Web Development

COMP 431 / COMP 531

JavaScript Libraries

Scott E Pollack, PhD

February 9, 2016

Recap

- HTML and HTML5
- JavaScript
- Forms
- CSS and Style Frameworks
- Events
- Homework Assignment 3 (Draft Front-End)
  - **Due TONIGHT**

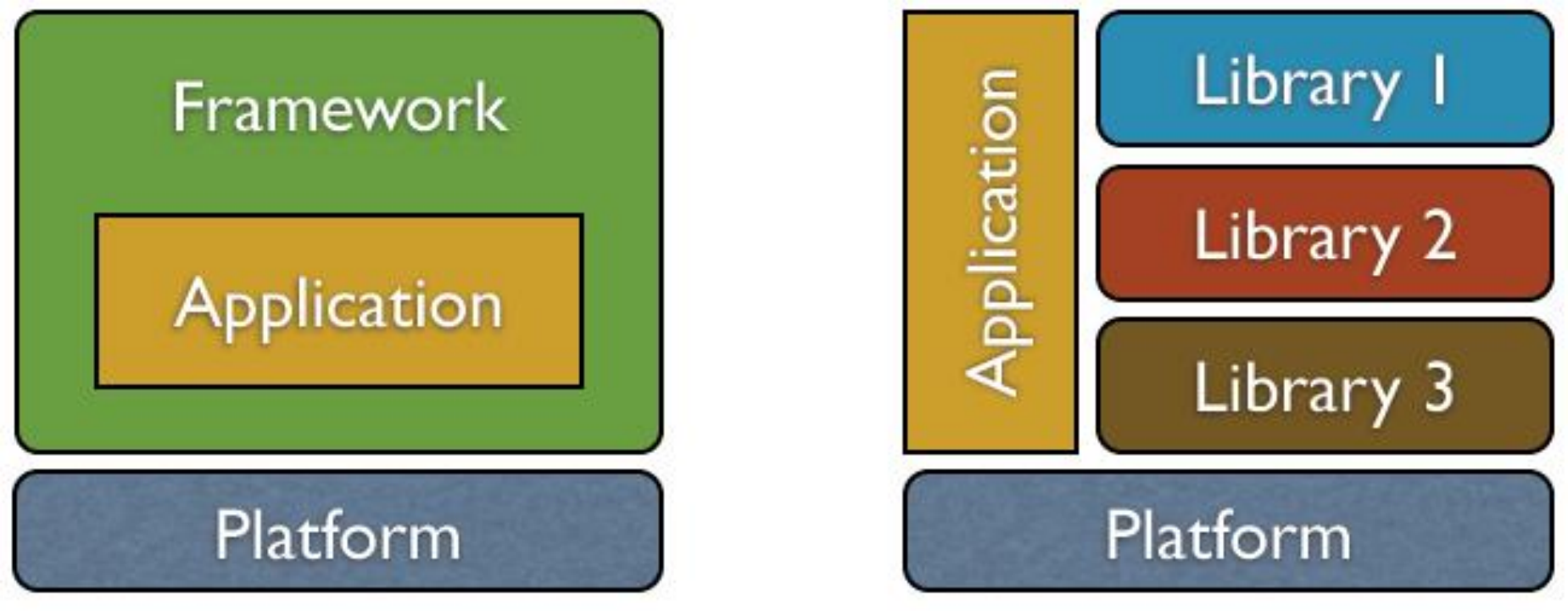
~~Style (Bootstrap)~~
Libraries (jQuery)
MVC
AngularJS

Homework Assignment 4 (JavaScript Game)
Due Thursday 2/18

Any questions?

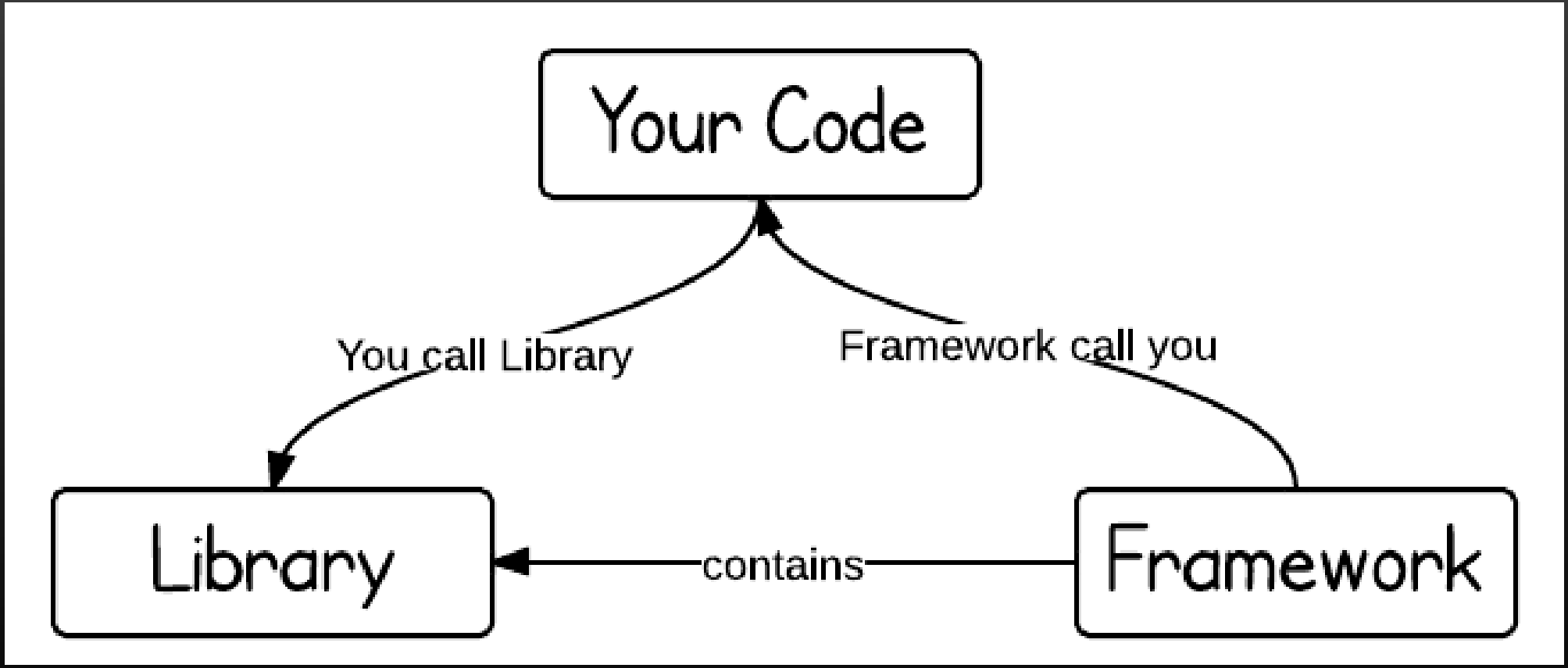
Homework Assignment 4
(JavaScript Game)
Due Thursday 2/18

Libraries vs Frameworks



http://tom.lokhorst.eu/2010/09/why-libraries-are-better-than-frameworks

Inversion of Control



Library or Framework?

- Bootstrap

FRAMEWORK

- jQuery

Library

- AngularJS

FRAMEWORK

- jQuery (DOM manipulation)
- Underscore || Lodash (helpers)
- Moment.js (time)
- RequireJS (infrastructure)
- ReactJS (view by Facebook)
- Oboe.js (Streaming AJAX)
- ~~YUI (Yahoo User Interface)~~
- Raphael (SVG)
- EaselJS, Fabric.js, Paper.js (canvas)
- D3.js (SVG)
- Velocity.js (SVG)
- Three.js (WebGL)

microjs·com

jQuery JavaScript Library

- Wrapper around DOM manipulation
- Instead of

```
window.onload = function() {
    document.getElementById("div1").on
        this.style.display = 'none'
    }

    document.getElementById("div2").onclick = function() {
        document.getElementById("div1").style.display = 'block'
    }

}
```

```
$(window).load(function() {

    $("#div1").click(function() {
        $("#div1").hide()
    })

    $("#div2").click(function() {
        $("#div1").show()
    })

})
```

jQuery uses CSS Selectors

- In fact, jQuery came first…

```
<div id="div1" style="background-color:blue;">
Click to Hide
</div>

<div id="div2" style="background-color:red;">
Click to Show
</div>
```

Selector Rules (the easy ones)

- Tag

```
body {
    background-color: #FFFFFF;
}
```

- Class

```
.linkInverted {
    color: #FFFF00;
}
```

- Id

```
#riceLogo {
    width: 6em;
    margin-top:-1em;
    margin-bottom:-1em;
}
```

- Attribute

```
[name="fancy"] {
    font-size: 2em;
}
```

```
$(window).load(function() {

    $("#div1").click(function() {
        $("#div1").hide()
    })

    $("#div2").click(function() {
        $("#div1").show()
    })

})
```

jQuery Manipulation

- $(…) returns a jQuery object or collection that is easier to manipulate than a DOM HTTP object.

- We just saw hide() and show() for updating the display style
- addClass(), hasClass(), removeClass(), toggleClass()
- parent(), siblings() → parents('div').last().siblings().children().andSelf()
- insertBefore(), wrap(), attr(), position()

- $(..).get() = DOM element, useful for some things still

```
> $('#div1').css('width')
<· "480px"
> $('#div1').css('width', '20px')
<· [
    <div id="div1" style="display: block; width: 20px;
    background-color: blue;">
    Click to Hide
    </div>
  ]
> $('#div1').css(['width', 'height'])
<· Object {width: "20px", height: "111px"}
> $('#div1').html('Some content')
<· [
    <div id="div1" style="display: block; width: 20px;
    background-color: blue;">Some content</div>
  ]
```

jQuery Events

- Naming ala Level 2 DOM Events, onclick → click

Regular DOM Event object

```
$("#div1").click(function(evt) {
    $(evt.target).hide()
})
```

- Initial visit event ordering

```
$(document).ready(function() {
    alert('Document Ready')
})

$(window).load(function() {
    alert('Window loaded')
```

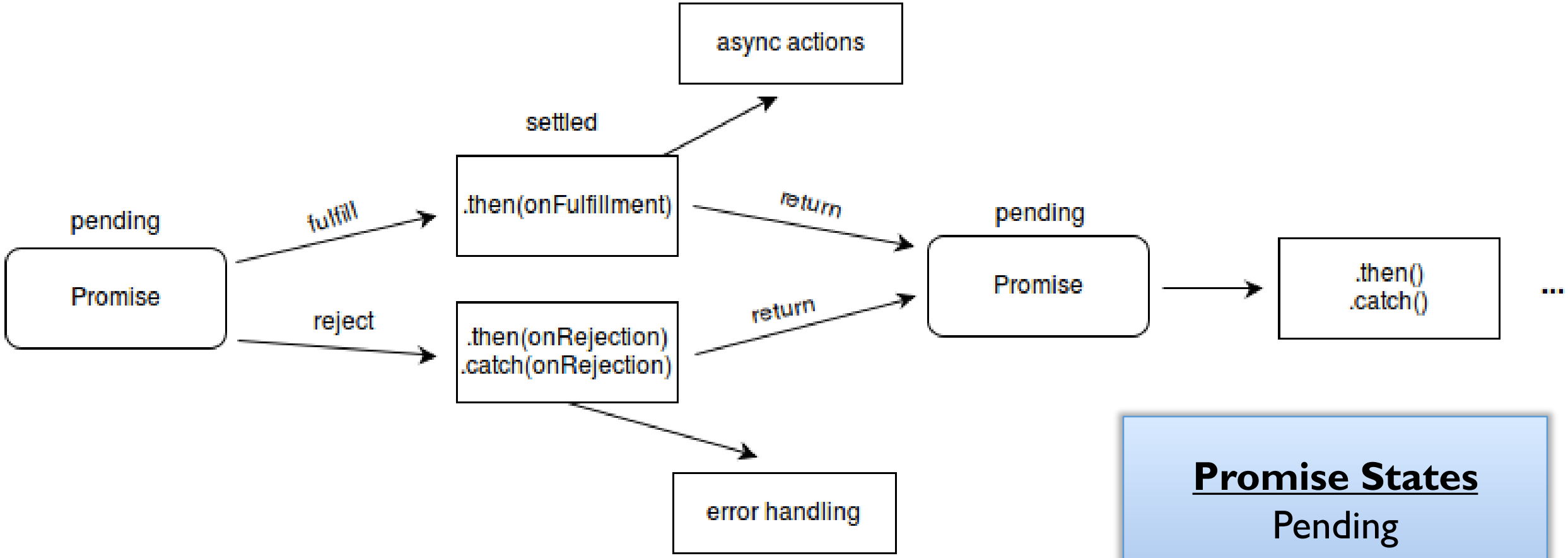
jQuery Effects

- slideUp()
- hide(2000)
- fadeOut()
- delay()
- why hide and show when you can toggle() ?

jQuery Callback vs Chaining

```
$("#div3").click(function() {
    $(this)
        .animate( {opacity: 0}, 2500)
        .animate( {opacity: 1, fontSize: '1em' }, 500 )
        .hide(1000, function() {
            $(this).css({ backgroundColor: "blue" })
        })
        .show(1000)
        .animate( { fontSize: '2em' }, function() {
            $(this).css({ backgroundColor: "green" })
        } )
})
```

Better than Callbacks => Promises



Promise States

Pending

Fulfilled / Rejected

Settled

https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Promise

See also http://www.html5rocks.com/en/tutorials/es6/promises/

Communicating with the Server

- GET
- POST
- …

- Always instantiated by the browser (i.e., user) perhaps using a link or button (e.g., to submit a form)

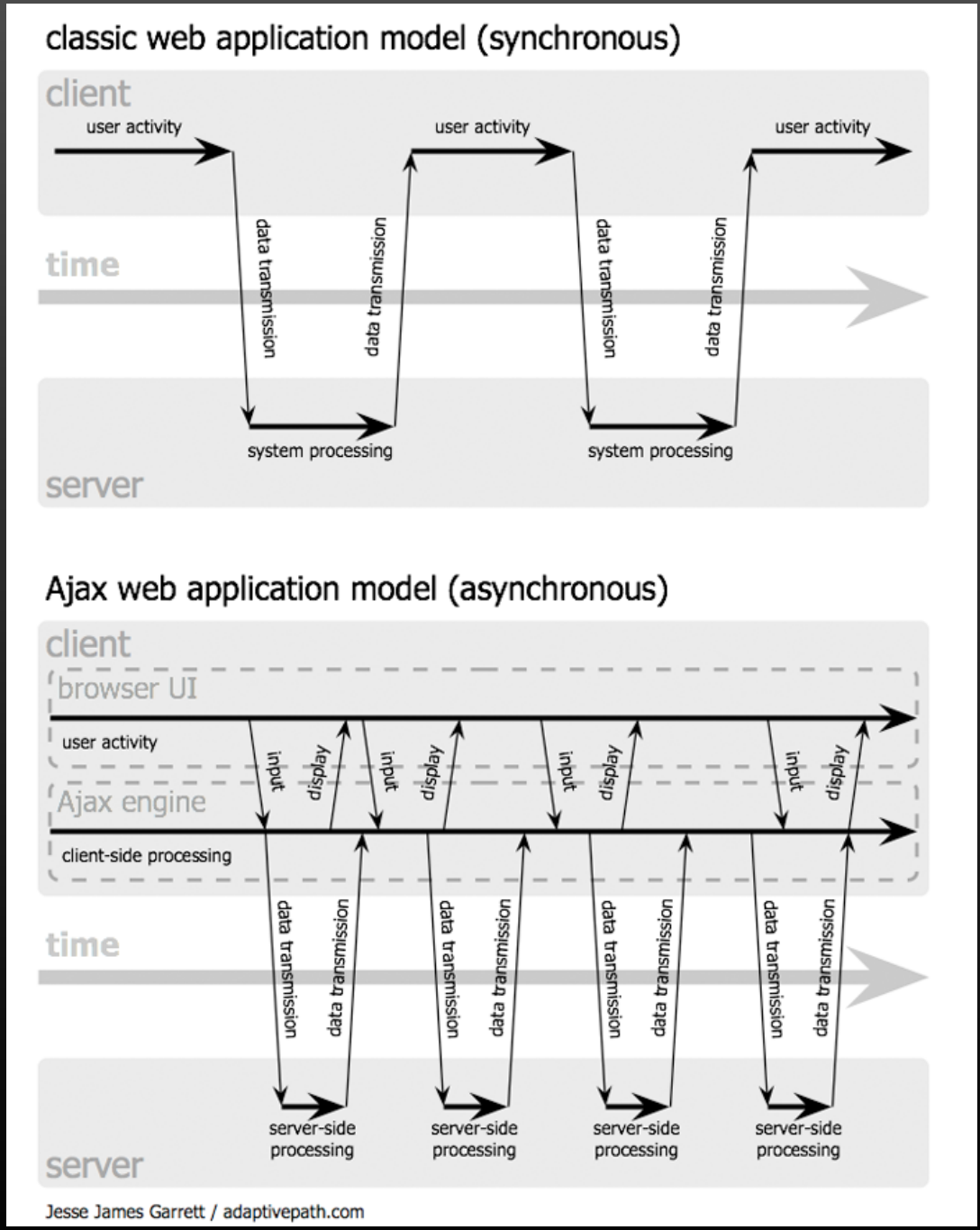
- We'd like to have JavaScript control to ask the Server for data

JavaScript Requests

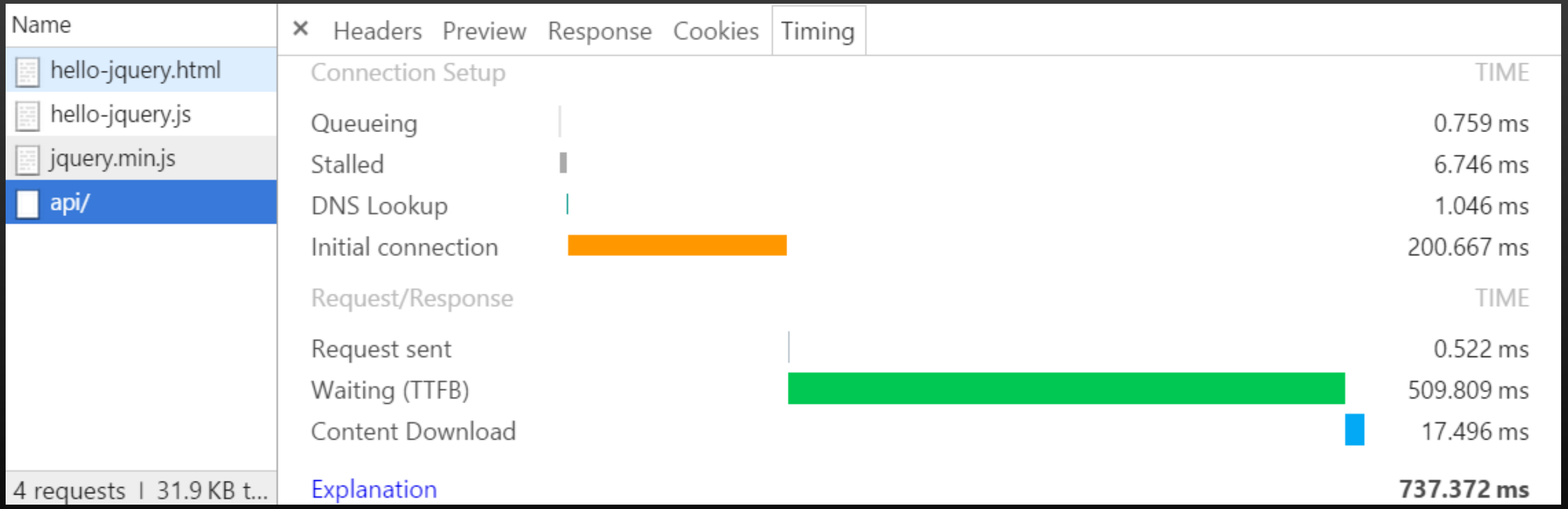

```
var url = 'http://hipsterjesus.com/api/'
console.log('Make request to ', url)
var req = new XMLHttpRequest()
req.open('GET', url)
req.onload = function() {
    console.log('Request status', req.status)
    console.log('Response size',
        req.response.toString().length)
}
req.send()
```

```
Make request to
http://hipsterjesus.com/api/
Request status 200
Response size 2342
```

The “a” in AJAX


classic web application model (synchronous)
client
user activity
user activity
user activity
data transmission
data transmission
data transmission
data transmission
time
system processing
system processing
server
Ajax web application model (asynchronous)
client
browser UI
user activity
input
display
input
display
input
display
input
display
Ajax engine
client-side processing
data transmission
data transmission
data transmission
data transmission
data transmission
data transmission
data transmission
data transmission
time
server-side processing
server-side processing
server-side processing
server-side processing
server
Jesse James Garrett / adaptivepath.com

Event Timeline



jQuery's $.ajax() … $.get() …. $.post()

```
$.getJSON('http://hipsterjesus.com/api/', function(data) {
  $('body').append(data.text);
});
```

```
$.getJSON('http://hipsterjesus.com/api/')

.done(function(data) {
  $('body').append(data.text);
})
.fail(function() {
  $('body').append('<p>Oh no, something went wrong!</p>');
})
.always(function() {
  $('body').append('<p>I promise this will always be added!.</p>');
});
```

http://davidwalsh.name/write-javascript-promises

Modernizr [sic]

Modernizr tells you what HTML, CSS and JavaScript features the user's browser has to offer.

```
<script>
  if (Modernizr.canvas) {
    alert("This browser supports HTML5 canvas!");
  } else {
    alert("no canvas :(");
  }
</script>
```

Polyfill

- Backport new technologies onto old platforms with ….
  JavaScript libraries of course!

```
Modernizr.load({
  test: Modernizr.canvas,
  nope: 'http://flashcanvas.net/bin/flashcanvas.js'
});
```

http://www.creativebloq.com/html5/10-best-polyfills-2133015

Another polyfill example

```
<script src="modernizr.js"></script>
<script>Modernizr.load({
  test: Modernizr.inputtypes.date,
  nope:
['http://ajax.googleapis.com/ajax/libs/jquery/1.4.4/jqu
ery.min.js',
'http://ajax.googleapis.com/ajax/libs/jqueryui/1.8.7/jq
uery-ui.min.js', 'jquery-ui.css'],
  complete: function () {
    $('input[type=date]').datepicker({
      dateFormat: 'yy-mm-dd'
    });
  }
});
</script>
```

Testing

1. Unit tests prove that your code actually works
2. You get a low-level regression-test suite
3. You can improve the design without breaking it
4. It's more fun to code with them than without
5. They demonstrate concrete progress
6. Unit tests are a form of sample code
7. It forces you to plan before you code
8. It reduces the cost of bugs
9. It's even better than code inspections
10. It virtually eliminates coder's block
11. Unit tests make better designs
12. It's faster than writing code without tests

CODING HORROR

programming and human factors

20 Jul 2006

I Pity The Fool Who Doesn't Write Unit Tests

J. Timothy King has a nice piece on the twelve benefits of writing unit tests first.

You'll get no argument from me on the overall importance of unit tests. I've increasingly come to believe that **unit tests are so important that they should be a first-class language construct**.

http://blog.codinghorror.com/i-pity-the-fool-who-doesnt-write-unit-tests/

http://www.jtse.com/blog/2006/07/11/twelve-benefits-of-writing-unit-tests-first

https://www.clear.rice.edu/comp431/sample/hello-jasmine.spec.js

Unit Testing with Jasmine

```
1   describe('Jasmine test of jQuery page', function() {
2
49    var clickSecondDiv = function($) {
50      return new Promise (function(resolve, reject) {
51        $('#div2').trigger("click");
52        setTimeout(function() {
53          resolve()
54        }, 500)
55      })
56    }
57
58    it("should click first div makes it vanish", function(done) {
59      var $div1 = $('#div1')
60      expect($div1.is(":visible")).toBe(true)
61      clickFirstDiv($).then(function() {
62        expect($div1.is(":visible")).toBe(false)
63        done()
64      })
65    })
66
67
68    it("should click second div makes first visible", function(done) {
```

DEMO

Jasmine

Behavior-Driven JavaScript

In-Class Exercise: Test Driven Development with Jasmine

***Turnin* 3 files *to* COMP431-S16:inclass-9**

- Download the following files:
  - https://www.clear.rice.edu/comp431/sample/jasmine-inclass9.html
  - https://www.clear.rice.edu/comp431/sample/jasmine-inclass9.js
  - https://www.clear.rice.edu/comp431/sample/jasmine-inclass9.spec.js

  Add a checkbox to the fixture
  - When the box is checked, turn the span text color to red
  - When the box is unchecked, turn the span text color to green
- Implement the test for the checkbox
- Flesh out the tests for the moveObject() function
  - I supplied "answers" that you want to write assertions for
- Implement the moveObject() function