

DOM events

- Every element in the DOM supports a variety of events
 - Page load, mouse click, mouse moving over an element, selecting from a drop-down menu, etc.
- JavaScript functions are registered as *event handlers*
 - Functions called when the events happen

jQuery event binding

- Get handle to element

```
$( "#mybutton" )
```

- Bind to valid events for that element

- Each event has a corresponding bind function:

```
$( "#mybutton" ).click( myFunction );
```

- Alternately, use the “on” function for any binding:

```
$( "#mybutton" ).on( "click", myFunction );
```

- Note no () after myFunction, otherwise it will:

- Call the function
- Bind the return value to the click event

Document loading

- JavaScript manipulates elements in the DOM
 - Web browsers load incrementally
 - So you can see parts of the page as it loads
 - This makes it possible to refer to a part of the DOM that hasn't loaded yet
- Solution: put initialization code in a function and delay execution until the page is loaded
 - `$(init);`
 - `$(document).ready(init);`
- Current convention is to put `<script>` at the end of page

Anonymous functions

- JavaScript makes heavy use of *anonymous* functions
 - Functions without a name that are only called once

- Instead of:

```
function myFunction()  
{  
    // do stuff  
}  
elem.click( myFunction );
```

- Do:

```
elem.click( function ()  
{  
    // do stuff  
} );
```

- In both cases a callable function is bound

this

- `this` refers to the element that owns the function
 - For most functions, that is the document (top-level)
- Registered event handlers are copied
 - `element.click(my_function);`
 - Creates a copy of `my_function` as a method on `element`
 - So `this` refers to that element inside the function
- Note: old-style inline registration does not copy!
 - The “global” function is called, so `this` is the document
 - Leads to some confusing things, so avoid it

Event handler return value

- Event handlers can suppress *default behavior*
 - If an `<a onclick>` handler returns false, the link is not followed
 - If a `<input type="submit">` handler returns false, the form is not submitted