jQuery

Roy McElmurry
jQuery

jQuery is a popular javascript framework like prototype. So why then do we learn Prototype in class?
Bread and Butter: $(), or jQuery()

This is very different from prototype's $ function.

If this confuses you or you need prototype as well, you can try using jQuery's noConflict() method or use the jQuery() function instead.
$( ) - Existing dom object

Calling $ with a DOM object gives the DOM object all the extra functionality of jQuery DOM objects.

```javascript
var element = document.getElementById('main');
var element = $(element);
// Now we can call jQuery methods on element
```
$() - Selector String

Calling $ with a css selector returns an array of jQuery DOM objects in the same way that document.querySelectorAll().

```javascript
var a1 = document.querySelectorAll('.small');
var a2 = $('.small');
// Same set of elements are found
```
$() - Tag String

Calling $ with a string that looks like html builds a new jQuery DOM object and returns it to you.

```javascript
var p1 = $('<p>');</var
var p2 = $('<p>Text in here</p>');
var ul = $('<ul><li>lists!</li></ul>');
```
$ - Selector String Context

You can perform selector calls with an optional second parameter which tells $ to search in that element rather than the whole page.

```javascript
var list = $('#list');
var elements = $('li', list);

// This is also equivalent to
var elements = list.find('li');
```
$(document).ready()

This is similar to `window.onload` but helps handle some inconsistencies across browsers.

```javascript
$(document).ready(function() {
    // Put your main code here
});
```
Traversing DOM methods

- children()
- parent()
- parents()
- each()
- sibling()
- last()
- first()
- next()
- not()
Manipulating DOM methods

- width()
- height()
- outerWidth()
- outerHeight()
- css()
- attr()
- text()
- html()
- val()
- addClass()
- append(elem)
- appendTo(elem)
- prepend(elem)
- prependTo(elem)
- after(elem)
- before(elem)
- remove()
- replaceAll(elems)
- wrap()
- unwrap()
Chaining

$ always returns an array of elements and methods operate on either every element when appropriate or just the first.

```javascript
var ps = $('p');
ps.css('backgroundColor', 'green');
// what will happen if there are many <p> tag
// on the page?
```
Events

- click()
- change()
- focus()
- hover()
- mouseover()
- mouseout()
- toggle()

Supply a function to set an event handler

Supply nothing to invoke the event!

Toggle is awesome, it gives you alternating behavior.
$.each

For each loops are broken in javascript, they loop over the keys....

$.each() takes a function and gives it both the key and the value as its first two parameters.
jQuery gives you a nice wrapper around AJAX.

```javascript
$.ajax({
    url: "someURL.php",
    type: "POST",
    data: {},
    dataType: "xml"
}).done(function(data) {
    // Do stuff with data
}).fail(function(xhr, status) {
    // Respond to an error
});
```
The best part

There are so many cool libraries that use jQuery. Check out their builtin plugin list.

http://plugins.jquery.com/