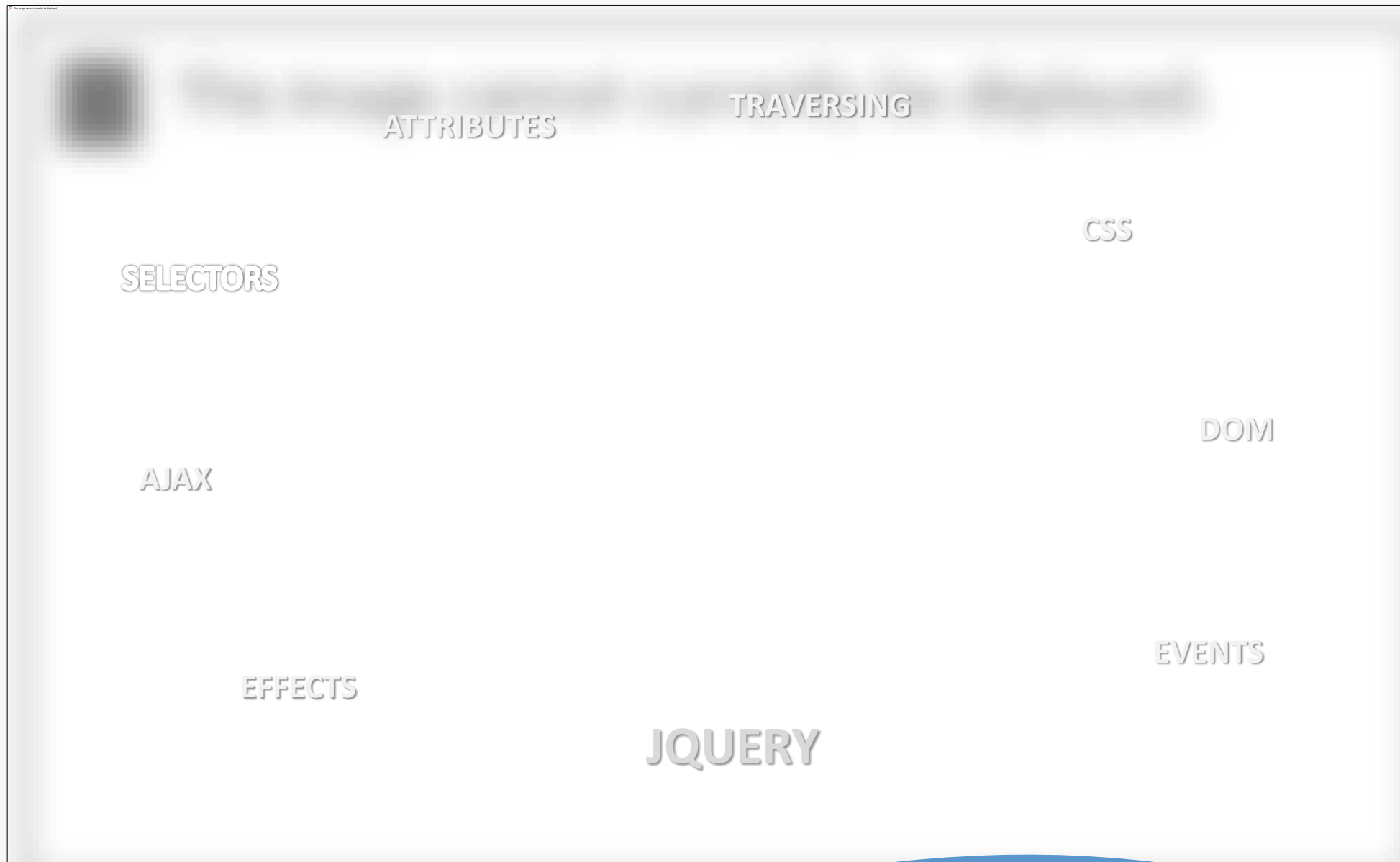




JQUERY



Audience

This tutorial is designed for software programmers who wants to learn the basics of jQuery and its programming concepts in simple and easy ways.

Prerequisites

That you are in this class means you have a basic understanding of HTML, CSS, JavaScript, Document Object Model (DOM) and any text editor.



Overview

What is jquery?

jQuery is a fast and concise JavaScript Library created by John Resig in 2006 with a nice motto – **Write less, do more.**

jQuery is a JavaScript toolkit designed to simplify various tasks by writing less code. Here is the list of important core features supported by jQuery.

Core Features

DOM Manipulation

select DOM elements, traverse them and modify their content

Animations

The jQuery comes with plenty of built-in animation effects which you can use in your websites.

Event Handling

Capture a wide variety of events, e.g. user clicking on a link, without the need to clutter the HTML code itself with event handlers.

AJAX Support

Helps you a lot to develop a responsive and feature-rich site using AJAX technology.

Cross Browser Support

IE 6.0+, FF 2.0+, Safari 3.0+, Chrome and Opera 9.0+



How to use jQuery?

Local Installation

You can download jQuery library on your local machine and include it in your HTML code.

<https://jquery.com/download/> to download the latest version available.

OR

CDN Based Version

You can include jQuery library into your HTML code directly from Content Delivery Network (CDN).

Google and Microsoft provides content deliver for the latest version.

```
<script type="text/javascript"  
src="http://ajax.googleapis.com/ajax/libs/jquery/2.1.3/jquery.mi  
n.js">  
</script>
```



jQuery Selector

What is jQuery Selector?

A jQuery Selector is a function which makes use of expressions to find out matching elements from a DOM based on the given criteria. Simply you can say, selectors are used to select one or more HTML elements using jQuery. Once an element is selected then we can perform various operations on that selected element.

The `$()` factory function

jQuery selectors start with the dollar sign and parentheses – `$()`. The factory function `$()` makes use of following three building blocks while selecting elements in a given document:

NOTE – The factory function `$()` is a synonym of `jQuery()` function. So in case you are using any other JavaScript library where `$` sign is conflicting with some thing else then you can replace `$` sign by `jQuery` name and you can use function `jQuery()` instead of `$()`.

jQuery Building Blocks

Tag Class

Represents a tag available with the given class in the DOM.
For example
`$('.myclass')` selects all elements in the document that have a class of myclass.

Tag ID

Represents a tag available with the given ID in the DOM.
For example
`$('#myid')` selects the single element in the document that has an ID of myid.

Tag Name

Represents a tag name available in the DOM.
For example `$('p')` selects all paragraphs `<p>` in the document.

Example

Tag Class

```
<script type="text/javascript" language="javascript">  
    $(document).ready(function() {  
        $(".myClass").css("background-color", "yellow");  
    });  
</script>
```

Example

Tag ID

```
<script type="text/javascript" language="javascript">
    $(document).ready(function() {
        $("#myid").css("background-color", "yellow");
    });
</script>
```


Example

Tag Name

```
<script type="text/javascript" language="javascript">  
    $(document).ready(function() {  
        $("p").css("background-color", "yellow");  
    });  
</script>
```

Selectors and Description

#ID

Selects a single element which matches with the given ID.

tagName

Selects all elements which match with the given element Name.

.Class

Selects all elements which match with the given Class.

**Universal
(*)**

Selects all elements available in a DOM

**Multiple
Elements E, F, G**

Selects the combined results of all the specified selectors E, F or G.

Selectors Examples

`$("tag
Name"
)`

This selector
selects all
elements with
the tag name
"tagName"

`$("p > *")`

This selector
selects all
elements that
are children of a
paragraph
element.

`$("#specialID")`

This selector
function gets
the element
with
id="specialID".

`$(".special
Class")`

This selector
gets all the
elements that
have the class
of specialClass.

`$("li:not(.myclass
)")`

Selects all
elements
matched by
 that do not
have
class="myclass.

Selectors Examples

`$("a#specialID.
specialClass")`

This selector matches links with an id of specialID and a class of specialClass.

`$("p
a.specialClass")`

This selector matches links with a class of specialClass declared within <p> elements.

`$("ul li:first")`

This selector gets only the first element of the .



jQuery Attributes

Get Attribute Value

The **attr()** method can be used to either fetch the value of an attribute from the first element in the matched set or set attribute values onto all matched elements.

Example

```
<script type="text/javascript" language="javascript">  
    $(document).ready(function() {  
        var title = $("em").attr("title");  
        $("#divid").text(title);  
    });  
</script>
```

Set Attribute Value

The **attr(name, value)** method can be used to set the named attribute onto all elements in the wrapped set using the passed value.

Example

```
<script type="text/javascript" language="javascript">  
    $(document).ready(function() {  
        $("#myimg").attr("src", "/images/jquery.jpg");  
    });  
</script>
```

Applying Styles

The **`addClass(classes)`** method can be used to apply defined style sheets onto all the matched elements. You can specify multiple classes separated by space.

Example

```
<script type="text/javascript" language="javascript">  
    $(document).ready(function() {  
        $("em").addClass("selected");  
        $("#myid").addClass("highlight");  
    });  
</script>
```

Attribute Methods and Description

**toggleClass(
class)**

Set a key/value object as properties to all matched elements.

**attr(
properties)**

Set a key/value object as properties to all matched elements.

attr(key, fn

Set a single property to a computed value, on all matched elements.

**removeAttr(
name)**

Remove an attribute from each of the matched elements.

**hasClass(
class)**

Returns true if the specified class is present on at least one of the set of matched elements.

**removeClass
(class)**

Removes all or the specified class(es) from the set of matched elements

Attribute Methods and Description

val()

Get the input value of the first matched element.

val(val)

Set the value attribute of every matched element if it is called on <input> but if it is called on <select> with the passed <option> value then passed option would be selected, if it is called on check box or radio box then all the matching check box and radiobox would be checked.

text()

Get the combined text contents of all matched elements.

text(val)

Set the text contents of all matched elements.

html()

Get the html contents (innerHTML) of the first matched element.

html(val)

Set the html contents of every matched element.

Example: Selector and Description

`$("#myID").attr("custom")`

This would return value of attribute custom for the first element matching with ID myID.

`$("img").attr("alt", "Sample Image")`

This sets the alt attribute of all the images to a new value "Sample Image".

`$("a").removeAttr("target")`

This would remove target attribute of all the links.

`$("p").text()`

Returns string that contains the combined text contents of all matched <p> elements.

`$("p").text("<i>Hello World</i>")`

This would set "<i>Hello World</i>" as text content of the matching <p> elements.

`$("p").html()`

This returns the HTML content of the all matching paragraphs.

jQuery-DOM Traversing

DOM Traversing

jQuery is a very powerful tool which provides a variety of DOM traversal methods to help us select elements in a document randomly as well as in sequential method.

Find Element by index

Consider a simple document with the following HTML content :

```
<html>  
<head> <title>JQuery Example</title> </head>  
<body>  
  <div>  
    <ul>  
      <li>one</li>  
      <li>two</li>  
      <li>three</li>  
    </ul>  
  </div>  
</body>  
</html>
```

Find Element by index

```
<html>
<head>
  <title>jQuery Example</title>
  <script type="text/javascript"
src="http://ajax.googleapis.com/ajax/libs/jquery/2.1.3/jquery.min.js"> </script>
  <script type="text/javascript" language="javascript">
    $(document).ready(function() { $("li").eq(2).addClass("selected"); });
  </script> <style> .selected { color:red; } </style> </head>
```

Filtering out Elements

The **filter(selector)** method can be used to filter out all elements from the set of matched elements that do not match the specified selector(s).

The selector can be written using any selector syntax.

Example: Filtering out Elements

```
<script type="text/javascript" language="javascript">  
    $(document).ready(function() {  
        $("li").filter(".middle").addClass("selected");  
    });  
</script>
```

Locating Descendent Elements

The **find(selector)** method can be used to locate all the descendent elements of a particular type of elements. The selector can be written using any selector syntax.

Example: Locating Descendent Elements

```
<script type="text/javascript" language="javascript">  
    $(document).ready(function() {  
        $("p").find("span").addClass("selected");  
    });  
</script>
```

```
<p>This is 1st paragraph and <span>THIS IS RED</span></p>  
<p>This is 2nd paragraph and <span>THIS IS ALSO  
RED</span></p>
```

JQuery DOM Filter Methods

Used to filter out various elements from a list of DOM elements

Method and Description

eq(index)

Reduce the set of matched elements to a single element

**filter(
selector)**

Removes all elements from the set of matched elements that do not match the specified selector(s).

not(selector)

Returns all elements that do not match the criteria. It is the opposite of the filter()

First()

The first() method returns the first element of the selected elements.

JQuery DOM Traversing Methods

JQuery DOM traversing methods are used to locate various elements in the DOM

Method and Description

andSelf()

Add the previous selection to the current selection

children()

Get a set of elements containing all of the immediate children of each of the matched set of elements.

add(selector)

Adds elements to the set of matched elements

next()

Returns the next sibling of the selected element

Method and Description

closest(selector)

Get a set of elements containing the closest parent element that matches the specified selector, the starting element included.

contents()

Find all the child nodes inside the matched elements (including text nodes), or the content document, if the element is an iframe.

find(selector)

Searches for descendent elements that match the specified selectors

offsetParent()

Returns the first positioned parent element.



JQuery CSS Selection Methods

Apply CSS Properties

Here is the syntax for the method –

selector.css(PropertyName, PropertyValue);

Example:

```
<script type="text/javascript" language="javascript">  
    $(document).ready(function() {  
        $("li").eq(2).css("color", "red");  
    });  
</script>
```

Apply Multiple CSS Properties

Here is the syntax for the method –

selector.css({key1:val1, key2:val2....keyN:valN})

Example:

```
<script type="text/javascript" language="javascript">  
    $(document).ready(function() {  
        $("li").eq(2).css({"color":"red", "background-  
color":"green"});  
    });  
</script>
```

Setting Element Width and Height

The **width(val)** and **height(val)** method can be used to set the width and height respectively of any element.

Example:

```
<script type="text/javascript" language="javascript">  
    $(document).ready(function() {  
        $("div:first").width(100);  
        $("div:first").css("background-color", "blue");  
    });  
</script>
```

Method and Description

width(val)

Sets the CSS width of every matched element.

css(name)

Returns a style property on the first matched element.

position()

Gets the top and left positions of an element relative to its offset parent.

offset()

Get the current top and left offsets of the first matched element, in pixels, relative to the document.

Method and Description

**css(name,
value)**

Set a single style property to a value on all matched elements.

innerHeight()

Gets the inner height (excludes the border and includes the padding) for the first matched element.

**scrollTop(val
)**

sets or returns the vertical scrollbar position for the selected elements.



JQuery-DOM Manipulation

Introduction

jQuery provides methods to manipulate DOM in efficient way. You do not need to write big code to modify the value of any element's attribute or to extract HTML code from a paragraph or division.

jQuery provides methods such as `.attr()`, `.html()`, and `.val()` which act as getters, retrieving information from DOM elements for later use.

Content Manipulation

The `html()` method gets the html contents (innerHTML) of the first matched element.

Here is the syntax for the method –

`selector.html()`

Example: Content Manipulation

```
script type="text/javascript" language="javascript">
    $(document).ready(function() {
        $("div").click(function () {
            var content = $(this).html();
            $("#result").text( content );
        });
    });    </script>
```

Example: Content Manipulation

```
<p>Click on the square below:</p>
```

```
<span id="result"> </span>
```

```
<div id="division" style="background-color:blue;">
```

```
  This is Blue Square!!
```

```
</div>
```

DOM Element Replacement

You can replace a complete DOM element with the specified HTML or DOM elements. The `replaceWith(content)` method serves this purpose very well.

Here is the syntax for the method –

`selector.replaceWith(content)`

Example: Element Replacement

```
<script type="text/javascript" language="javascript">  
    $(document).ready(function() {  
        $("div").click(function () {  
            $(this).replaceWith("<h1>jQuery is Great</h1>");  
        });  
    });  
</script>
```


Example: Element Replacement

p>Click on the square below:</p>

<div id="division" style="background-color:blue;">

This is Blue Square!!

</div>

Removing DOM Elements

The `empty()` method remove all child nodes from the set of matched elements where as the method `remove(expr)` method removes all matched elements from the DOM.

Here is the syntax for the method –

`selector.remove([expr])`
or
`selector.empty()`

Examples: Removing DOM Elements

```
<script type="text/javascript" language="javascript">  
    $(document).ready(function() {  
        $("div").click(function () {  
            $(this).remove( );  
        });  
    });  
</script>
```

Examples: Removing DOM Elements

<<p>Click on any square below:</p>

<div class="div" style="background-color:blue;"></div>

<div class="div" style="background-color:green;"></div>

<div class="div" style="background-color:red;"></div>

Inserting DOM Elements

The `after(content)` method insert content after each of the matched elements where as the method `before(content)` method inserts content before each of the matched elements.

Here is the syntax for the method –
`selector.after(content)`

or

`selector.before(content)`

Examples: Inserting DOM Elements

```
<script type="text/javascript" language="javascript">  
    $(document).ready(function() {  
        $("div").click(function () {  
            $(this).before('<div class="div"></div>' );  
        });  
    });  
</script>
```

Examples: Inserting DOM Elements

`<p>Click on any square below:</p>`

` `

`<div class="div" style="background-color:blue;"></div>`

`<div class="div" style="background-color:green;"></div>`

`<div class="div" style="background-color:red;"></div>`

DOM Manipulation Methods

**replaceAll(
selector)**

Replaces the elements matched by the specified selector with the matched elements.

**insertAfter(
selector)**

Insert all of the matched elements after another, specified, set of elements.

**insertBefore(
selector)**

Insert all of the matched elements before another, specified, set of elements.

**after(content
)**

Insert content after each of the matched elements.

DOM Manipulation Cont.

**append(
content)**

Append content to the inside of every matched element.

**appendTo(
selector)**

Append all of the matched elements to another, specified, set of elements.

**before(
content)**

Insert content before each of the matched elements.



jQuery - Events Handling

Binding Event Handlers

Using the jQuery Event Model, we can establish event handlers on DOM elements with the `bind()` method as follows –

The full syntax of the `bind()` command is as follows –

`selector.bind(eventType[, eventData], handler)`

Introduction

We have the ability to create dynamic web pages by using events. Events are actions that can be detected by your Web Application.

- A mouse click
- A web page loading
- Taking mouse over an element
- Submitting an HTML form
- A keystroke on your keyboard etc.

Example: Binding Event Handlers

```
<script type="text/javascript" language="javascript">  
    $(document).ready(function() {  
        $('div').bind('click', function( event ){  
            alert('Hi there!');  
        });  
    });  
</script>
```

Example: Binding Event Handlers

<p>Click on any square below to see the result:</p>

```
<div class="div" style="background-color:blue;">ONE</div>
```

```
<div class="div" style="background-  
color:green;">TWO</div>
```

```
<div class="div" style="background-  
color:red;">THREE</div>
```

Removing Event Handlers

jQuery provides the `unbind()` command to remove an exiting event handler. The syntax of `unbind()` is as follows :

`selector.unbind(eventType, handler)`

or

`selector.unbind(eventType)`



jQuery Effects

Showing and Hiding elements

The commands for showing and hiding elements are pretty much what we would expect – **show()** to show the elements in a wrapped set and **hide()** to hide them.

Syntax

The commands for showing and hiding elements are – **show()** to show the elements in a wrapped set and **hide()** to hide them.

Here is the simple syntax for show() method –

[selector].show(speed, [callback]);

Following is the simple syntax for hide() method –

[selector].hide(speed, [callback]);

Example

```
<script type="text/javascript" language="javascript">  
    $(document).ready(function() {  
  
        $("#show").click(function () {  
            $(".mydiv").show( 1000 );  
        });  
  
        $("#hide").click(function () {  
            $(".mydiv").hide( 1000 );  
        });  
    });  
</script>
```

Example

```
<div class="mydiv">  
  This is a SQUARE  
</div>
```

```
<input id="hide" type="button" value="Hide" />  
<input id="show" type="button" value="Show" />
```

Toggling Elements

jQuery provides methods to toggle the display state of elements between revealed or hidden. If the element is initially displayed, it will be hidden; if hidden, it will be shown.

Syntax

Here is the simple syntax for one of the toggle() methods –
`[selector]..toggle([speed][, callback]);`

Example

```
<script type="text/javascript" language="javascript">
```

```
$(document).ready(function() {  
    $(".clickme").click(function(event){  
        $(".target").toggle('slow', function(){  
            $(".log").text('Transition Complete');  
        });  
    });  
});  
</script>
```

Example

```
<div class="content">  
  <div class="clickme">Click Me</div>  
  <div class="target">  
      
  </div>  
  <div class="log"></div>  
</div>
```



JQuery Effects Methods

DOM Manipulation Methods

**animate(params,
[duration, easing,
callback])**

A function for making custom animations.

**fadeIn(speed,
[callback])**

Fade in all matched elements by adjusting their opacity.

**slideToggle(speed,
[callback])**

Toggle the visibility of all matched elements by adjusting their height.

**stop([clearQueue,
gotoEnd])**

Stops all the currently running animations on all the specified elements.

DOM Manipulation Cont.

`hide(speed,
[callback])`

Hide all matched
elements using a
graceful animation

`slideDown(
speed,
[callback])`

Reveal all
matched
elements by
adjusting their
height

`fadeTo(speed,
opacity, callback
)`

Fade the
opacity of all
matched
elements to a
specified



Questions