

# Interacting with DOM

Lesson 04

## Lesson Objectives

- Iterating Through Nodes
- Working with Attributes and DOM Content
- DOM Insertion and Removal
- Working with Classes



4.1. Iterating Through Nodes

## 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

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

4.1. Iterating Through Nodes

## Iterating through Nodes

- The `this.PropertyName` statement can be used to modify an object's properties directly.

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

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

## Working with Attributes and DOM Content

- Object attributes can be used using `attr()`:
  - `var val = $('#customDiv').attr('title');` - Retrieves the title attribute value
- `.attr(attributeName,value)` is the method used to access an object's attributes and modify the values.
  - `$('#img').attr('title','Image title');` - changes the title attribute value to Image title.
- To modify multiple attributes, pass JSON object.

```
$('#img').attr({  
  "title": "image title",  
  "style" : "border:2px solid black"  
});
```

4.3. DOM Insertion and Removal

## DOM Insertion and Removal

- To insert new nodes four methods are available in jQuery
- Appending adds children at the end of the matching elements
  - `.append()`
  - `.appendTo()`
- Prepending adds children at the beginning of the matching elements
  - `.prepend()`
  - `.prependTo()`
- To wrap the elements use `.wrap()`
- To remove nodes from an element use `.remove()`

## Working with Classes

- The four methods for working with css class attributes are
  - `.addClass()`
  - `.hasClass()`
  - `.removeClass()`
  - `.toggleClass()`
- `.addClass()` adds one or more class names to the class attribute of each element.
  - `$('.p').addClass('classOne');`
  - `$('.p').addClass('classOne classTwo');`
- `.hasClass()` returns true if the selected element has a matching class that is specified
  - `if($('.p').hasClass('classOne')) { //perform operation }`

4.4. Working with Classes


## Working with Classes (Contd)


- `.removeClass()` can remove one or more classes
  - `$('.p').removeClass('classOne classTwo');`
- To remove all class attributes for the matching selector
  - `$('.p').removeClass();`
- `.toggleClass()` alternates adding or removing a class based on the current presence or absence of the class.
  - `$('#targetDiv').toggleClass('highlight');`
- `.css()` function is used to modify an object's style
  - `$('.div').css('color','red');`
- Multiple styles can be modified by passing a JSON Object
  - `$('.div').css({"color":"red", "font-weight":"bold"});`



# Demo

- jQuery-DOM-Manipulation.html





Copyright © Capgemini 2015. All Rights Reserved 9

Add the notes here.

### 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 Class
  - Toggle Class



Add the notes here.

## Lab

- 4.1. Iterate Nodes using jQuery
- 4.2. Sort list items using jQuery



Copyright © Capgemini 2015. All Rights Reserved 11

Add the notes here.

## Review Question

- Question 1 : Which jQuery method is used to set one or more style properties for selected elements?
  - `css()`
  - `style()`
  - `html()`
  - `text()`



- Question 2 : Which of the snippets below creates a new div and appends it to the first span on the page?
  - `$('span').createElement('div','This is a new div!');`
  - `$('<div>',{html:"This is a new <b>div</b>"}).appendTo('span:first');`
  - `$('span').html('<div>This is a new div!</div>')`

Add the notes here.