

Interacting with the DOM

Dan Wahlin

www.pluralsight.com



Agenda

- Iterating Through Nodes
- Modifying DOM Object Properties
- Modifying Attributes
- Adding and Removing Nodes
- Modifying Styles
- Modifying Classes

Iterating Through Nodes

- `.each(function(index, Element))` is used to iterate through jQuery objects:

```
$('#div').each(function(index) {  
    alert(index + ' = ' + $(this).text());  
});
```

Iterates through each div element and returns its index number and text

```
$('#div').each(function(index,elem) {  
    alert(index + ' = ' + $(elem).text());  
});
```

elem = this

Agenda

- Iterating Through Nodes
- **Modifying DOM Object Properties**
- Modifying Attributes
- Adding and Removing Nodes
- Modifying Styles
- Modifying Classes

Modifying Object Properties

- The ***this.propertyName*** statement can be used to modify an object's properties directly:

```
$('div').each(function(i) {  
    this.title = "My Index = " + i;  
});
```

Iterates through each div and modifies the title. If the property does not exist, it will be added

Agenda

- Iterating Through Nodes
- Modifying DOM Object Properties
- **Modifying Attributes**
- Adding and Removing Nodes
- Modifying Styles
- Modifying Classes

Accessing Attributes

- Object attributes can be accessed using **attr()**:

```
var val = $('#CustomerDiv').attr('title');
```

Retrieves the value of the title attribute

Modifying Attributes

- `.attr(attributeName, value)` is the method used to access an object's attributes and modify the value:

```
$('img').attr('title', 'My Image Title');
```

Changes the title attribute to a value of *My Image Title*

Modifying Multiple Attributes

- To modify multiple attributes, pass a JSON object containing name/value pairs:

```
$('img').attr({  
  title: 'My Image Title',  
  style: 'border:2px solid black';  
});
```

JSON object passed and used to change title and border

Wait...What's JSON?

- JSON delimits objects using { and }
- The : character separates properties and values

```
{  
  FirstName: 'John',  
  LastName: 'Doe',  
  Address: {  
    Street: '1234 Anywhere St.',  
    City: 'Phoenix',  
    State: 'AZ',  
    ZipCode: 85249  
  }  
}
```

Agenda

- Iterating Through Nodes
- Modifying DOM Object Properties
- Modifying Attributes
- **Adding and Removing Nodes**
- Modifying Styles
- Modifying Classes

Adding and Removing Nodes

- Four key methods handle inserting nodes into elements:

`.append()`

`.appendTo()`

`.prepend()`

`.prependTo()`

- To remove nodes from an element use `.remove()`

Appending to Nodes

- Appending adds children at the end of the matching element:

```
$('<span> (office)</span>').appendTo('.officePhone');
```

OR

```
$('.officePhone').append('<span> (office)</span>');
```

Would result in (office) being added into each .officePhone class element

Prepending to Nodes

- Prepending adds children at the beginning of the matching element:

```
$('<span>Phone: </span>').prependTo('.phone');
```

OR

```
$('.phone').prepend('<span>Phone: </span>');
```

Would result in Phone: being added into each .phone class element

Wrapping Elements

- The following HTML and `.wrap()` function:

```
<div class="state">Arizona</div>
```

```
$('.state').wrap('<div class="US_State" />');
```

Results in:

```
<div class="US_State">  
  <div class="state">Arizona</div>  
</div>
```

Removing Nodes

- `.remove()` will remove matched elements from the DOM:

```
$('.phone, .location').remove();
```

Will result in objects with `.phone` or `.location` classes being removed from the DOM

Agenda

- Iterating Through Nodes
- Modifying DOM Object Properties
- Modifying Attributes
- Adding and Removing Nodes
- **Modifying Styles**
- Modifying Classes

Modifying Styles

- The `.css()` function can be used to modify an object's style:

```
$("#div").css("color", "red");
```

Modifying Multiple Styles

- Multiple styles can be modified by passing a JSON object:

```
$('#div').css( {  
    'color' : '#ccc',  
    'font-weight' : 'bold'  
});
```

Agenda

- Iterating Through Nodes
- Modifying DOM Object Properties
- Modifying Attributes
- Adding and Removing Nodes
- Modifying Styles
- **Modifying Classes**

Modifying CSS Classes

- The four methods for working with CSS Class attributes are:

`.addClass()`

`.hasClass()`

`.removeClass()`

`.toggleClass()`

Adding a CSS Classes

- **.addClass()** adds one or more class names to the class attribute of each matched element:

```
$('p').addClass('classOne');
```

- More than one class:

```
$('p').addClass('classOne classTwo');
```

Matching CSS Classes

- **.hasClass()** returns true if the selected element has a matching class that is specified:

```
if($('p').hasClass('styleSpecific')) {  
    //Perform work  
}
```

Removing CSS Classes

- `.removeClass()` can remove one or more classes:

```
$('p').removeClass('classOne classTwo');
```

- Remove all class attributes for the matching selector:

```
$('p').removeClass();
```


Toggling CSS Classes

- **.toggleClass()** alternates adding or removing a class based on the current presence or absence of the class:

```
$('#PhoneDetails').toggleClass('highlight');
```

```
<style type="text/css">  
  .highlight { background:yellow; }  
</style>
```

Summary

- **jQuery selectors can be combined with DOM functions to perform a variety of functionality:**
 - ☐ Iterate through elements
 - ☐ Add elements
 - ☐ Remove elements
 - ☐ Wrap elements
 - ☐ Change styles
 - ☐ Change classes
 - ☐ Toggle classes

For more in-depth **online** developer **training** visit



on-demand content from authors you **trust**