

DIG4503: Rapid Application Development

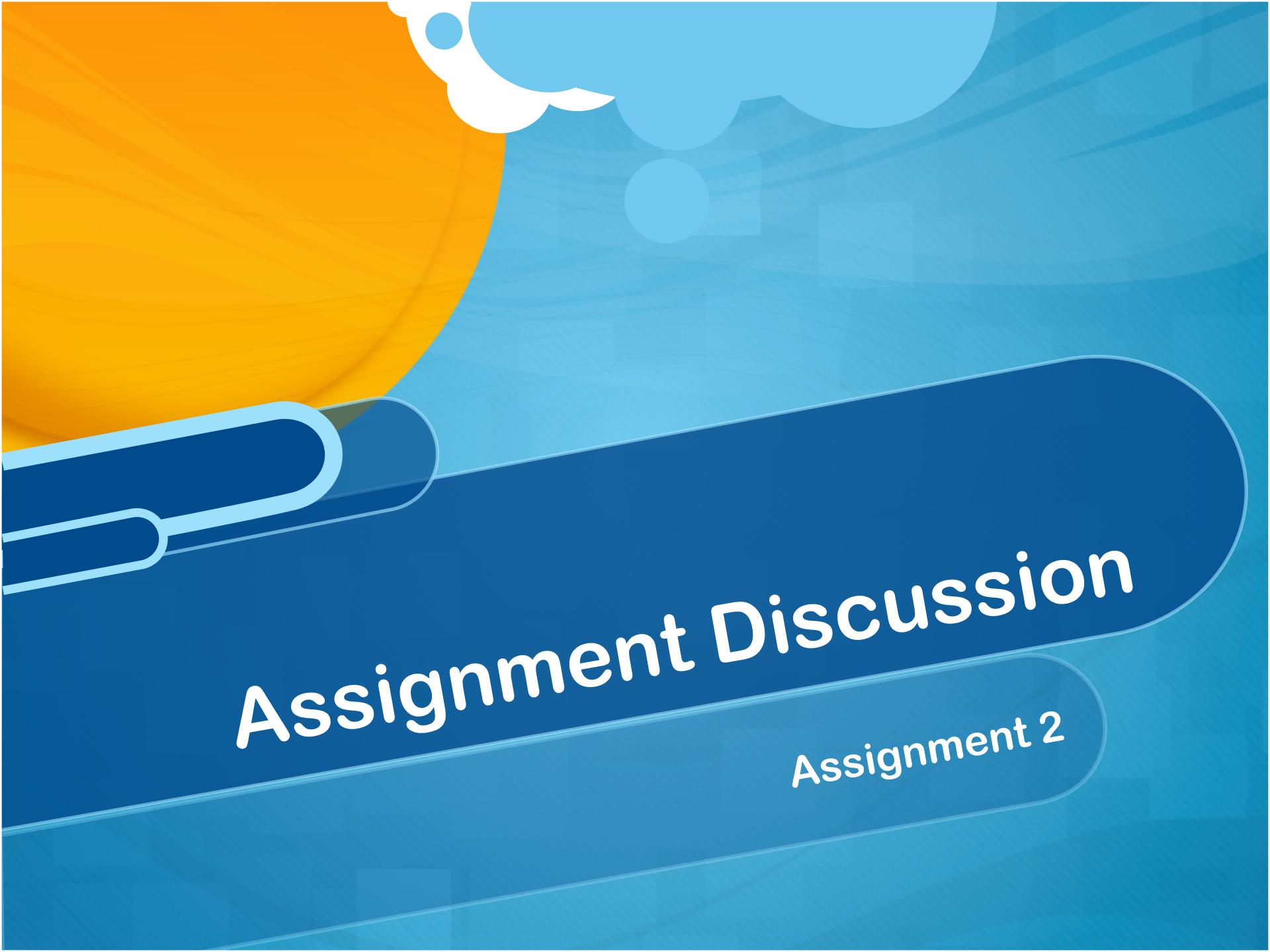
Dan Novatnak
Week 7

Last Time

- JavaScript Libraries, Frameworks, & Toolkits
 - What are they
 - Choosing the right one for a project
 - Comparisons / contrasts
 - Usage
- Finally...the jQuery library

This time

- Assignment 2 Discussion
- jQuery AJAX
 - AJAX
 - Get, post
 - handling data
- jQuery UI
 - Overview
 - Installation
 - Features
- Conversion from “Vanilla” JS to jQuery



Assignment Discussion

Assignment 2

Assignment 2

- According to the syllabus:
 - Project jQuery Framework
 - In this assignment you are tasked with the design and development of a web application that leverages the techniques of AJAX through the jQuery Library.
 - Database data must be retrieved, manipulated, and returned though asynchronous means.
 - Coded site, delivered on the server.
 - This is an individual assessment.

I figure that this is rational

- Reconfiguration, extension, and conversion of Assignment 1 ...or...
- Fuller richer more responsive an media savvy project ...or...
- One of these quizzes...maybe build one
- Discussion?

AS2 Components

- Required
 - AJAX
 - jQuery
- Optional
 - jQuery UI elements



AJAX just got easier

AJAX with jQuery

Weeks ago we learned AJAX manually – it was work

- XMLHttpRequest object
 - .open() .close() .onreadystatechange .readyState .status .responseText
- Create an XMLHttpRequest object the multi-browser way
- Get whatever data you need from the Web form.
- Build the URL to connect to.
- Open a connection to the server.
- Set up a function for the server to run when it's done.
- Send the request

Now that we have the jQuery Libraries

- jQuery, the JavaScript library provides a powerful set of jQuery AJAX API's to handle AJAX requests.
- Let's face it:
 - The manual way of making AJAX calls using JavaScript is a bit odd as you have to first create that XMLHttpRequest object that depends on the browser and then make an AJAX call.
 - Sending a form data using AJAX is also bit difficult using manual JavaScript

Enter jQuery

- jQuery provides simple yet powerful functions
 - `$.get`, `$.post` `$.ajax`, ...etc
- which have extended the JavaScript AJAX methods in a more flexible way.
- http://www.w3schools.com/jquery/jquery_ref_ajax.asp

`$.get()`

- `$.get(URL,data,function(data,status,xhr),dataType)`
 - `url`: (`String`) The URL of the page to load.
 - `data` (`Optional`): (`Map`) Key/value pairs that will be sent to the server.
 - `callback` (`Optional`): (`Function`) A function to be executed whenever the data is loaded successfully.
 - `type` (`Optional`): (`String`) Type of data to be returned to callback function: “`xml`”, “`html`”, “`script`”, “`json`”, “`jsonp`”, or “`text`”.

`$.get()` example

```
0 $.get(  
0   "http://some-remote-site",  
0   "{key:value}",  
0   function(data) { alert(data); },  
0   "html"  
0 );
```

`$.get()` live example

- `01_get_jquery.html`
- Grabs data from file
- Alerts the data

\$.load()

- `$(selector).load(url,data,function(response,status,xhr))`
 - url: Required. Specifies the URL you wish to load
 - data: Optional. Specifies data to send to the server along with the request
 - function(response,status,xhr): Optional. Specifies a callback function to run when the load() method is completed.
- Accomplishes the same as get and you can just load data directly to an element

\$.load live example

- Accomplishes the same as get
- Can just load data directly to an element
 - [02_load_jquery.html](#)
- Yea...that was one step...boom.

`$.post()`

- Sending post method is also very easy with jQuery.
- All you have to do is just to call `.post ()` method instead of `.get ()`

`$.post()`

- `$.post(url, [data], [callback], [type])`
 - `url`: (`String`) The URL of the page to load.
 - `data` (`Optional`): (`Map`) Key/value pairs that will be sent to the server.
 - `callback` (`Optional`): (`Function`) A function to be executed whenever the data is loaded successfully.
 - `type` (`Optional`): (`String`) Type of data to be returned to callback function: “`xml`”, “`html`”, “`script`”, “`json`”, “`jsonp`”, or “`text`”.

`$.post()` live example

- `03_post_jquery.html`
- Grabs data from file
- Alerts the data

Get JSON using jQuery

- Many said Tuesday that it was a difficult thing to manipulate JSON in their projects
- jQuery provides a function that can be used to make an AJAX call and get the data in JSON format.
- Normally the data that we get from AJAX has to be converted in JSON by calling eval () method of JavaScript.
- Yea, the function provided by jQuery handles this internally and provides direct JSON object as output.

`$.getJSON()`

- `$.getJSON(url, [data], [callback])`
- **url:** Required. Specifies the url to send the request to
- **data:** Optional. Specifies data to be sent to the server
- **success(data,status,xhr):** Optional. Specifies the function to run if the request succeeds

`$.getJSON()` example

- `04_get_json_jquery.html`
- Let's grab some data from flickr
 - [here kitty, kitty...](#)
- And then let's display them

Let's clean that up

- If you noticed there was a delay getting cats off of flickr
- Let's make sure the user knows there are cats coming
- AJAX start and end methods

\$.ajaxStart()

- `$.ajaxStart(function() {`
- `$("div#loading").text("Loading...");`
- `});`
- register a function that set the innerHTML of div to Loading...

\$.ajaxComplete()

- `$.ajaxComplete(function() {`
- `$("div#loading").text("");`
- `});`
- Method to register a handler function which gets called by jQuery when an AJAX request get completed and any other active request are not in progress.

Handling form data

- While submitting a form using AJAX, you have to create a input string that contains the value of all the input elements on the screen.
- It is very difficult to create this string if your form is very big.
- jQuery's `serialize()` and `serializeArray()` methods handle this.

`$.serialize()`

- Serializes a set of input elements into a string of data.
- Serialize is typically used to prepare user input data to be posted to a server.
- The serialized data is in a standard format that is compatible with almost all server side programming languages and frameworks.
- In order to work properly serialize requires that form fields have a name attribute. Having only an id will not work.

`$.serialize()` example

- `05_serialize_jquery.html`
- Take some data, return it to the browser
- This could also send the data via AJAX to the server

`$.serializeArray()`

- `serializeArray()` does the similar job. Except it creates JSON output.

`$.serializeArray()` example

- `06_serializeArray_jquery.html`
- Take some data, return it to the browser
- Have to loop through it because it's JSON
- `[{name: 'FirstName', value: ''},`
- `{name: 'LastName', value: ''}]`
- This could also send the data via AJAX to the server

A full example

- I found a good overall example of jQuery and AJAX
- It's pretty simple yet covers what we need
 - Let's take a look at the demo they provide
 - Demo/



Interface just got easier
with jQuery UI

Current trends in UI libraries

- One of a few other JavaScript UI libraries:
 - [YUI](#) - Yahoo! User Interface library
 - [Prototype UI](#)
 - [Fluid UI](#)
- Let's look at some numbers:
 - [Google Trends](#)
 - jQuery UI v. (YUI, Prototype UI, Fluid UI)

jQuery UI - Overview

- Used to build highly interactive web applications
- Provides abstractions for low-level interaction and animation, advanced effects and high-level, themeable widgets
- Built on top of the jQuery JavaScript Library
- All info is at: <http://jqueryui.com/>

Integrating jQuery UI

- Since it is built on top of jQuery, you'll need to include three files on any page to use jQuery UI widgets and interactions:
 - <link rel="stylesheet" href="css/themename/jquery-ui.custom.css" />
 - <script src="js/jquery.min.js"></script>
 - <script src="js/jquery-ui.custom.min.js"></script>
- jQuery has to be before jQuery UI

... but what can it do?

- Core features
 - Animations
 - Transition Effects
 - Visibility Effects
 - Widgets
 - We previously mentioned the ‘Theme Roller’

Animation

- Basic effect transition
- Visibility transition
- Color transition
- Class transition
- Advanced easing

Basic effect transition

- Call effect on any DOM element
- `.effect(effect, [options], [speed], [callback])`
 - **effect**
 - 'blind', 'bounce', 'clip', 'drop', 'explode', 'fold', 'highlight', 'puff', 'pulsate', 'scale', 'shake', 'size', 'slide', 'transfer'.
 - **options**
 - <http://api.jqueryui.com/category/effects/>
 - **speed**
 - "slow", "normal", or "fast" or # of milliseconds
 - **callback**
 - function called after effect is completed

Visibility transition

- Show
 - jQuery show:
 - `.show([duration],[easing],[callback])`
 - jQuery UI
 - `.show(effect,[options], [speed], [callback])`
- Hide
 - `.hide(effect,[options], [speed], [callback])`
- Toggle
 - `.toggle(effect,[options], [speed], [callback])`

There are so many great demos

- <http://jqueryui.com/demos/>



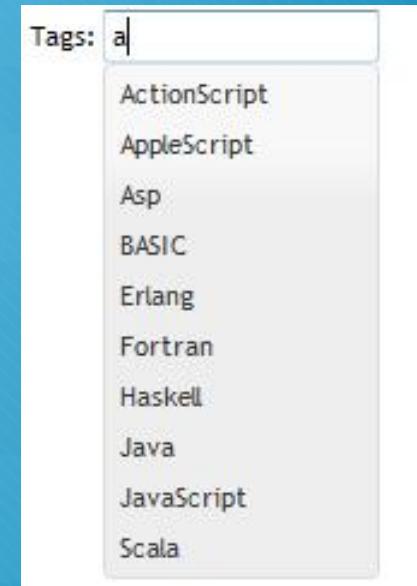
Date picker

- Would you rather do this...
- <http://answers.oreilly.com/topic/136-how-to-create-a-calendar-date-picker-with-javascript/>
- Or this... `$('#date').datepicker();`
- `<input type="text" name="date" id="date" />`



Autocomplete

- Would you rather do this:
<http://www.webreference.com/programming/javascript/gr/column5/index.html>
- Or this? <http://jqueryui.com/autocomplete/>
 - `var availableTags = ["ActionScript", "AppleScript", "Asp", "BASIC", "C", "C++"];`
 - `$("#tags").autocomplete({ source: availableTags });`
 - `<input type="text" id="tags" />`



Theme Roller

- Create themed widget for each jQuery UI widget
 - <http://jqueryui.com/themeroller/>
- Makes it easy to create widgets follow a color scheme



How would we approach this?

- I have this little gallery I built a while back
- It was built using standard DOM manipulation
- Let's fix that and make it better!

Conclusion

- **jQuery really simplifies AJAX!**
- **Many of the functions that were cumbersome in JavaScript manually are one line of code now.**
- **jQuery UI really simplifies creating user interface elements!**
- **There are so many options that solve lines and lines of code in a single line of jQuery UI**

What now?

- Practice, practice practice!
- Try some of the examples from the jQuery UI site and integrate them where possible to make a gallery of your own!
- Take your first assignment and see how fast you can redo it with jQuery AJAX and jQuery UI methods
- And go!

Additional Reading

- <http://jqueryui.com/>
- <http://api.jqueryui.com/category/effects/>
- <http://jqueryui.com/themeroller/>
- <http://www.jquery.com>
- http://www.w3schools.com/jquery/jquery_ref_ajax.asp