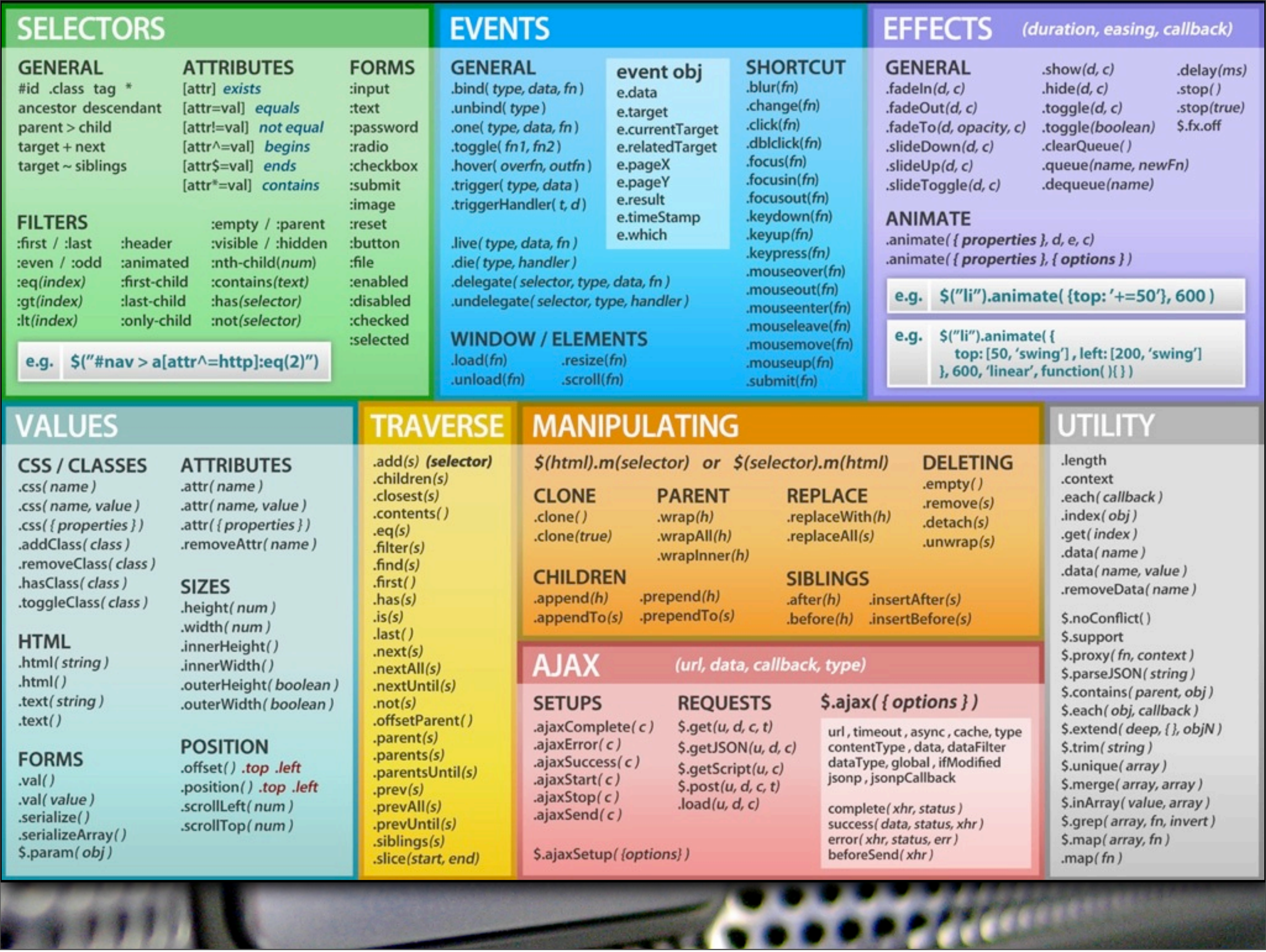
FULL SAIL
UNIVERSITY

scripting for web applications

jQuery UI

PWA-2 DUE Dates

Item	Due Dates
Branding / Logo	11/25/13 - After Lab on the First Day
Project Pitch	12/02/13 - Before Lecture 2
Creative Brief - Finished Document	12/4/13 - Before Lecture 3
Site Prototype (html/css)	12/09/13 - Before Lecture 5
Development Milestone (javascript)	12/13/13 - Due End of Lab 7
Inclusion of 5 media center items	12/20/13 - Last Day of Class After Lab
Aesthetics & Usability (finished site)	12/20/13 - Last Day of Class After Lab
Functionality (finished site)	12/20/13 - Last Day of Class After Lab
Professionalism	The duration of the course
Class Participation	The duration of the course



SELECTORS

GENERAL
#id .class tag *
ancestor descendant
parent > child
target + next
target ~ siblings

ATTRIBUTES
[attr] *exists*
[attr=val] *equals*
[attr!=val] *not equal*
[attr^=val] *begins*
[attr$=val] *ends*
[attr*=val] *contains*

FORMS
:input
:text
:password
:radio
:checkbox
:submit
:image
:reset
:button
:file
:enabled
:disabled
:checked
:selected

FILTERS
:first / :last
:even / :odd
:eq(index)
:gt(index)
:lt(index)

:header
:animated
:first-child
:last-child
:only-child

:empty / :parent
:visible / :hidden
:nth-child(num)
:contains(text)
:has(selector)
:not(selector)

e.g. $("#nav > a[attr^=http]:eq(2)")

EVENTS

GENERAL
.bind(type, data, fn)
.unbind(type)
.one(type, data, fn)
.toggle(fn1, fn2)
.hover(overfn, outfn)
.trigger(type, data)
.triggerHandler(t, d)

.live(type, data, fn)
.die(type, handler)
.delegate(selector, type, data, fn)
.undelegate(selector, type, handler)

event obj
e.data
e.target
e.currentTarget
e.relatedTarget
e.pageX
e.pageY
e.result
e.timeStamp
e.which

SHORTCUT
.blur(fn)
.change(fn)
.click(fn)
.dblclick(fn)
.focus(fn)
.focusin(fn)
.focusout(fn)
.keydown(fn)
.keyup(fn)
.keypress(fn)
.mouseover(fn)
.mouseout(fn)
.mouseenter(fn)
.mouseleave(fn)
.mousemove(fn)
.mouseup(fn)
.submit(fn)

WINDOW / ELEMENTS
.load(fn)
.unload(fn)
.resize(fn)
.scroll(fn)

EFFECTS (duration, easing, callback)

GENERAL
.fadeIn(d, c)
.fadeOut(d, c)
.fadeTo(d, opacity, c)
.slideDown(d, c)
.slideUp(d, c)
.slideToggle(d, c)

.show(d, c)
.hide(d, c)
.toggle(d, c)
.toggle(boolean)
.clearQueue( )
.queue(name, newFn)
.dequeue(name)

.delay(ms)
.stop( )
.stop(true)
$.fx.off

ANIMATE
.animate({ **properties** }, d, e, c)
.animate({ **properties** }, { **options** })

e.g. $("li").animate({top: '+=50'}, 600)

e.g. $("li").animate({
top: [50, 'swing'] , left: [200, 'swing']
}, 600, 'linear', function( ){ })

VALUES

CSS / CLASSES
.css(name)
.css(name, value)
.css({ properties })
.addClass(class)
.removeClass(class)
.hasClass(class)
.toggleClass(class)

HTML
.html(string)
.html( )
.text(string)
.text( )

FORMS
.val( )
.val(value)
.serialize( )
.serializeArray( )
$.param(obj)

ATTRIBUTES
.attr(name)
.attr(name, value)
.attr({ properties })
.removeAttr(name)

SIZES
.height(num)
.width(num)
.innerHeight( )
.innerWidth( )
.outerHeight(boolean)
.outerWidth(boolean)

POSITION
.offset( ) .top .left
.position( ) .top .left
.scrollLeft(num)
.scrollTop(num)

TRAVERSE
.add(s) **(selector)**
.children(s)
.closest(s)
.contents( )
.eq(s)
.filter(s)
.find(s)
.first( )
.has(s)
.is(s)
.last( )
.next(s)
.nextAll(s)
.nextUntil(s)
.not(s)
.offsetParent( )
.parent(s)
.parents(s)
.parentsUntil(s)
.prev(s)
.prevAll(s)
.prevUntil(s)
.siblings(s)
.slice(start, end)

MANIPULATING
$(html).m(selector) or $(selector).m(html)

CLONE
.clone( )
.clone(true)

PARENT
.wrap(h)
.wrapAll(h)
.wrapInner(h)

REPLACE
.replaceWith(h)
.replaceAll(s)

DELETING
.empty( )
.remove(s)
.detach(s)
.unwrap(s)

CHILDREN
.append(h)
.appendTo(s)
.prepend(h)
.prependTo(s)

SIBLINGS
.after(h)
.before(h)
.insertAfter(s)
.insertBefore(s)

AJAX (url, data, callback, type)

SETUPS
.ajaxComplete(c)
.ajaxError(c)
.ajaxSuccess(c)
.ajaxStart(c)
.ajaxStop(c)
.ajaxSend(c)

$.ajaxSetup({options})

REQUESTS
$.get(u, d, c, t)
$.getJSON(u, d, c)
$.getScript(u, c)
$.post(u, d, c, t)
.load(u, d, c)

$.ajax({ options })
url , timeout , async , cache, type
contentType , data, dataFilter
dataType, global , ifModified
jsonp , jsonpCallback

complete(xhr, status)
success(data, status, xhr)
error(xhr, status, err)
beforeSend(xhr)

UTILITY
.length
.context
.each(callback)
.index(obj)
.get(index)
.data(name)
.data(name, value)
.removeData(name)

$.noConflict( )
$.support
$.proxy(fn, context)
$.parseJSON(string)
$.contains(parent, obj)
$.each(obj, callback)
$.extend(deep, {}, objN)
$.trim(string)
$.unique(array)
$.merge(array, array)
$.inArray(value, array)
$.grep(array, fn, invert)
$.map(array, fn)
.map(fn)

jQuery UI

review jqueryui.com

jQuery UI

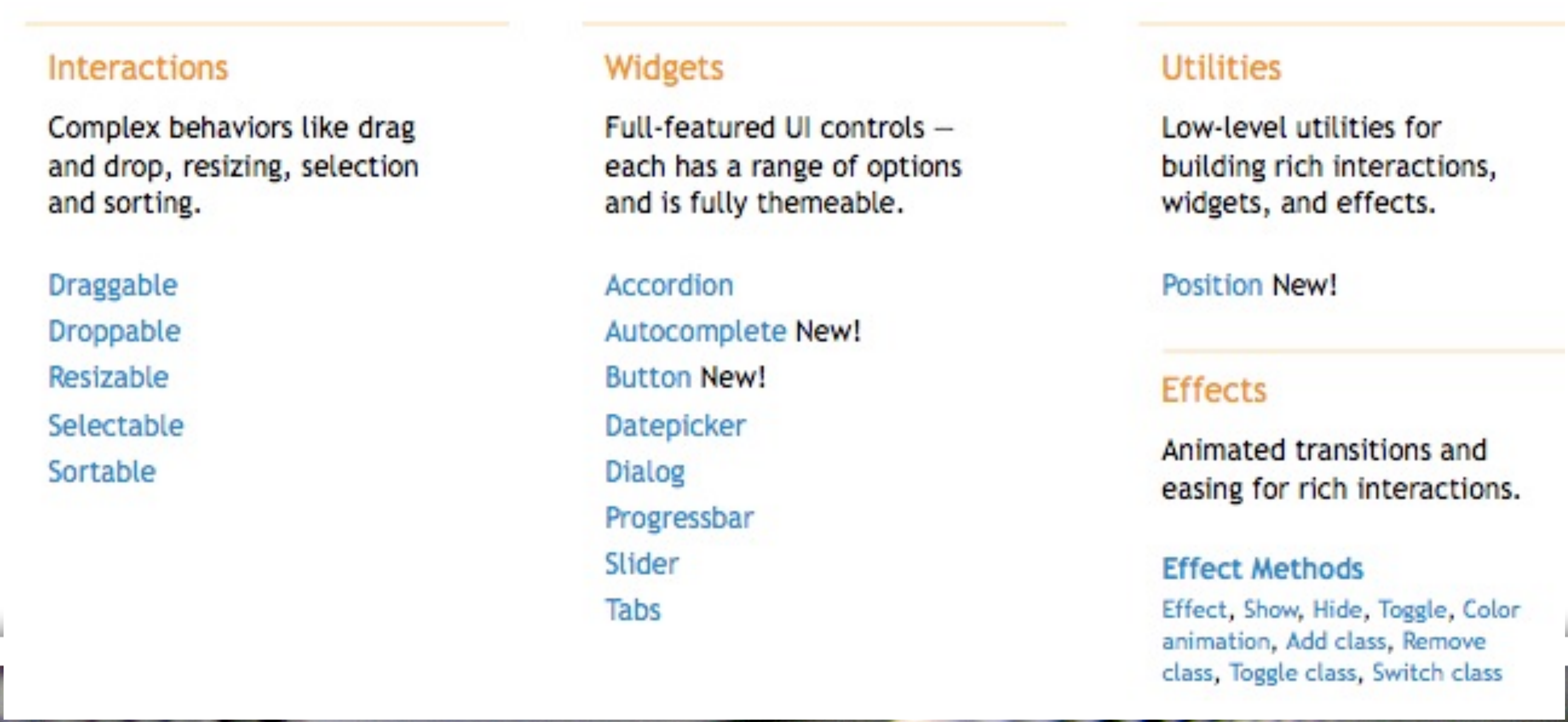
- UI v-1.7.x
  - Version 1.7.x fixed alot of the bugs that 1.6.x had created out of trial/error, and brought the collection into jQuery 1.3.x. As developers, following the naming conventions that jQuery employs makes our own code more friendly
- UI v-1.8.x
  - To coincide with jQuery 1.4.x, the UI platform released v1.8, with efficiency tweaks and a couple new features.
- UI v-1.9.x
  - Add a nested menu widget, a spinner widget, and a Tooltip widget. Many enhancements made to other existing widgets. Improved API to improve consistency across plugins.

jQuery UI

- UI v-1.10.x
  - Version 1.10 includes dozens of bugs fixes and improved accessibility. The dialog and progressbar widgets have undergone API redesigns, making them easier to use and create more consistency across plugins. Compatible with jQuery 1.6+

jQuery UI

- jQuery UI Collection



Interactions

Complex behaviors like drag and drop, resizing, selection and sorting.

Draggable
Droppable
Resizable
Selectable
Sortable

Widgets

Full-featured UI controls – each has a range of options and is fully themeable.

Accordion
Autocomplete New!
Button New!
Datepicker
Dialog
Progressbar
Slider
Tabs

Utilities

Low-level utilities for building rich interactions, widgets, and effects.

Position New!

Effects

Animated transitions and easing for rich interactions.

Effect Methods

Effect, Show, Hide, Toggle, Color animation, Add class, Remove class, Toggle class, Switch class

jQuery UI

- UI Cores
  - The Widgets and Interactives make up the bulk of UI, and here is how they all break down into unified sets of usage conventions. See the "API Documention" link on each page for each method.
    - **Overview:** *Basic information on the front page of how the plugin works.*
    - **Options:** *An object-literal set of options to customize the plugin.*
    - **Events:** *Custom events the plugin fires, that you can attach handlers to.*
    - **Methods:** *Public API of the plugin to change it after initialization.*
    - **Example:** *Sample jQuery UI and HTML code.*

jQuery UI

jQuery UI - Useful Resources:

The ThemeRoller App:	http://jqueryui.com/themeroller/
Documentation on Custom ThemeRolls:	http://jqueryui.com/docs/Theming
jQuery UI Planning Blog:	http://wiki.jqueryui.com/
Developer Guide	http://jqueryui.com/docs/Developer_Guide

UI Sortables

- jQuery UI: Sortables

target.sortable(*options*)	The sortable plugin uses an html list target as a base, and automatically assigns *drag/drop* functionality to the *target*. The options argument is an object of properties and callbacks to initialize the list with.

- The *draggable* and *droppable* plugins are actually the foundation for *sortable*, but are much more raw. Sortable comes with some useful pre-defined functionality, specifically with how it takes a list and automatically performs snapping.
- Sortable also has an extra function allowing connection between two or more lists, which is exactly what we need for our project.

Lab 6 *(no turn-in today)*

lab begins in 50 min

- **Next Milestone:** Developement Milestone
  - Login / Logout should both work
  - Logging in changes the page to the App UI
  - Logging out changes the page to the Landing UI
  - App UI loads all necessary content to ***fill the page*** *(a list of projects or tasks)*
  - New user registration should be working *(and should log the user in automatically)*
  - **Due at the END of Lab 7 as: *lastname_firstname_milestone.zip***