Introduction to Jquery

Agenda

What is jQuery ?

Why use jQuery ?

How to use ?

DOM - Document Object Model

jQuery *Refresher!*

Events

Ajax

Utility Functions

Code Organization

Exercises

What is

jQuery ?

What is jQuery ?

jQuery is a lightweight, "write less, do more", JavaScript library.

The purpose of jQuery is to make it much easier to use JavaScript on your website.

Why use

jQuery ?

Why use jQuery ?

- Write less, do more:
  - *$("p.neat").addClass("ohmy").show("slow");*
- Performance
- Plugins
- It’s standard
- … and fun!

How to use

jQuery ?

Add jQuery library

```
<html>
    <head>
        <title> jQuery Bootcamp </title>
    </head>
    <body>
    <!-- content goes here -->
    <script
        type=”text/javascript”
        src=”https://ajax.googleapis.com/ajax/libs/jquery/3.1.1/jquery.min.js”>
    </script>
    </body>
</html>
```

How to use ?

External link

```
<script type=”text/javascript” src=”../”></script>
```

Embedded

```
<script type=”text/javascript”>
//write your code here
</script>
```

DOM

What is DOM ?

DOM is basically a tree of elements we see on our pages

How we *traverse* and search through that tree directly affects **performance**

jQuery is majorly focused towards DOM *manipulation*

DOM Ready

```
$(document).ready(function() {
    // safe to use here
});
```

OR

```
$(function() {
    // shorthand version
});
```

jQuery - selectors

```
$("div")           // select by tag name
$(".my-class")     // select by class name
$("#container")    // select by element ID
$("[type=text]")   // select by attribute
```

In short, all the **CSS** selectors work here too ;)

jQuery - selectors

```
$("div > span")    // child selector
$("div span")      // descendant selector
$("p ~ div")       // Next siblings
$("[name$=val]")   // ends with
$("[name^=val]")   // begins with
$("[name*=val]")   // contains
```

jQuery - selectors

```
$("div:first")  // first selected element
$("div:last")  // last selected element
$("div:even")  // even elements
$("div:odd")   // odd elements
```

Traversal

Traversal

first, last, next, prev

parents, parent

Children

Siblings

Find

Element Selector

- :text <input type="text" />
- :password <input type="password" />
- :radio <input type="radio" />
- :checkbox <input type="checkbox" />
- :submit <input type="submit" />
- :image <input type="image" />
- :button <input type="button" />
- :file <input type="file" />
- :hidden <input type="hidden" />

Manipulation

Manipulation

Manipulating `DOM` comprises of *changing* the existing *structure* of `DOM` tree

`Adding`, `removing`, `moving` nodes from the `DOM`

jQuery makes is very easy to add and remove nodes from a page

Manipulation: Adding Nodes

It's extremely easy to add new nodes to our DOM

```
var el = $("<p></p>", {
   id: "abc",
   class: "myClass"
});
```

Manipulation: Adding Nodes

We can then easily insert this node several ways

```
$("body").append/prepend(el);
$("#section").html(el);
$("body").before/after(el);
el.appendTo/prependTo($("body"));
el.insertAfter/insertBefore($("body"));
```

Manipulation: Adding Nodes

Changing contents

.html(), .html(val);

.text(), .text(val);

.css('property','value') or .css({'property1':'value1','property2':'value2'})

.attr('data-id',”5”) //.prop in new version of jQuery

.addClass() .removeClass() .toggleClass()

Exercise :

Create an anchor tag using jQuery and add href attribute and assign it a value http://www.google.com

Manipulation: Deleting Nodes

It's extremely easy to delete nodes from our DOM

```
$("p.testClass").remove();
$("p.testClass").empty();
$("p.testClass").unWrap();
var el = $("p.testClass").detach();
```

Manipulation: Replacing Nodes

It's extremely easy to delete nodes from our DOM

```
$("p.testClass").replaceWith(el);
```

```
$("<p>Hey</p>").replaceAll("p.testClass");
```

To avoid the confusion. $(' Hey ') is not the selector but it will be the output of .replaceAll('p.testClass'). Elements can not be selected this way i.e Hey

Manipulation: Replacing Nodes

It's extremely easy to copy nodes in our DOM

```
var cloning = $("p.testClass").clone();
```

Please note .clone() would not work when used alone. It needs any variable to store result or append/prepend to traverse it in DOM.

Methods: HTML/CSS Methods

addClass()

```
<script>
$(document).ready(function(){
    $("button").click(function(){
        $("p:first").addClass("intro");
    });
});
</script>
```

Methods: HTML/CSS Methods

removeClass()

```
<script>
$(document).ready(function(){
    $("button").click(function(){
        $("p").removeClass("intro");
    });
});
</script>
```

Methods: HTML/CSS Methods

toggleClass()

```
<script>
$(document).ready(function(){
    $("button").click(function(){
        $("p").toggleClass("main");
    });
});
</script>
```

Methods: HTML/CSS Methods

width()

```
<script>
$(document).ready(function(){
    $("button").click(function(){
        alert("Width of div: " + $("div").width());
    });
});
</script>
```

Methods: HTML/CSS Methods

height()

```
<script>
$(document).ready(function(){
    $("button").click(function(){
        alert("Height of div: " + $("div").height());
    });
});
</script>
```

Methods: HTML/CSS Methods

css()

```
<script>
$(document).ready(function(){
    $("button").click(function(){
        $("p").css("color", "red");
    });
});
</script>
```

Methods: HTML/CSS Methods

position()

```
<script>
$(document).ready(function(){
    $("button").click(function(){
        var x = $("p").position();
        alert("Top position: " + x.top + " Left position: " + x.left);
    });
});
</script>
```

Methods: HTML/CSS Methods

Offset()

```
<script>
$(document).ready(function(){
    $("button").click(function(){
        var x = $("p").offset();
        alert("Top: " + x.top + " Left: " + x.left);
    });
});
</script>
```

Methods: HTML/CSS Methods

.attr()

```
<script>
$(document).ready(function(){
    $("button").click(function(){
        $("img").attr("width", "500");
    });
});
</script>
```

Methods: HTML/CSS Methods

val()

```
<script>
$(document).ready(function(){
    $("button").click(function(){
        $("input:text").val("Glenn Quagmire");
    });
});
</script>
```

Methods: HTML/CSS Methods

animate()

```
<script>
$(document).ready(function(){
    $("button").click(function(){
        $("div").animate({left: '250px'});
    });
});
</script>
```

Methods: HTML/CSS Methods

animate()

```
<script>
$(document).ready(function(){
    $("button").click(function(){
        $("div").animate({left: '250px'});
    });
});
</script>
```

What's up with

Events

Events - Basics

Event methods trigger or attach a function to an event handler for the selected elements.

jQuery Events

```
$("selector").on("event", function(e) {
    // event happened
});

$("body").on("click", function(e) {
    // body was clicked
});
```

Defaults / propagation

```
$("body").on("click", function(e) {
    e.preventDefault();
    // default action of event stops
    e.stopPropagation();
    // event stops bubbling up
});
```

$: Events

- .hover();
- .toggle();
- .focus();
- .change();
- .blur();
- .click()
- .dblClick();
- .keyDown();
- .keyUp();
- .keyPress();
- mouseover();
- mousenter();

AJAX

AJAX

AJAX stands for Asynchronous JavaScript and XML. It can send as well as receive information in a variety of formats, including JSON, XML, HTML, and even text files. AJAX’s most appealing characteristic, however, is its "asynchronous" nature, which means it can do all of this without having to refresh the page.

AJAX SYNTAX

$.ajax(*{name:value, name:value, ... }*);

```
// working example
$(document).ready(function(){
   $("button").click(function() {
      // data path
       $.ajax({url: "demo_ajax_load.txt", async: false, success: function(result) {
      // get Result
      $("div").html(result);
       }});
   });
});
```

UTILITY FUNCTIONS

Utility Functions

JQuery provides several utilities in the format of $(name space). These methods are helpful to complete the programming tasks.

Utility Functions

- $.trim();
- $.each();
- $.map();
- $.inArray();
- $.extend();
- $.contains();
- $.data();
- $.isWindow();
- $.now();

CODE ORGANIZATION

Code Organization

The essence of building applications is understanding how to organize code so that it is navigable and well-encapsulated instead of a whole slew of global functions.

Code Organization

- Always include scripts at the bottom of the page.
- Divided into units of functionality.
- Always use latest and stable version of jquery.
- Minified version of javascript in production and also minified other scripts.
- Use of For instead jquery each.
- Use IDs instead of class selector.
- Give your Selectors a Context
  $(expression, context)
- Avoid Dom manipulation
  use html() instead of prepend(), append(), after()
- Make Use of HTML5 Data Attributes/jquery data()
- Line breaks and chainability

Exercise :

Create a simple button and one(or two?) paragraph with text “Hi”. Once the button is clicked, the paragraph text should change to “Hello”. If button is clicked once again, the paragraph text should again change to “Hi”. This process must repeat on subsequent clicks.

Show text value of Textbox in a tag when user press ENter key.

Thank you!!