

Javascript, part 2

CS147L Lecture 5
Mike Krieger

Intro

Welcome back!

By the end of today...

- Getting stuff onto your Stanford space
- What is AJAX?
- How to make asynchronous Web requests
- Using jQTouch to set up touch events
- Learn animations with jQTouch

Javascript Recap

Arrays: [1,2,3,"hi"]

Objects: {"propertyName": "value"}

Functions:

```
function fn(args){  
    doStuff();  
}
```

```
var fn = function() { alert("HI"); }  
fn();
```

jQuery

```
$("#hello"); //id:hello  
$(".hello"); //class:hello  
$("div"); //all divs  
$("div", "#main"); // all divs inside main  
$("div").each(function(){  
    $(this).doSomething();  
}); // act on every div  
$("#hello").css("background-color", "black");
```

Getting stuff onto the
Stanford space

FTP client: What you'll need

- On the Mac
 - Cyberduck
- On Windows
 - SecureFX (<http://ess.stanford.edu>)

Then...

- Sign on to **cardinal.stanford.edu**
- Transmit files to WWW/ folder
- Important: request CGI setup at **<http://www.stanford.edu/services/web/cgi/personal.html>**

Then...

- files will be available at:

<http://www.stanford.edu/~yoursunetid>

- next week, we'll write server scripts that go into

<http://www.stanford.edu/~yoursunetid/cgi-bin>

Setting up Google Code

Google Code

- Free SVN hosting!
- [http://code.google.com/hosting/
createProject](http://code.google.com/hosting/createProject)

After creating your app

Project members authenticate over HTTPS to allow committing changes.

```
svn checkout https://projectname.googlecode.com/svn/trunk/  
projectname --username yourusername
```

In Tortoise, just right click a folder and choose "Checkout...", enter the information above

After making changes

```
>> svn status  
? stuff.js  
M main.html
```

(means: you have a file called `stuff.js` that's not in your SVN repository, and a file called `main.html` which is, & has changes)

```
>> svn add stuff.js  
A stuff.js
```

(means: on next commit, `stuff.js` will be added to the SVN repository)

```
>> svn commit -m "adding stuff.js, refactoring main.html"  
A stuff.js  
M main.html
```

(sends the changes to the repository, `-m` passes message)

Then, your team-mates

```
>> svn up  
A stuff.js  
U main.html
```

(means: SVN added `stuff.js` to your local copy, and **U** updated `main.html`)

Network & JS

This section

- GET vs POST
- Asynchronous vs Synchronous
- XML vs JSON

Basics: GET & POST

- GET (<http://example.com/index.html?hello=yes&goodbye=no>)
 - No message body
 - Query is in the URL
 - Shouldn't change data on server
 - Carries context in it, can give to others
- POST (<http://example.com/index.py>)
 - Message body has parameters
 - Can have side-effects, change data on server
 - Can carry more data
 - This is an *action*

In other words...

- *fetch* using GET
- *change* using POST

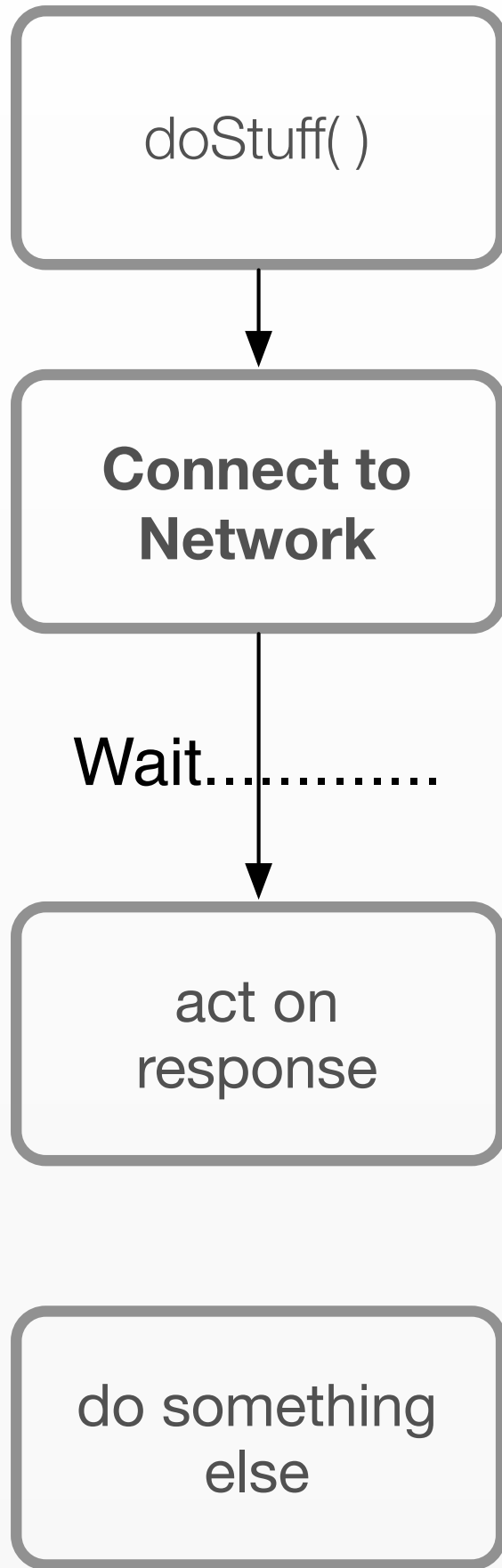
Okay, so:

Asynchronous
Javascript
And
XML

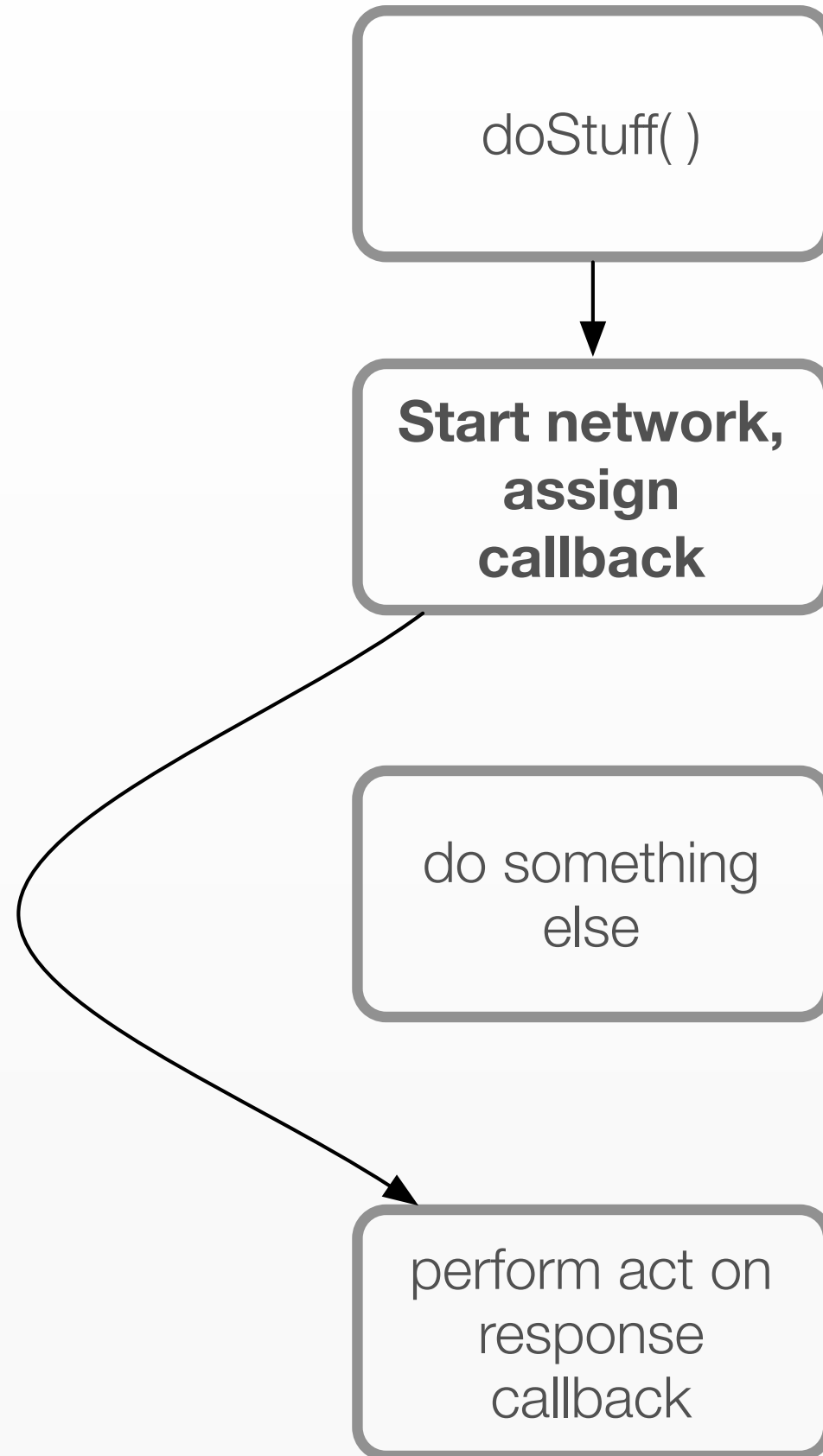
What does asynchronous mean?

- In context of network connections: let other stuff happen while a network request is going on

Synchronous



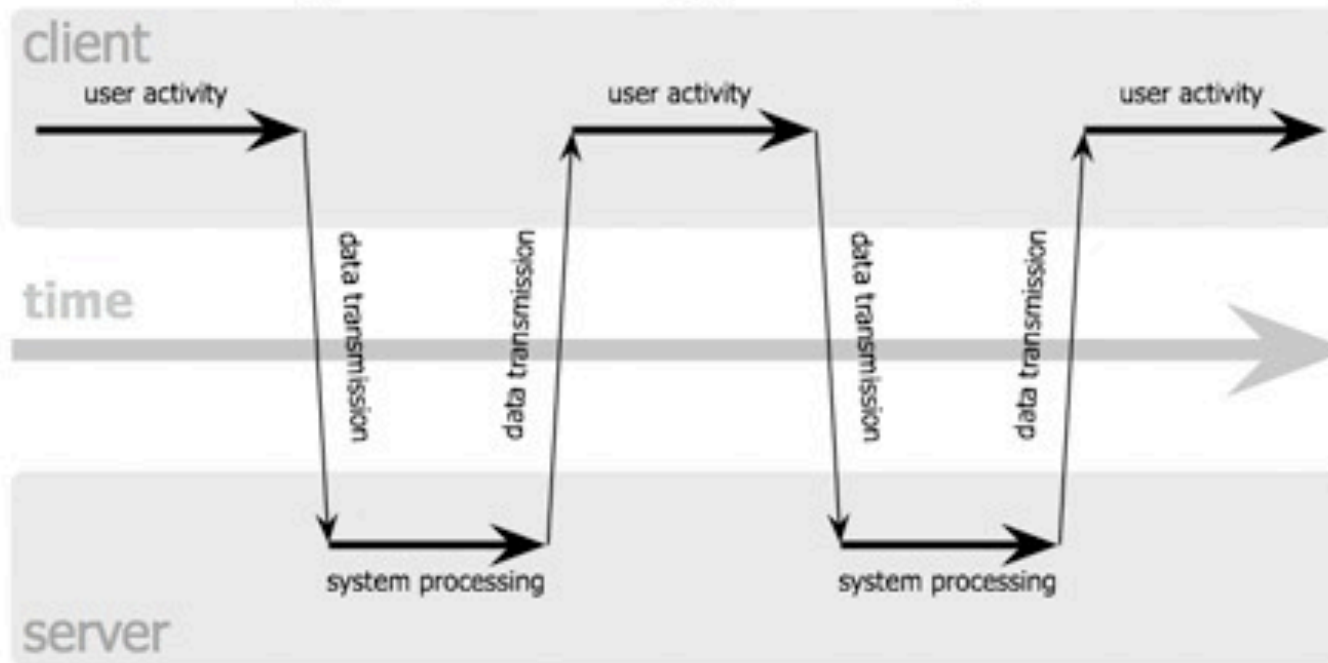
Asynchronous



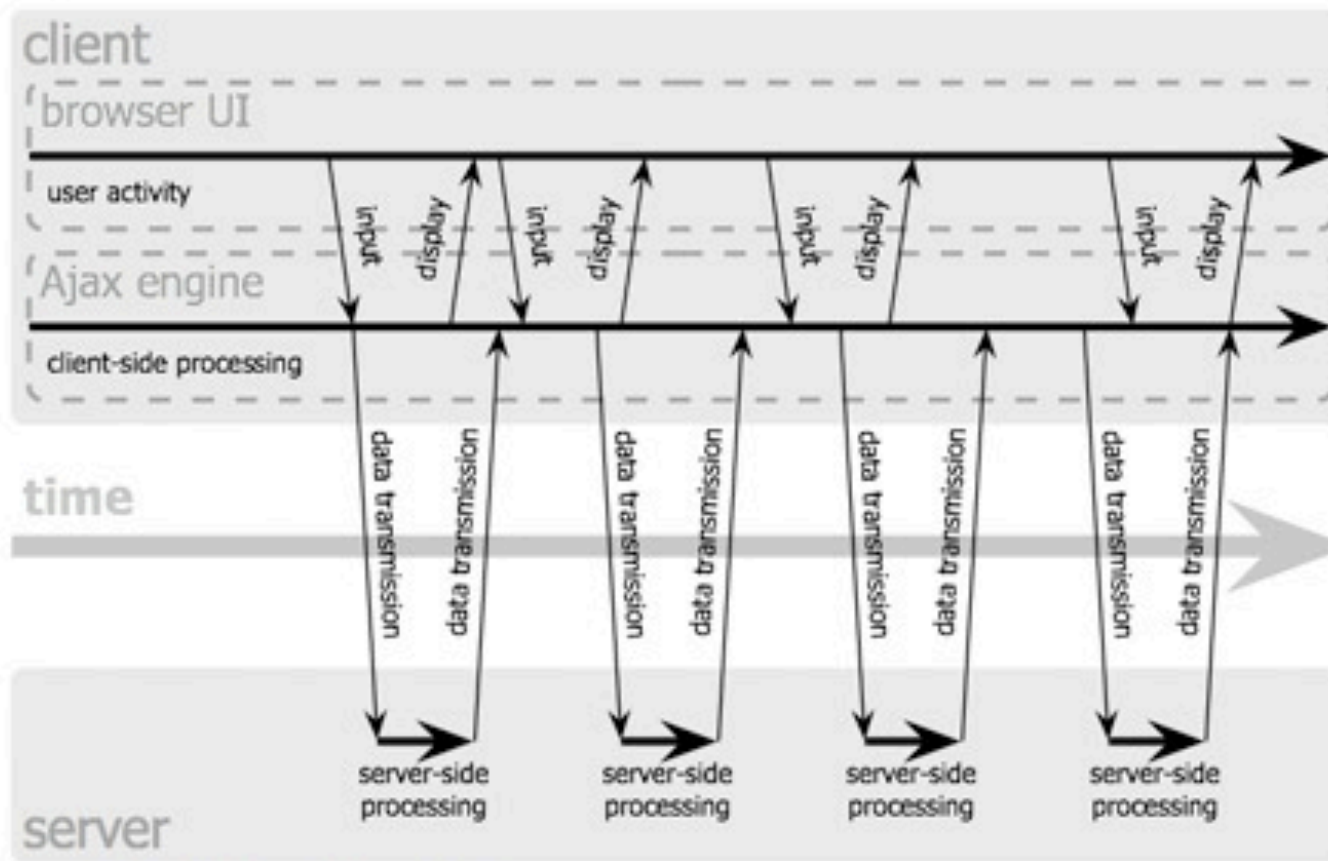
Brief history of asynchronous Web

- Microsoft introduced XMLHttpRequest for Outlook Web Access
- Term “AJAX” coined by Jesse James Garrett after seeing Google Maps & Google Suggest
- Now ubiquitous

classic web application model (synchronous)



Ajax web application model (asynchronous)



Jesse James Garrett / adrianath.com

XML?

- Many people use JSON instead of XML
- We'll do JSON, easier to build/parse than XML

What is JSON?

- **J**ava**S**cript **O**bject **N**otation
- Takes the array, object, and string notation from JS & makes it a transport format (can send around a string that evaluates to JSON)

Sample JSON

```
{
  classes: [
    {
      "name": "CS147",
      "location": "Gates 101"
    },
    {
      "name": "ME101",
      "location": "Design loft"
    }
  ]
}
// this is an object with an array inside with
two objects
```

JSON.parse()

- Takes JSON data, encoded in a string
- Returns an object you can use in JS

Example

```
>> var a = "[5,4,3,2,1,\"hello\"]"  
>> var b = JSON.parse(a);  
>> a.length  
6  
>> a[5]  
"hello"
```

JSON.stringify

- Takes a Javascript Array, Object, or String, and turns it into JSON

Example

```
var a = {"dog": "bark", "cat": "meow"};
```

```
>> JSON.stringify(a);
```

```
"{\ \"dog\ ": \"bark\ ", \ \"cat\ ": \"meow\ "}"
```

Asynchronous Requests in jQuery

Available functions

- overall function: `.ajax()`
- Shortcuts:
 - `.get`, `.post`, `.getJSON`

\$.ajax

```
var options = {  
  data: {"animals":["dog", "cat", "cow"]},  
  dataType: "json",  
  error: onError,  
  success: successFn,  
  url: "/postAnimal.py"  
};  
  
$.ajax(options);
```

Success & Failure

```
/*response:
  {'dog': 'bark', 'cat': 'meow', 'cow': 'moo'}
*/

function onSuccess(data) {
  $.each(data, function(i, item) {
    $("

" + i + " says " + item + "</div>").appendTo("#animals");
  })
}

function onError(function(request, settings) {
  $("#status").append("<div>Error fetching " + settings.url + "</div>")
})


```

Shortcuts

```
$.get(url, data, successCallback);  
// note, no error callback  
// grabs HTML or XML
```

```
$.getJSON(url, data, successCallback);  
// also no error callback  
// but grabs & parses JSON
```

Cross-site issues

- You can only **GET** and **POST** from same domain
- <http://stanford.edu> <-> stanford.edu
OK, <http://stanford.edu> <->
google.com Not OK
- What if you need to **GET** from a different domain?

JSONP

- If server supports JSONP, pass in a callback that gets wrapped around the response
- Data gets loaded using a `<script>` tag, which doesn't have cross-site restrictions

JSONP Example

<http://example.com/getJSON.php>

Response:

```
{"cat": "meow", "dog": "bark"}
```

<http://example.com/getJSON.php?callback=processAnimals>

Response:

```
processAnimals({"cat": "meow", "dog": "bark"})
```

JSONP in jQuery

- Add a "callback=?", jQuery will call the callbackFunction you pass in as the third parameter
- You can even define an anonymous function in the successFunction slot and jQuery will call that

JSONP in jQuery

```
function processAnimals (data) {  
    $.each(data, function(i, item) {  
        $("<div>" + i + " says " + item + "</div>").appendTo  
        ("#animals");  
    }  
}  
  
$.getJSON("http://example.com/getJSON.php?callback=?",  
    {'animals': ['dog', 'cat']}, processAnimals);
```

JSONP in jQuery

```
$.getJSON("http://example.com/getJSON.php?callback=?", {'animals':  
  ['dog', 'cat']},  
  function(data) {  
    $.each(data, function(i, item) {  
      $("<div>" + i + ": " + item + "</div>").appendTo("#animals");  
    }  
  });
```

Demo

ajax.html

A simple form

```
<form id="mainform" method="POST" action="server.py">
...
  <td>Your name</td><td><input type="text" name="name" /></td>
...
  <td>E-mail</td><td><input type="text" name="email" /></td>
...
  <td colspan="2"><input type="submit" value="send form" /></td>
...
</form>
```

Form submission in \$

```
$("#mainform").submit(function(event){
    var keyvals = {};
    $("input[type!=submit]", "#mainform").each(function(i,el){
        el = $(el);
        keyvals[el.attr("name")] = el.val();
    })
    $.post($(this).attr("action"), keyvals, function(response){
        $("#response").html(response);
    })
    return false; //prevents normal submit event from firing
})
```

Meanwhile, server-side...

- We'll cover the server-side next week

jQueryTouch

Next few sections

- jQuery Touch Basics & Workflow
- Changing list input style
- Overriding styles
- Setting up animations
- Swipes & Touches

jQuery Touch

- You don't have to use it, but it provides some convenient functions
- Include it after jQuery gets loaded
- There's also a “jQuery Touch”, that's a different project
- **Beta software**

Demo

`jqtouch-basic.html`

jQTouch Basics

```
<head>
```

```
  <style type="text/css" media="screen">@import "../jqt/jqtouch.css";</style>
  <style type="text/css" media="screen">@import "../jqt/theme.css";</style>
  <script src="../jquery.js" type="text/javascript" charset="utf-8"></script>
  <script src="../jqt/jqtouch.js" type="text/javascript" charset="utf-8"></script>
  <script type="text/javascript" charset="utf-8">
    var jQT = new $.jQTouch();
  </script>
```

```
</head>
```

jQTouch HTML

```
<div id="example" class="current">
  <ul class="rounded">
    <li><a href="#one">Page 1</a>
    <li><a href="#two">Page 2</a>
  </ul>
</div>
<div id="one">
  <div class="toolbar">
    <h1>Page 1</h1><a class="back">Back</a>
  </div>
  This is page one.
</div>
<div id="two">
  <ul><li>This is another page. <a href="#three">go to page 3?</a></li></ul>
</div>
<div id="three">
  <div class="toolbar">
    <h1>Page 3</h1><a class="back">Back</a>
  </div>
  This is page 3.
  <ul>
    <li><a href="http://cs147.stanford.edu/" target="_blank" rel="external">CS147
web</a></li>
  </ul>
</div>
```

Workflow

- Add divs to your page with meaningful IDs
- Set up links to those ids with anchors inside **LIs**:

```
<ul><li><a href="#yourid">your link</a></li></ul>
```

- jQTouch handles the animations
- target=_blank on external links

Content can be fetched asynchronously

- Just specify a URL instead of a hash

Loading asynch

```
<div id="example" class="current">
  <ul class="rounded">
    <li><a href="#one">Page 1</a>
    <li><a href="jqtouch-loadme.html">Page 2</a>
  </ul>
</div>
<div id="one">
  This is page one.
</div>
```

In a different file...

```
<div id="two">
  <ul>
    <li>This is another page. <a href="#three">go to page 3?</a></li>
  </ul>
</div>
```


Demo

`jqtouch-asyncload.html`

Webapp features

Demo: `jqtouch-app.html`

Configuring webapp

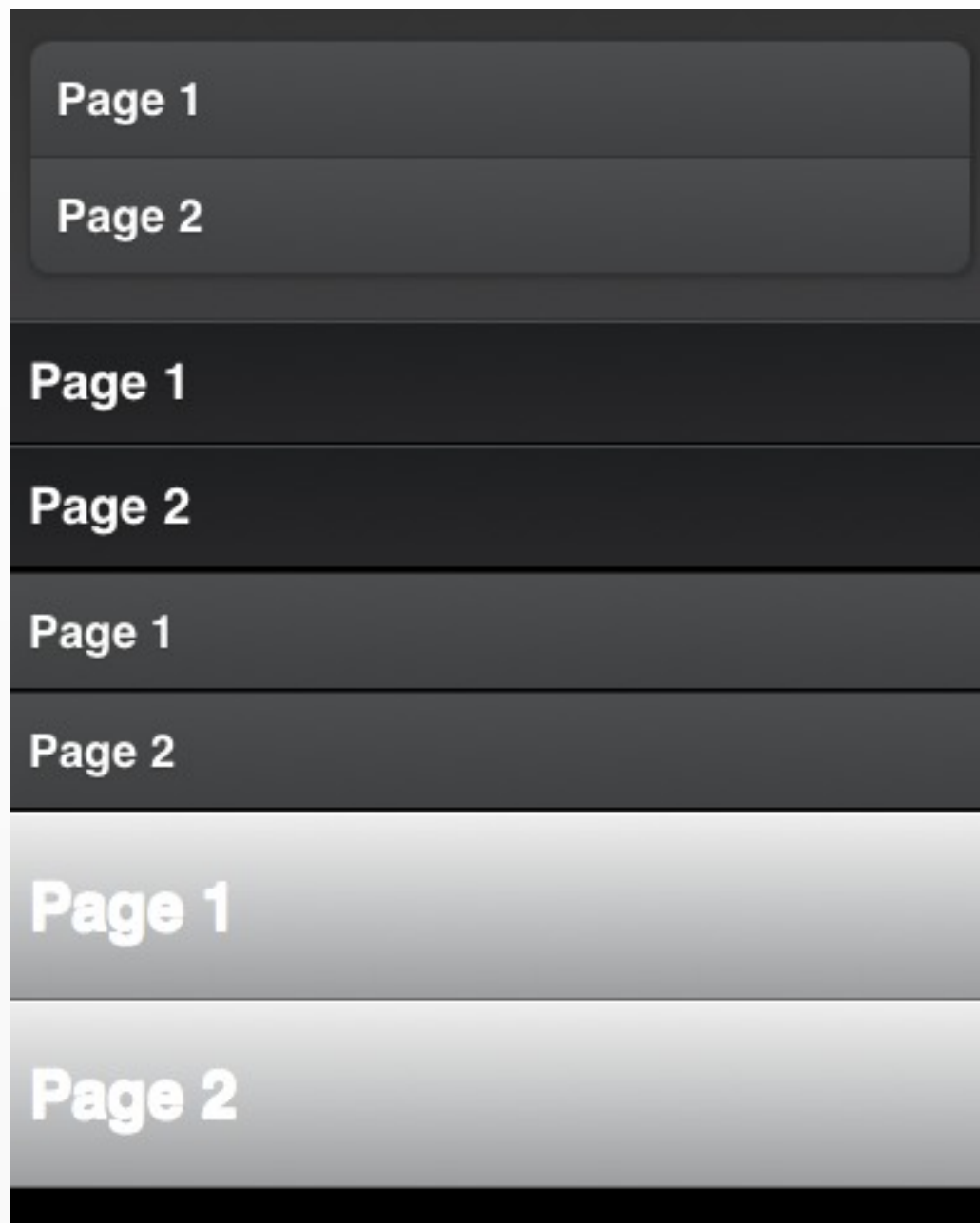
```
var jQT = new $.jQTouch({
  statusBar: 'black',
  icon: 'cs147.png',
  addGlossToIcon: true, // defaults to true
  startUpScreen: 'cs147-start.png'
});
```

List Types

Specifying list type

- Change class of ``
- Default theme options: rounded, edgetoedge, plastic, metal

Examples



Rounded

edgetoedge

plastic

metal

Changing Styles

Styling

- Don't want your app to look like all apps in the class

Workflow

- Include basic jqtouch.css
- Optionally include theme.css
- Add your own styles by overwriting
- Use Safari Web Inspector to check what styles are active

Page 1


Page 2

Open Link in New Window
Open Link in New Tab

Save Linked File to "Downloads"
Save Linked File As...
Add Link to Bookmarks...

Copy Link

Inspect Element

1Password 

The image shows a browser's developer tools interface. At the top, a breadcrumb trail indicates the current element path: `current > ul.rounded > li > a`. The `li` element is selected. The main pane displays the following CSS rules:

- `border-top-right-radius: 8px 8px;`
- ul li** (from `theme.css`):
 - `background: -webkit-gradient(line...`
 - `border-bottom: 2px solid #555858;`
 - `border-top: 1px solid #333;`
 - `color: #666;`
 - `list-style-type: none;`
 - `overflow: hidden;`
 - `padding: 10px 10px 10px 10px;`
- *** (from `jqtouch.css`):
 - `margin: 0px;`
 - `padding: 0px;`

Some style

```
<style type="text/css" media="screen">
  body > * {
    background: -webkit-gradient(linear, 0% 0, 0% 100%, from(rgb(144, 124,
250)), to(rgb(44, 104, 151)))
  }
  ul li {
    background: -webkit-gradient(linear, 0% 0, 0% 100%, from(rgb(44, 94,
250)), to(rgb(70, 94, 101)))
  }
  ul li .active {
    background: -webkit-gradient(linear, 0% 0, 0% 100%, from(rgb(44, 94,
250)), to(rgb(70, 94, 250)))

  }
</style>
```

Page 1

Page 2

Notes

- Some things are images; you'll see background-image (might need to make your own)
- Great chance to use all of those WebKit styles we covered!

Changing animations

Demo

- jqTouch Demo App in iPhone Simulator

Specifying animation

- Specify class on your `<a>` link

Available Animations

slide
slideup
dissolve
fade
flip
pop
swap
cube

Example code

```
<div id="example">  
  <ul class="rounded">  
    <li><a class="flip" href="#one">Page 1</a>  
    <li><a class="swap" href="#two">Page 2</a>  
  </ul>  
</div>
```

Demo

`jqtouch-animation.html`

Notes

- You can also write your own animations, like we covered

Adding custom anim

```
<script>
  var jQT = new $.jQTouch();

  $(function(){
    jQT.addAnimation({
      name: 'reveal',
      selector: '.revealme'
    });
  });
</script>
// from jQTouch docs
```

Reveal's anim code

```
<style>
  .reveal.in {
    -webkit-animation-name: dontmove;
    z-index: 0;
  }
  .reveal.out {
    -webkit-animation-name: revealout;
    z-index: 10;
  }
  .reveal.out.reverse {
    z-index: 0;
    -webkit-animation-name: dontmove;
  }
  .reveal.in.reverse {
    z-index: 10;
    -webkit-animation-name: revealin;
  }

  @-webkit-keyframes revealin {
    from { -webkit-transform: translateX(100%); }
    to { -webkit-transform: translateX(0); }
  }

  @-webkit-keyframes revealout {
    from { -webkit-transform: translateX(0); }
    to { -webkit-transform: translateX(100%); }
  }
</style>
```

Detecting Touches

Touch and hover

- jQuery will set an “active” class on touch&hover for anything with a “.touch” class
- Will take off the active class if the touch ends up not being a tap (if you move your finger)
- And if you swipe sideways, will fire .swipe

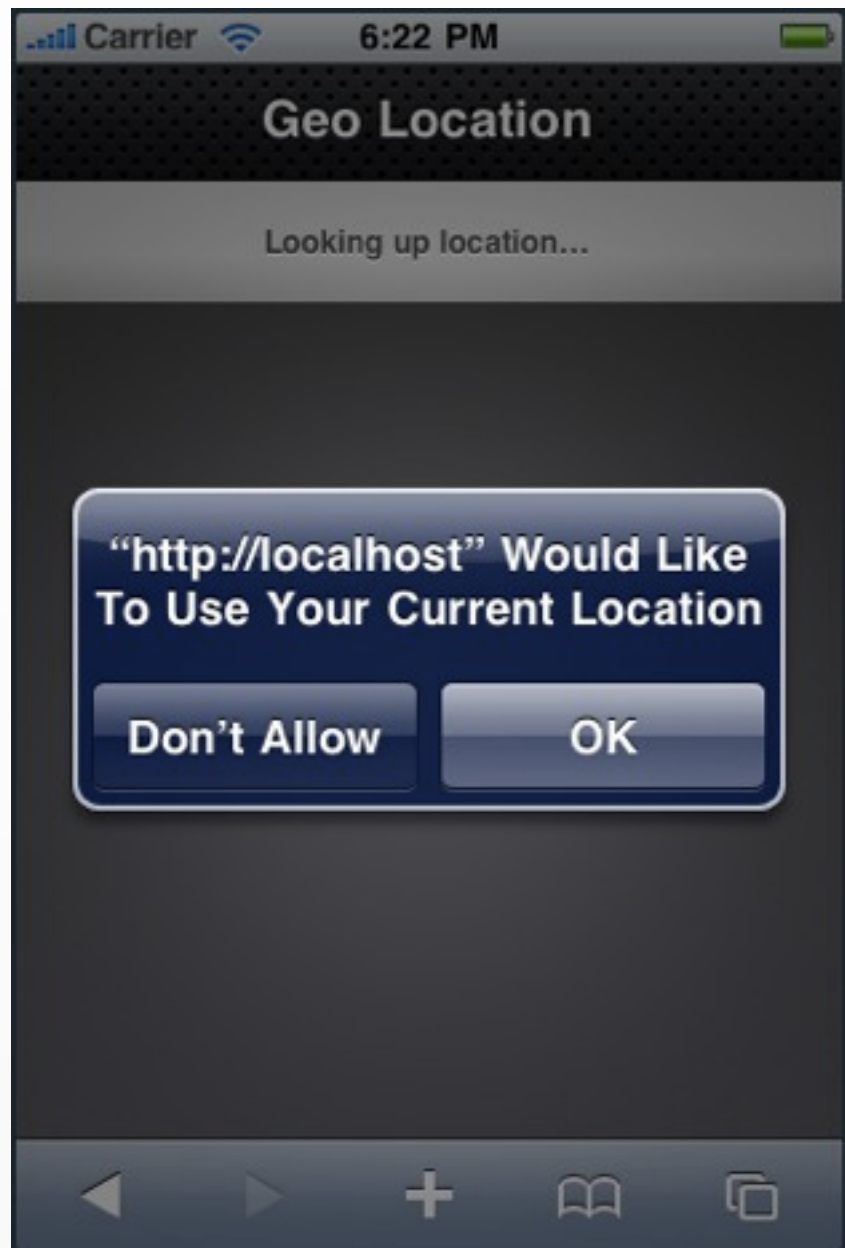
Demo

`jqtouch-touchables.html`

Source

```
<script type="text/javascript" charset="utf-8">
  var jQT = new $.jQTouch({});
  $(document).ready(function(){
    $(".touch").swipe(function(){
      alert("SWIPE");
    })
    $(".touch").bind("touchstart", function(){
      $(".touch").unselect();
    })
  })
</script>
...
<div id="example">
  <div class="touch cta">Touch this.</div>
  <div class="touch cta">Or touch that.</div>
</div>
```

Location, location



Geo Location

Latitude: 37.331689
Longitude: -122.030731

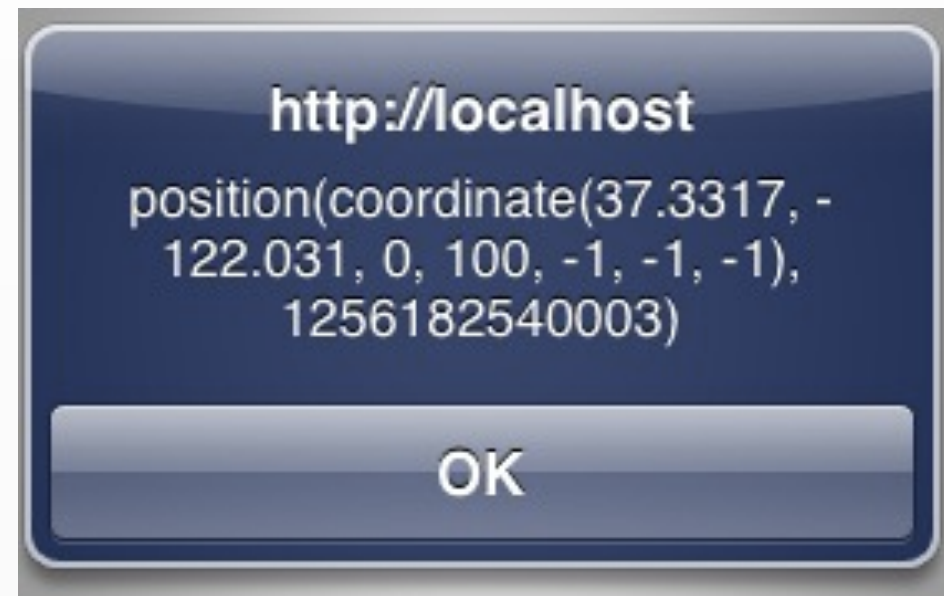
Geolocation API

- navigator.geolocation.getCurrentPosition
(callback)

Sample code

```
navigator.geolocation.getCurrentPosition(handlePosition)

function handlePosition(position) {
    $("#lat").html(position.coords.latitude);
    $("#lon").html(position.coords.longitude);
}
```

Reverse Geocoding

- idea: take latitude & longitude and get “real world location”
- problem: we need to make an AJAX request to tinygeocoder.com (different domain)
- solution: `getJSON!`

Demo

geocode.html

A simple body

```
<body>  
  <div id="status"></div>  
  <div id="result"></div>  
</body>
```

Get the location

```
$(document).ready(function(){
    $("#status").html("Finding location...");
    if(navigator.geolocation) {
        navigator.geolocation.getCurrentPosition(positionFound);
    } else {
        // fake it
        window.setTimeout(function(){
            positionFound({'coords':{'latitude':37.428337,
'longitude':-122.175822}})
        }, 2500);
    }
})
```

On position found...

```
function positionFound(position) {
    $("#status").html("Location found; reverse geocoding...");
    var lat = position.coords.latitude;
    var lon = position.coords.longitude;
    $.getJSON("http://tinygeocoder.com/create-api.php?
callback=?", {'g': lat+', '+lon}, function(data){
        $("#status").hide();
        $("#result").html("" + data);
    })
}
```

Orientation

Orientation

Getting orientation change

```
// event name: orientationchange
// using jQuery:

$("body").bind("orientationchange", function(event) {
    react()
})
```

Example app

orientation.html

The Code

```
function setOrientation() {
    var orientation = (window.innerWidth <
window.innerHeight) ? 'portrait' : 'landscape';
    $("#orientation").html(orientation);
}
$(document).ready(function(){
    setOrientation();
    $("body").bind("orientationchange", function(event){
        setOrientation();
    })
})
```

jQTouch Shortcut

```
// "turn" event  
$("body").turn(function(event, data) {  
    console.log(data.orientation);  
})
```

Putting it together: Portfolio 5

This week

- Try a few animations in the menu
- Locate people with geolocation
- Use jQTouch to make our lives easier

Demo

- week05.html in portfolio folder

Home screen

```
<div id="home-screen" class="current content">
  <div id="info">
    <div id='fullname'>Joe <span id='last-name'>Student</span></div>
    <div id="tagline">joe@student.com<br/>Interface Designer</div>
  </div>
  <div class="clear"></div>
  <ul>
    <li><a href="#bio" class="flip" id="bio-link">See bio</a></li>
  </ul>
  <h2>Course work</h2>
  <ul >
    <li><a href="#howmightwe">How Might We?</a></li>
    <li><a class="dissolve" href="#inspiration">Inspiration</a></li>
    <li><a class="cube" href="#discovery">Discovery</a></li>
  </ul>

  <!-- we need this for our float layout -->
  <div class='clear'></div>
</div>
```


Other sections

```
<div id="howmightwe" class='content'>
```

```
    This is some great work I did for the How Might We? Assignment.
```

```
</div>
```

```
<div id="inspiration" class='content'>
```

```
    Wow, that was super inspirational.
```

```
</div>
```

```
<div id="discovery" class='content'>
```

```
    Can you discover?
```

```
</div>
```

Bio section

```
<div id='bio'>
  <div class='content'>
    <h2>Bio</h2>
    <p>I love talking about places!
    <p>Let me see if I can find where you are.
    <div id="status"></div>
    <div id="result"></div>
    <div id="know"></div>
    <ul>
      <li><a href="#" class="goback">Back home</a></li>
    </ul>
  </div>
</div>
```

document.ready

```
$(document).ready(function(){
    $("#bio").bind("pageAnimationEnd",function(e){
        $("#status", "#bio").html("Finding location...");
        if(navigator.geolocation) {
            navigator.geolocation.getCurrentPosition(positionFound);
        } else {
            // fake it
            window.setTimeout(function(){
                positionFound({'coords':{'latitude':37.428337,
'longitude':-122.175822}})
            }, 2500);
        }
    })
})
```

positionFound

```
function positionFound(position) {
    $("#status", "#bio").html("Location found; reverse geocoding...");
    var lat = position.coords.latitude;
    var lon = position.coords.longitude;
    $.getJSON("http://tinygeocoder.com/create-api.php?callback=?", {'g':
lat+', '+lon}, function(data){
        $("#status", "#bio").html("It looks like you're at:");
        $("#result", "#bio").html("" + data);
        var regexp = /Palo Alto|Stanford|San Francisco/gi;
        if (String(data).match(regexp)) {
            $("#know", "#bio").html("I know a lot about this place, you
should ask me about it.");
        } else {
            $("#know", "#bio").html("I've never been there, why don't you
tell me about it?");
        }
    })
}
```

Q's?