Awarding Great British Qualifications

Dynamic Websites

Topic 7:

Using Scripts (2)

Scope and Coverage

This topic will cover:

- Making use of jQuery to enhance their front ends;
- Selecting elements using jQuery selectors and filters;
- Manipulate and animate HTML elements through jQuery.

Learning Outcomes

By the end of this topic students will be able to:

- An introduction to jQuery
- Effects in jQuery
- Selectors and Filters
- HTML manipulation with jQuery
- jQuery for mobile devices
- JSON

Introduction

- In this lecture we look further into jQuery which is a library for JavaScript which simplifies and extends what it can do.
- jQuery is most popular JavaScript library in use on the Internet.
- It is essentially a huge JavaScript program that someone has written for use.
- There are several ways to make it available to a web page and for mobile devices.

jQuery

- The best way is to download and install the library into a local directory.
- You can also use an external repository
  - With the understanding that this may slow down you applications because of the extra latency involved.
- You may choose which of these strategies to use yourself.
  - For this lecture, we will assume you are using an external repository.

The Structure of jQuery

- jQuery provides an extension to the standard DOM model of JavaScript.
  - In technical terms, it is a ***wrapper***. It embraces and adapts the functionality of DOM.
- jQuery provides for greater expressiveness when coding.
  - The code statements you write can be made to do a lot more than they would otherwise in plain JavaScript.
- All of this functionality is accessed through the jQuery API.

A First jQuery Example - 1

- We place jQuery code in an event handler inside script tags. The event handler we use is called ***ready***.
- This handler is triggered after the DOM is loaded, but before the page contents are placed on the document.
- We added the ready handler to our document, using the special jQuery notation.

A First jQuery Example - 2

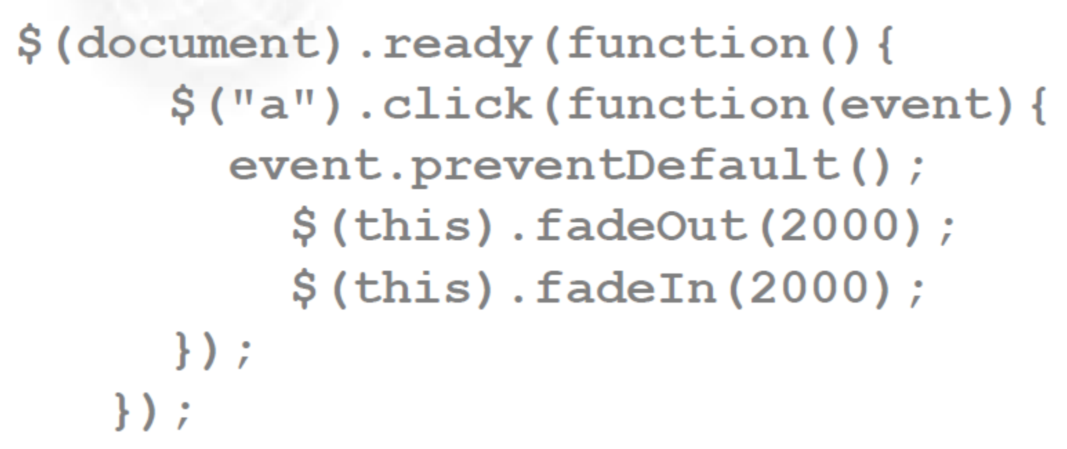
```
<html>
  <head>
    <title>jQuery Demo</title>
  </head>
  <body>
    <a href="http://www.nccedu.com/">NCC Education</a>
    <script src="http://code.jquery.com/jquery-1.6.1.js"></script>
    <script>

    $(document).ready(function(){
      $("a").click(function(event){
        alert("Thanks for clicking that link!");
      });
    });

    </script>
  </body>
</html>
```

jQuery Presentation Flourishes

- One of the other things provided by the jQuery library are a series of presentation flourishes.
  - These are known as ***effects***.
  - Much as you often see in Powerpoint Presentations.
- We can make elements slowly appear, slowly disappear, animate and more.
  - As with any kind of presentational flourish, we should be wary of over-using the effect.
- If we wanted to stop the link working, then fade it out and in, we could.

Fade In, Fade Out

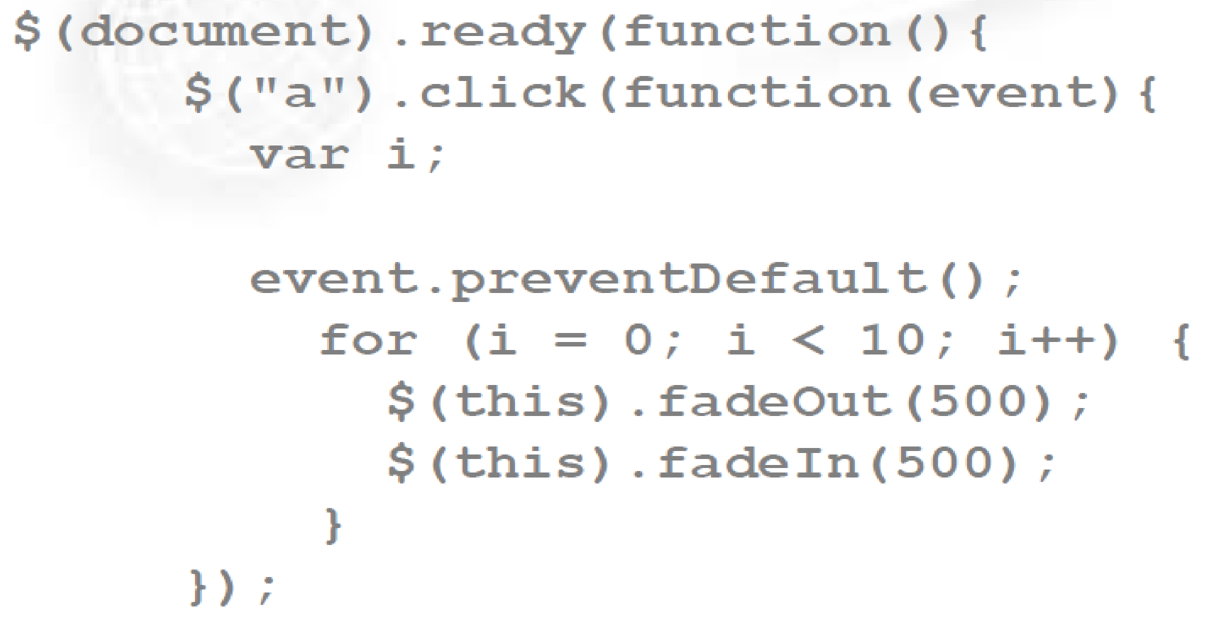
- preventDefault stops the default interaction with the element (in this case, it stops us navigating to the destination).



```
$(document).ready(function(){
     $("a").click(function(event){
       event.preventDefault();
         $(this).fadeOut(2000);
         $(this).fadeIn(2000);
     });
   });
```

jQuery is JavaScript

- jQuery is a library of JavaScript, which means we can use normal code within jQuery functions too:



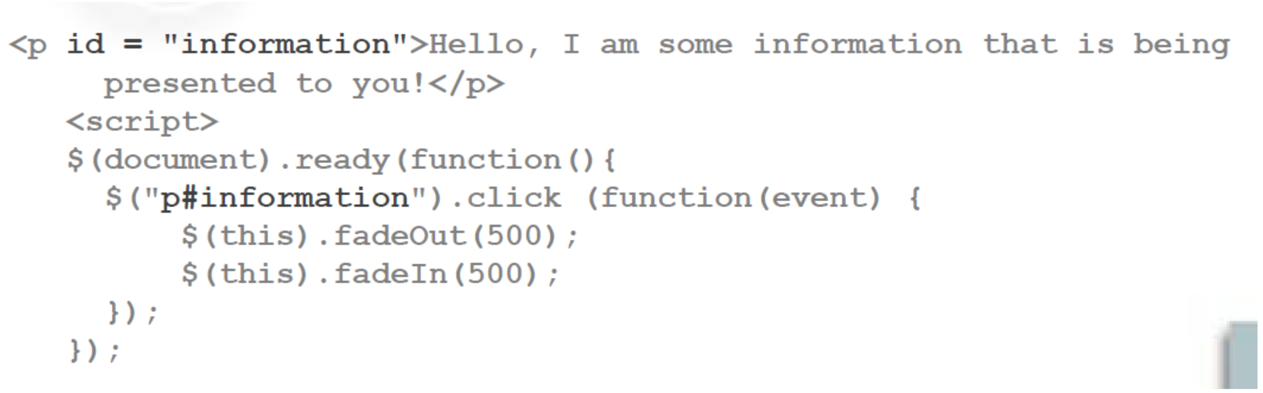
```
$(document).ready(function(){
    $("a").click(function(event){
      var i;

      event.preventDefault();
        for (i = 0; i < 10; i++) {
          $(this).fadeOut(500);
          $(this).fadeIn(500);
        }
    });
```

Finding Elements with jQuery - 1

- This $ notation indicates that we want to make use of jQuery – what follows is the element of which we wish to make use.
    - $(document) means “access the document element via jQuery”).
- If we want to get an element with a specific tag, we use the $(tag) syntax, such as $(a).
    - Get all anchor tags
- We can also get more specific.

Finding Elements with jQuery - 2

- jQuery permits us to match with class/name attributes with more ease than PHP or JavaScript.
  - The following example gets all <p> elements with the ID of “information”



```
<p id = "information">Hello, I am some information that is being
     presented to you!</p>
   <script>
   $(document).ready(function(){
     $("p#information").click (function(event) {
         $(this).fadeOut(500);
         $(this).fadeIn(500);
     });
   });
```

Selector - 1

- jQuery offers a range of selectors that allow us to select elements with either great precision or through broad criteria.
  - P#information is an example of a selector.
- The # symbol allows us to get an element by ID.
  - It does not need a tag, but this can be used to specialise the search.
- Classes (such as those defined in CSS) are indicated by a . Symbol.

Selectors - 2

- For classes, we can indicate our desire to match those that have multiple matches by chaining the Symbols:
  - $(“.person.student”) will get any elements that have both the person and student classes applied to them.
- We can also select multiple elements by using a comma separated list:
  - $(“.person,#thingy”) will get any element that have had the CSS class “person” applied, as well as any with an ID of “thingy”

Filters - 1

- jQuery also offers us a powerful mechanism for finding and manipulating elements.
  - These are ***filters.***
- We indicate our wish to use these by following the selector with a colon and the filter to use.
  - :odd and :even will get all the odd and even items that match a tag.
  - :header will filter out any elements that are not indicated by header tags (h1, h2, h3 etc).

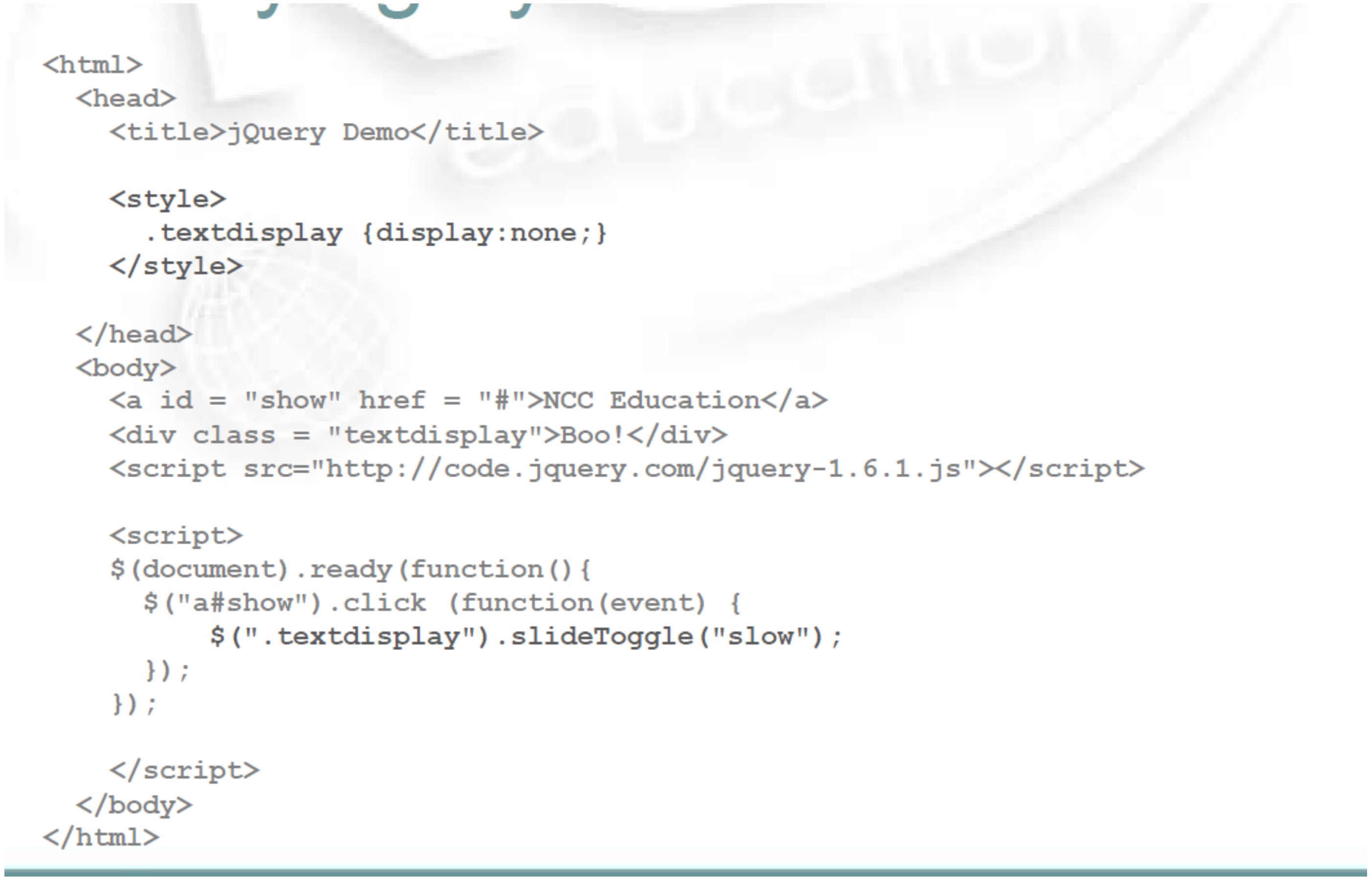
Filters - 2

- There are around 20 different filters defined in the jQuery library.
  - We do not have time to cover all of them.
- You can use them to achieve very sophisticated ends in matching elements.
  - $(“#things,.stuff:not”) will get all the things that are ***not*** matched by the selector.
  - $(“things,.stuff:has(b)”) will get anything with the ID “things” or the class “stuff”, provided that it contains somewhere within its innerHTML a<b> tag.

jQuery Events

- We can bind handlers to a wide range of events on a wide range of elements.
  - These map onto JavaScript events for the main part.
- Some of the common events we want to trap:
  - Mouseover
  - Click
  - Load
  - Change
  - Blur
  - However

Modifying by Class



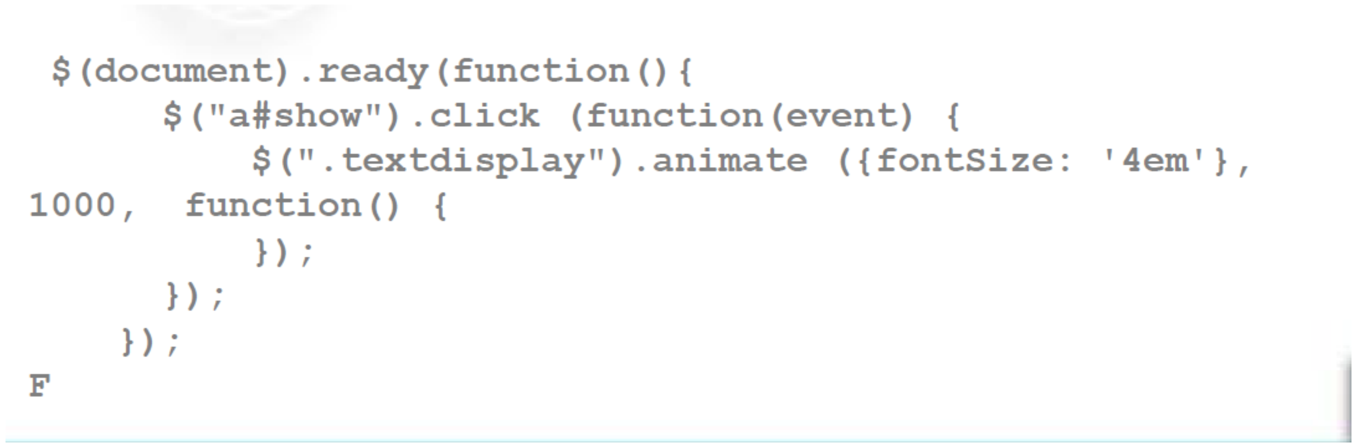
```
<html>
  <head>
    <title>jQuery Demo</title>

    <style>
      .textdisplay {display:none;}
    </style>

  </head>
  <body>
    <a id = "show" href = "#">NCC Education</a>
    <div class = "textdisplay">Boo!</div>
    <script src="http://code.jquery.com/jquery-1.6.1.js"></script>

    <script>
    $(document).ready(function(){
      $("a#show").click (function(event) {
          $(".textdisplay").slideToggle("slow");
      });
    });

    </script>
  </body>
</html>
```

Animation

- We are not limited to the animation provided by the default methods.
  - jQuery permits us to modify any arbitrary CSS definition using the animate method.
- We provide the desired end point of the value, a duration, and a callback function.
  - Longer durations indicate slower animations.
- The callback function is executed at the end of the process.

Example Animations - 1

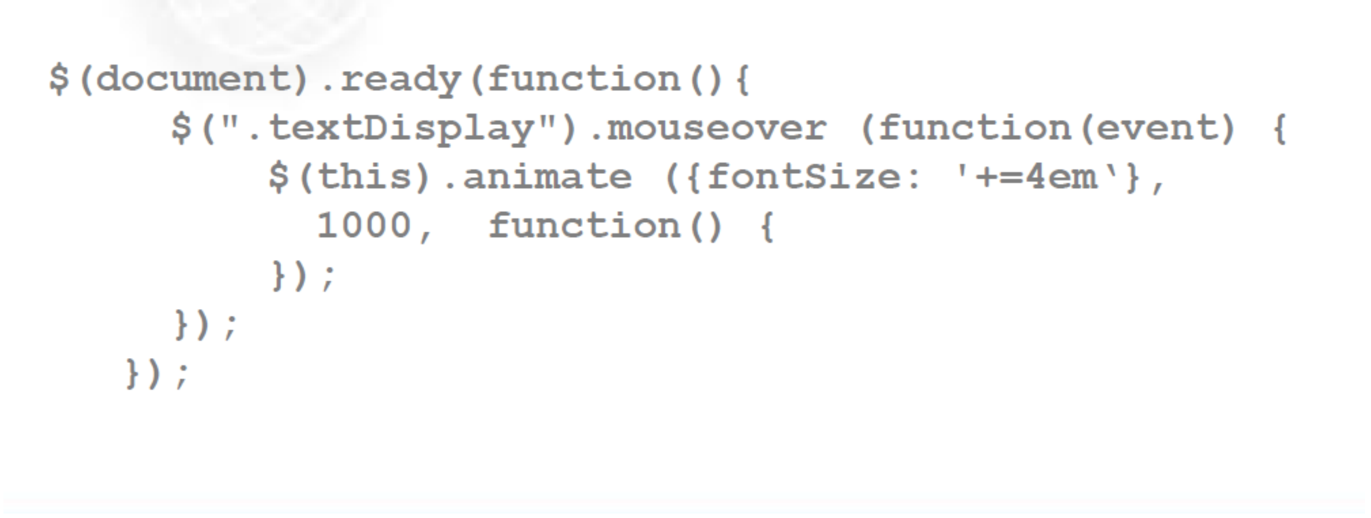
- We can have an element animate to a particular state, or to one that is relative.
- The below example shows animating an element so that its font size is changed to a maximum:



```
 $(document).ready(function(){
      $("a#show").click (function(event) {
          $(".textdisplay").animate ({fontSize: '4em'},
1000,  function() {
          });
      });
    });
F
```

Example Animations - 2

- This function shows an animation that increases the font size every time the function is triggered via a relative adjustment:

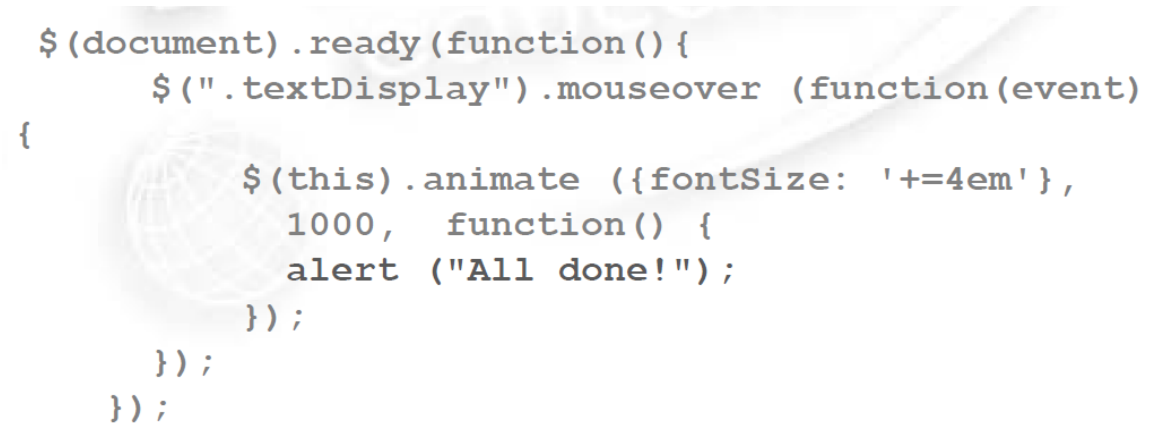


```
$(document).ready(function(){
     $(".textDisplay").mouseover (function(event) {
          $(this).animate ({fontSize: '+=4em'},
            1000,  function() {
          });
     });
   });
```

Callbacks

- Most jQuery effects permit you to define a callback function to be called when the animation is complete.
  - They are normally provided as a third parameter after the duration.
- JavaScript executes statements line by line.
  - Without using a callback, code following the effect will be executed before the animation completes.
  - We can see a callback stub in the calls made to the animate function.

Animation with Callbacks



```
 $(document).ready(function(){
      $(".textDisplay").mouseover (function(event)
{
           $(this).animate ({fontSize: '+=4em'},
             1000,  function() {
             alert ("All done!");
           });
      });
    });
```

- Callbacks allow us to be sure that our animations sync up correctly with anything dependant on them.

Manipulating HTML via jQuery

- As with JavaScript, we can change the innerHTML of an element, this time using the html method.
  - When used with no parameters, it returns the current innerHTML.
  - When used with a string parameter, it replaces it.
- Accessing elements for this is done in the same ways as for other methods:
  - $(“.textdisplay”).html (“Bing”);

Adding Attributes

- jQuery even lets us change the attributes on elements.
  - This is done using the attr function.
- While attributes are best avoided in XML, they are used constantly in HTML.
  - And it is an extremely powerful technique to alter them using jQuery.
- We could easily add alt tags to every image that lacks them, by using jQuery.
  - $(“img:not([alt])”.attr(“alt”, “An image of some kind.”);

Adding CSS

- We can add CSS “on the fly” to elements using jQuery.
  - This is done using the css function.
- The code below shows how to “zebra-stripe” a table:

```
$(document).ready(function(){
     $("tr:odd").css('background', "#abcabc");
     $("tr:even").css('background', "#cbacba");
   });
```

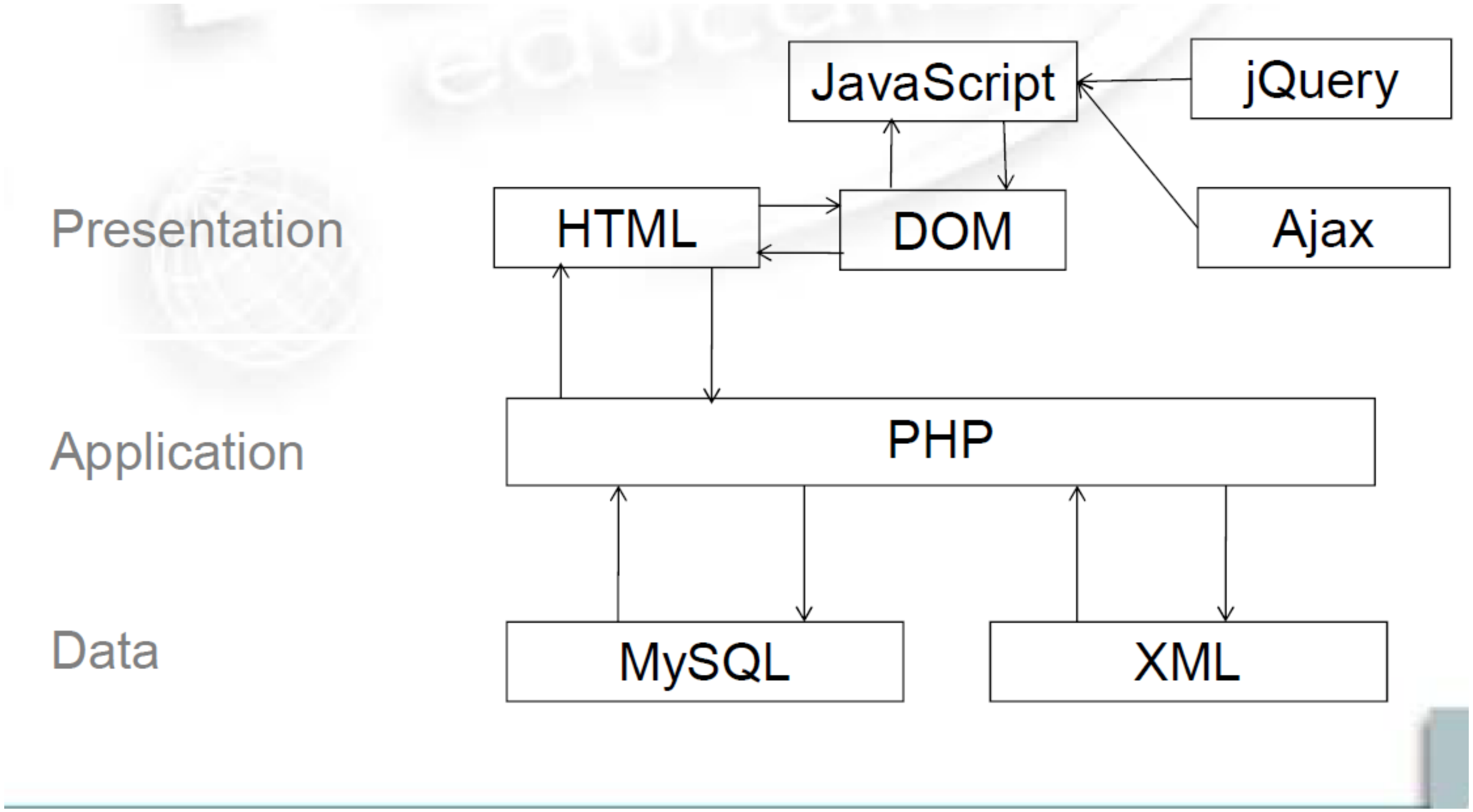
jQuery and JavaScript

- All of this is very neat of course, but beyond simplifying coding, what is the benefit?
  - It turns out, the benefit is considerable!
- jQuery is a ***cross browser*** library.
  - Ajax requests are dependent on browsers.
  - jQuery handles most of the compatibility problems within its libraries.
    - We write the jQuery code, and the library makes it work on different browsers.

Benefits of jQuery

- There are other benefits too:
  - Ease of element selection and manipulation
  - Ease of adding presentational flourishes
  - Simplified event handling
  - Small footprint
  - Support plug-ins
- However, as with all frameworks of this nature, the danger is that you end up relying on it.
  - You should always know how to accomplish the same goal in plan JavaScript.

Our Architectures So Far



jQuery for Mobile Devices

- jQuery Mobile can be used to create mobile web applications.
- jQuery Mobile works on most popular smartphones and tablets and is viewed best through GoogleChrome.
- You should add a style sheet and libraries to each page.

Example jquery for mobile

```
<!DOCTYPE html>
<html>
<head>
<meta name="viewport" content="width=device-width, initial-scale=1">
<link rel="stylesheet" href="http://code.jquery.com/mobile/1.4.2/jquery.mobile-1.4.2.min.css">
<script src="http://code.jquery.com/jquery-1.10.2.min.js"></script>
<script src="http://code.jquery.com/mobile/1.4.2/jquery.mobile-1.4.2.min.js"></script>
</head>
<body>
<div data-role="page">
  <div data-role="header">
    <h1>Welcome to Durham Zoo</h1>
  </div>
   <div data-role="main" class="ui-content">
    <p>We are open 362 days a year.  We are only closed Christmas Day and Easter Sunday.</p>
  </div>
   <div data-role="footer">
    <h1>contact:  durhamzoo@yahoo.co.uk</h1>
  </div>
</div>
</body>
</html>
```

JSON

- JavaScript Object Notation (JSON) stores and exchanges data.
- JSON is text, written with JavaScript object notation.
- JSON is independent to any programming language.
- XML is set of rules for encoding documents into machine-readable form.

JSON or XML

- XML items are written in open and close tags
  - JSON you name the tags once
- JSON can bypass the XMLHttpRequest object when getting data.
- JSON is easier to read than XML
- AJAX includes XML, whereas JSON does not.

Example JSON –v- XML

JSON Example

```
{"employees":[
   { "firstName": "Sarah", "lastName": "Hamilton" },
   { "firstName":"Steven", "lastName": "Vettel" },
   { "firstName":"James", "lastName": "Palmer" }
]}N
```

XML

```
<employees>
   <employee>
      <firstName>Sarah</firstName> <lastName>Hamilton</lastName>
   </employee>
   <employee>
      <firstName>Steven</firstName> <lastName>Vettel</lastName>
   </employee>
   <employee>
      <firstName>James</firstName> <lastName>Palmer</lastName>
   </employee>
</employees>
```

Conclusion

- Dynamic and engaging web front ends are possible through the use of Ajax and JavaScript.
  - However, they can be made easier for us to do through the use of jQuery.
- jQuery is the most popular JavaScript library in use on the Internet at the moment.
  - Its popularity is largely due to the extremely high quality of the library itself.
- jQuery extends our ability to create our user interfaces.

Terminology

- *jQuery* – library of JavaScripts.
- *JSON* – JavaScript Object Notation.
- *Callback* - allows us to be sure that our animations sync up correctly with anything dependant on them.

References

- Code.query.com, 2017. [online] Available at https://code.query.com/mobile
- W3schools.com, 2017. [online] Available at www.w3schools.com

Awarding Great British Qualifications

Topic 7 - jQuery

Any Questions?