



# jQuery (Day 1)

Presentation By: Asfiya Khan (Sr. Technical Trainer)

## Agenda



Selectors



DOM  
Manipulation



Events

## Introduction to jQuery

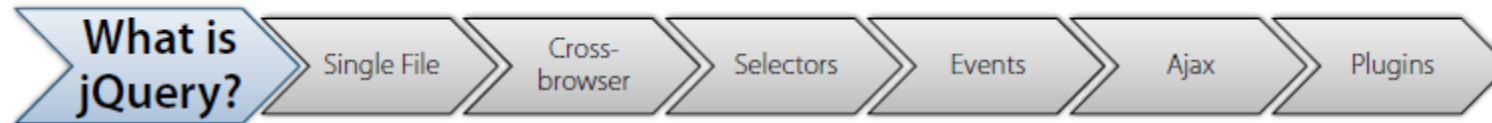


- Fast, small, and feature-rich JavaScript library.
- Makes HTML document traversal and manipulation, event handling, animation, and Ajax much simpler.
- Easy-to-use API that works across a multiple browsers.
- A combination of versatility and extensibility it has changed the way that millions of people write JavaScript.

## Introduction to jQuery



- DOM Traversal and Manipulation
- Event Handling
- CSS Manipulation
- Ajax
- And many more..



# jQuery Makes Client-Side Dev Fun!



## Including jQuery



- Download it and include using your standard script tags.
- Uncompressed - <https://code.jquery.com/jquery-3.3.1.js>
- Minified - <https://code.jquery.com/jquery-3.3.1.min.js>

## Including jQuery



### Using a Content Delivery Network (CDN)

- Alternatively, you can use the Microsoft or Google CDN:

```
<head>  
  <script type="text/javascript"  
    src="http://ajax.microsoft.com/ajax/jquery/jquery-[version].js"></script>  
</head>
```

```
<head>  
  <script type="text/javascript"  
    src="http://ajax.googleapis.com/ajax/libs/jquery/[version]/jquery.min.js">  
  </script>  
</head>
```



## Detecting When a Page has Loaded

- Use `$(document).ready()` to detect when a page has loaded and is ready to use
- Called once DOM hierarchy is loaded (but before all images have loaded)

```
<script type="text/javascript">  
    $(document).ready(function(){  
        //Perform action here  
    });  
</script>
```



# Using jQuery Selectors >>>

## Accessing Elements - Selectors



### Selector Syntax

`$(selectorExpression)`

`jQuery(selectorExpression)`



## Accessing Elements - Selectors

- **Select all divs**

```
var jqDivs = $('div');
```

- **Select element with id as "myid"**

```
var jqElement = $('#myid');
```

- **Select div with id as "mydiv"**

```
var jqDiv = $('div#mydiv');
```



## Accessing Elements – Class Selectors

- **Select elements with class as “myclass”**

```
var jqElements = $('.myclass');
```

- **Select elements with classes as “odd” and “even”**

```
var jqElements = $('.odd.even');
```

- **Select first div**

```
var jqDiv = $('div:first');
```

- **Select last div**

```
var jqDiv = $('div:last');
```



## Accessing Elements - Class Selectors

- **Select all even trs**

```
var jqRows = $('tr:even');
```

- **Select all odd trs**

```
var jqRows = $('tr:odd');
```

- **Select all elements with href attribute**

```
var jqLinks = $('[href]');
```

- **Select all anchor elements with href attribute equal to [www.google.com](http://www.google.com)**

```
var jqLinks = $('a[href="www.google.com"]');
```



## Accessing Elements – Attribute Selectors

- **Select anchor elements with href attribute not equal to [www.google.com](http://www.google.com)**

```
var jqLinks = $('a[href!="www.google.com"]');
```

- **Select all input elements with name attribute ending with NAME**

```
var jqInputs = $('input[name$="NAME"]');
```

- **Select all input elements with name attribute starting with PAYOR**

```
var jqInputs = $('input[name^="PAYOR"]');
```



## Accessing Elements – Tree Traversal

- **Select all descendants that match within given jQuery object**

```
var jqInputs = $('#mydiv').find('input[name="NAME"]');
```

- **Select all direct descendants that match within given jQuery object**

```
var jqInputs = $('#mydiv').children('input[name="NAME"]');
```

- Note that the difference between `find()` and `children()` is that –
  - `children()` traverses down only a single level
  - in `children()` the selector is optional



## Accessing Elements – Tree Traversal

- **Select 3rd 'li' that match within given jQuery object**

```
var input=$( "li" ).eq( 2 ).css( "background-color", "red" );
```

- **The .map() method is particularly useful for getting or setting the value of a collection of elements and .get() on the result to work with a basic array:**

```
$( ":checkbox" )  
  .map(function() {  
    return this.id;  
  })  
  .get()  
  .join();
```







# Using jQuery for modifying styles >>>



## Changing Elements

- **To get / set html - `html()`**

```
var html = $('#mydiv').html();  
$('#mydiv').html(html);
```

- **To get / set input values – `val()`**

```
var value = $('#myinput').val();  
$('#myinput').val (value);
```

- **To get / set css properties – `css()`**

```
var fontcolor = $('#mydiv').css('color');  
$('#mydiv').css('color', fontcolor);
```



## Changing Elements

- To add a css class – **addClass()**

```
$('#row').addClass('even');
```

- To remove a css class – **removeClass()**

```
$('#row').removeClass('even');
```

- To hide elements – **hide()**

```
$('#mydiv').hide();
```

- To show elements – **show()**

```
$('#mydiv').show();
```



## Changing Elements

- **To get / set width – width()**  

```
var width = $('#mydiv').width();  
$('#mydiv').width(width);
```
- **To get / set height – height()**  

```
var height = $('#mydiv').height();  
$('#mydiv').height(height);
```
- **To get / set attribute – attr()**  

```
var titleMessage = $('#myinput').attr('title');  
$('#myinput').attr('title', titleMessage);
```



# Using jQuery for manipulating DOM >>>



## Changing Elements

- **To append content in selected element – `append()`**  
`$('#mydiv').append('<span>Hello World!</span>');`
- **To prepend content in selected element - `prepend()`**  
`$('#mydiv').prepend('<span>Hello World!</span>');`
- **To add content before selected element – `before()`**  
`$('#mydiv').before('<span>Hello World!</span>');`
- **To add content after selected element – `after()`**  
`$('#mydiv').after('<span>Hello World!</span>');`



# Using jQuery for Event Handling>>>



# jQuery Event Shortcut Functions



- **jQuery event shortcuts:**
  - `click()`
  - `blur()`
  - `focus()`
  - `dblclick()`
  - `mousedown()`
  - `mouseup()`
  - `mouseover()`
  - `keydown()`,
  - `keypress()`
  - See more at <http://api.jquery.com/category/events>



## Events - .on() and .trigger()

- **To attach event listeners – on()**  

```
$('#dataTable tbody tr').on( 'click', function() {  
    console.log('Click triggered on row');  
});
```
- **To attach delegated event listeners – on()**  

```
$('#dataTable tbody').on( 'click', 'tr', function() {  
    console.log('Click triggered on row');  
});
```
- **To trigger an event – trigger()**  

```
$('#tr').trigger('click');
```



## .on('load') vs .ready()

- To check if the DOM has loaded - **.ready()**

```
$(document).ready(function() {  
    // The DOM is ready!  
});
```

```
$(function() {  
    // or use a shorter version...  
});
```

- To check if the page has fully loaded - **.on('load')**

```
$(window).on('load', function() {  
    // Do something ... after images, subframes etc. are loaded.  
});
```

## jQuery Method Chaining



- It makes your code short and easy to manage.
- It gives better performance.
- The chain starts from left to right. So left most will be called first and so on.

```
$(document).ready(function(){  
    $("button").click(function(){  
        $("p").animate({width:  
"100%"}).animate({fontSize:  
"46px"}).animate({borderWidth: 30});  
    });  
});
```



**Thank You!**