



CS472 WAP

jQuery Intro

DOM, JQuery

Except where otherwise noted, the contents of this document are Copyright 2012 Marty Stepp, Jessica Miller, Victoria Kirst and Roy McElmurry IV. All rights reserved. Any redistribution, reproduction, transmission, or storage of part or all of the contents in any form is prohibited without the author's expressed written permission. Slides have been modified for Maharishi University of Management Computer Science course CS472 in accordance with instructors agreement with authors.

Problems with DOM

- “The *DOM*. It is what most people hate when they say they hate JavaScript” (Douglas Crockford)
- the DOM can be clunky to use
- the same code doesn't always work the same way in every browser
 - code that works great in Chrome, Firefox, Safari, ... may fail in IE and vice versa
- Many improvements in recent years

jQuery framework

- the jQuery JavaScript library adds many useful features to JavaScript:
 - many useful extensions to the DOM
 - adds utility functions for built-in types String, Array, Object, Function
 - improves event-driven programming
 - many cross-browser compatibility fixes
 - makes Ajax programming easier (seen later)

```
<script src="https://code.jquery.com/jquery-3.4.1.min.js"  
  integrity="sha256-CSXorXvZcTkaix6Yvo6HppcZGetbYMGWSFlBw8HfCJo="   
  crossorigin="anonymous"> </script>
```

jQuery Design Principles

- jQuery is powerful because of these design principles
 - an expressive method for defining a set of elements, a superset of CSS selectors
 - useful and commonly needed methods for navigating the DOM tree
 - heavily overloaded APIs
 - functional programming techniques that apply operations to sets of elements at a time
 - method chaining for succinct operations

jQuery boilerplate

```
/* select the id=square, and give it  
event handler with an alert for click */
```

```
$( '#square' ).click(function()  
    { alert("Boink.");  
});
```

window.onload

- We cannot use the DOM before the page has been constructed. jQuery gives us a more convenient way to do this.

- The DOM way

```
window.onload = function() {  
    // do stuff with the DOM - when everything (images, etc) is ready  
}
```

- The direct jQuery translation

```
$(document).ready(function() {  
    // do stuff with the DOM - as soon as html loaded  
});
```

- The jQuery way

```
$(function() {  
    // do stuff with the DOM  
});
```

