

jQuery

Peter J Barclay

with material from Andrew Cumming

Write less, do more...

<http://40011071.soc-web-liv-11.napier.ac.uk/engweek>

Lecture Plan

- jQuery – what and why
- The ready function
- Creating Content
- Changing Elements
- Applying CSS
- Applying functions
- Lists of useful JavaScript and jQuery functions
- Arrays and Objects in JavaScript

jQuery – What and Why

- Write less, do more
- jQuery is a JavaScript library
- It makes writing JavaScript easier
- It smooth out some browser incompatibilities
- It is one of the most successful libraries

JavaScript is unavoidable

- HTML and CSS and JavaScript (ECMA Script) are built into the browsers and built into the W3C specifications. They cannot be avoided.
- Unfortunately JavaScript has been implemented rather differently in different browsers.
- The programmer has to write different code for each browser (IE6, IE8, Firefox, Opera etc.)
- The jQuery library takes care of these differences.
- jQuery will be compatible with new browsers (probably)

jQuery adds functionality

- jQuery provides additional functions.
- It provides the programmer with a large number of easy to use methods.
- It allows you to write complex scripts in just a few lines.
- It demands an unusual style of programming.

Preliminaries

```
<script  
  src="https://code.jquery.com/jquery.min.js">  
</script>
```

You can download the library from google.

- This means it may be your user's cache.

`$(function(){alert(42)})`

- This is the **ready** function.
- `$` takes a function as a parameter.
- The function is executed when the page has loaded.
- This is a useful time to set up the page

Example

The HTML page – home.html

```
<!doctype html>  
<html>  
<head><title>Timetables</title>  
<script src='https://code.jquery.com/jquery.min.js'> </script>  
<script src='hello.js'></script>  
</head>  
<body>  
<div id='container'>  
...
```

The JavaScript page – hello.js

```
// hello.js  
$(function(){  
    alert("Hello world");  
});
```

The Anonymous Function

- This statement:
 - Creates a function.
 - Sends that function to the ready function.
- The Anonymous Function will execute when the page is ready.

```
$(  
  function(){  
    alert("Hello world");  
  }  
);
```


Creating Content

- You can create content using the **\$** function.
- You can add the content to your page using the **append** method.

```
$(function(){  
  var msg = $('<div>Hello World</div>');  
  $('body').append(msg);  
});
```

Creating Content and Changing it

- After you create a node you can change it:

```
$(function(){
  for (var i=0;i<100;i++){
    var msg = $('<div>Hello World</div>');
    msg.css('font-size',i);
    $('body').append(msg);
  }
});
```



Some Useful jQuery functions

- **\$**
 - Ready function, create elements, find elements
- **append**
 - The append function adds to the end of an element
- **prepend**
 - The **prepend** function adds to the beginning of an element
- **css**
 - The function **css** is used to set a css property of an element
- **html**
 - Change the contents

Creating structures

- You often need to create elements that contain other elements.



```
<div id='countries'>
  <div id='fr'>
    <div>France</div>
    <img src='flags/fr.gif'/>
  </div>
  <div id='fi'>
    <div>Finland</div>
    <img src='flags/fi.gif'/>
  </div>
</div>
```

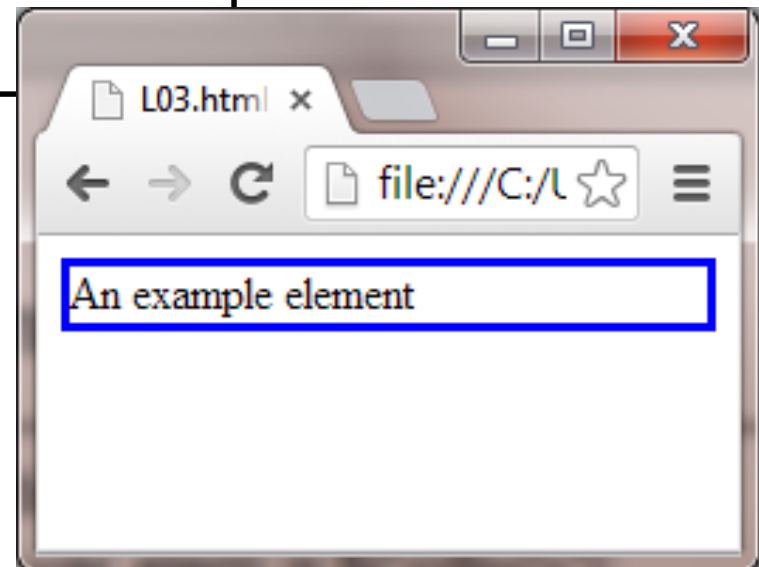
Adding another country

```
var gb = $('<div/>');  
gb.append($('<div/>',{text:'Great Britain'}));  
gb.append($('<img/>',{src:'flags/gb.gif'}));  
$('#countries').append(gb);
```



Examples of use

```
$(function(){  
    var elem = $('<div/>');  
    elem.html('An example element');  
    elem.css('border','solid blue');  
    $('body').append(elem);  
})
```



Self Test

- Select the correct ready function syntax
 - a) `$()(function{alert(42);});`
 - b) `$(function()(alert(42);));`
 - c) `$(function(){alert(42);});`
 - d) `function $({alert(42);});`
 - e) `function $()({alert(42);});`

Self Test

- What is the missing word _____ in this code?

- a) css
- b) append
- c) html
- d) forfend
- e) \$

```
$(document).ready(function(){  
  var d = $('<div>42</div>')  
  d.css('color','red');  
  $('body')._____(d);  
});
```


Self Test

- In the flags example each image has a grey border. Select the best way to achieve this in jQuery
 - a) `$('#countries img',{css:{ border:'solid gray' }});`
 - b) `$('#countries img').{css:{ border:'solid gray' }};`
 - c) `$css('#countries img','border:solid gray');`
 - d) `$('#countries img').css('border','solid gray')`

Selecting existing items

- You can select items with the \$ function
- The result is a list of items
- `$('.li').css('color','red');`
 - Locate all `` elements and make them red.
- `$('.reference').css('background-color','red');`
 - Locate all elements with css class='reference'
- `$('#menu1').css('border','solid thick red');`
 - Locate the element with id "menu1"

Select items with items

- A common pattern is to select elements within a named element:
- `table#main td`
 - Select all `td` elements within the table with id **main**
- `ul#nav li`
 - Select all `li` elements inside the `ul` named **nav**

Selecting Existing Items

- You can modify the selectors
- `$('p:first-child').css('border','solid');`
 - Identifies the first p in each block
- `$('p:nth-child(1)').css('border','solid');`
 - Selects the first child in each block
- `$('p:eq(1)').css('border','solid');`
 - This gets item 1 – the second one
- `$('p:last').css('border','solid');`
 - The last one of all

Combining terms

- As with css you can combine terms.
- When one term follows another separated by a space then the second is a descendant of the first
 - table tr td
 - div#content p
- A > indicates a direct descendant, a child
 - ul.menu>li prevents nested li elements matching
- A comma allows an independent rule - OR

Data Structures

```
$(function(){  
  var ca = {  
    cname:"A J CUMMING",  
    address:{  
      street:"10 Colinton Rd",  
      town:"Edinburgh",  
      post_code:"EH14 5DT"  
    },  
    transaction:[  
      {whn:"2014-01-11",nar:"Cash Withdrawal",amt:-100},  
      {whn:"2014-01-11",nar:"BUS FARE",amt:-35}  
    ]  
  };  
  alert(ca.transaction[1].nar);  
})
```

What expression would give each of the following:

“Edinburgh”
-100
“EH14 5DT”

ca.transaction[1].nar
This code gives “BUS FARE”



Data Structures used to create elements

- We can set a whole load of properties when an element is created.
- In this example we set the **text** property and the **css** property.
- The **css** property is itself a complex object

```
var d = $('<div/>', {text:'Hello',css:{color:'green',border:'solid'}});
```

More to learn

- You can navigate through the list
- You can animate items
- You can show, hide and toggle
- You can apply functions to all items in the list
- You can store data against elements and retrieve it.
- You can use ajax functions
- You can create other effects using add-ons

Summary

- jQuery – what and why
- The ready function
- Creating Content
- Changing Elements
- Applying CSS
- Applying functions
- Lists of useful JavaScript and jQuery functions
- Arrays and Objects in JavaScript