.org/>
- manages sequence of script modules
- combines scripts for deployment
- loads code as needed
- Alternatives:
  - [Shepherd.js](#)

# Libraries - misc

- PDF generation
  - <http://parall.ax/products/jspdf>
- Table to JSON
  - <http://lightswitch05.github.io/table-to-json>
- Text editor
  - <http://hallojs.org/> - simple in-place
- jQueryUI replacement libraries
  - <http://jquery-jkit.com/>

# Plugin collections

- <http://www.thepetdesign.com/>
- <http://www.unheap.com/>

# Learning

- Tutorials
  - <http://jqfundamentals.com/>
- Conferences
  - <http://events.jquery.org/>
  - <http://lanyrd.com/topics/jquery/>

# Last tasks

- Certificate
- Evaluations
  - MTM link on desktop
  - <http://www.MetricsThatMatter.com/centriqfoss1>
- Take your code