Intro to JQuery

Topics

- History + Introduction
- Selecting + manipulating elements
- Manipulating JavaScript objects and collections
- Chaining
- The jQuery event model
- Writing your own plugins

History

- Project started by John Resig in 2005
- Original aim was to make selecting elements in a web page easier
- Since then it has become the most popular JavaScript library
- Used by 70% of all sites

What is Jquery?

- Jquery is a free, open source JavaScript library designed to make it easier to navigate a document, select DOM elements, handle events, and make server calls (AJAX).
- Also provides mechanism to create plugins that provide additional functionality (ie. validation and animations).
- Let's start with navigating and selecting DOM elements in detail..

Selecting DOM elements

- Let's say you want to select elements by id – not that difficult.
- But let's say you want to select the third element in a <li>.
  - In native JavaScript, you would have to write code to loop through the
  - Also, some older browsers might have different implementations = have to code for that.
  - In jQuery:

```
$('li:nth-child(3)')
```

Selecting DOM elements

- Now suppose you want to get the closest of that <li>:

```
$('li:nth-child(3)').closest('div')
```

- Jquery provides a ton of convienence methods for selecting and traversing elements:

```
.parent(), .prev(), siblings(),
next()
```

CSS Selectors

- Web developers have become familiar with a small, but powerful and very useful, group of selection expressions that work across all browsers
- Those expressions can select by an element's ID, by CSS class names, by tag names, and by the hierarchy of the page elements within the DOM.
- jQuery piggybacks off these css selectors for selecting the elements you want.

CSS Selectors

- `jQuery('a')` - Matches all anchor (<a>) elements
- `jQuery('#specialID')` - Matches the element with the id value of specialID
- `jQuery('.specialClass')` - Matches all elements with the class specialClass
- `jQuery('a#specialID.specialClass')` - Matches the element with the id value specialID if it's an anchor tag and has class specialClass
- `jQuery('p a.specialClass')` - Matches all anchor elements with the class specialClass that are descendants of <p> elements

Advanced CSS Selectors

- The following are the more advanced selectors:

  ```
  $(“div[title^='my']”)
  ```

  - This selects all elements with a title attribute whose value begins with 'my'.

  ```
  $(“a[href$='.pdf']”)
  ```

  - This is a useful selector for locating all links that reference PDF files.

  ```
  $(“li:4th-child”)
  ```

  - This matches the 4th child of the li element

CSS Selectors with filters

- So far we have quite a bit of power when selecting elements but there are even more selectors that give us further ability to filter the selections.

  ```
  :checked, :disabled, :not(selector), :parent
  ```

- Some examples:

  ```
  $('img:not([src*="dog"])')
  $('input:checked')
  ```

  - Selects all checkbox elements

- (source selectors.html)

Managing the matching element set

- Once you have declared your selectors, you want can get the nth element for that set.
- jQuery wraps the result of the returned set and adds convienence methods to them, in addition to the regular Array methods.
- Common operations are:
  - `get(index), first(), last()`

Performing operations on the matching set

- A common use case is to perform some action(s) on the matching set.
- To do that, jQuery provides functions you can call and it provides the ability to chain operations.
- Let's look at operations first..

Performing operations on the matching set

- Examples of the `each()` usage:

```
$('img').each(function(n) {
  this.alt = 'This is image[' + n + '] with an
  id of ' + this.id;
});


//another example
$([1,2,3]).each(function() { alert(this); });
```

Chaining method calls

- jQuery also allows you to chain method calls so that you can write less code:

```
// you can code like this
jQuery('#content').addClass('dummy');
jQuery('#content').find('img');
jQuery('#content').find('bubble');
jQuery('#content').addClass('polaroid');


//or this
jQuery('#content').addClass('dummy').find('img
').addClass('bubble').addClass('polaroid');
```

Manipulating the DOM

- jQuery provides convenience methods for adding, removing, and updating DOM elements.
  - `css(), text(), html(), append(content), prepend(content), wrap(wrapper), remove()`
- (dom_manipulation.html)

Event Bubbling

- Javascript has the *event bubbling* concept.
- When an event is triggered on an element in the DOM tree, the event-handling mechanism of the browser checks to see if a handler has been established for that particular event on that element and, if so, invokes it.
- After the target element has had its chance to handle the event, the event model checks with the parent of that element to see if it has established a handler for the event type, and if so, it’s also invoked—after which its parent is checked, then its parent, then its parent, and on and on, all the way up to the top of the DOM tree.
- (source event_bubbling.html)

Event Bubbling

- Sometimes it is useful to stop the event from propagating
- We can stop this behavior in jQuery by calling the `stopPropgation()` on the event (inside the handler function).
- (source stop_event_bubbling.html)

Adding events listeners via jQuery

- As we've seen before, jQuery provides ways in which we can hook in to the event lifecycle:
  - on(event, function):

```
$( "#dataTable tbody tr" ).on( "click",
function() {
  alert( $( this ).text() );
});
```

  - trigger(event):

```
$( "#foo" ).trigger( "click" );
```

Extending jQuery

- We can build on top of jQuery by building plugins ourselves.
- We can wrap commonly used features into a plugin that we can maintain in one place.
- Also promotes code reuse for different pages/projects.

Skeleton of a plugin

- The following shows a skeleton of a plugin:

```
(function($){
  // Plugin content goes here
})(jQuery);
```

- Does this look familiar? Think back to last week.
- (jquery_plugin.html)

The End

- Questions?
- Any feedback would be great!
- tochan@contractor.ea.com