jQuery Tutorial 1

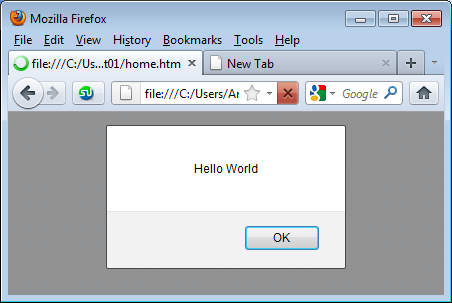
## 1) Hello World

You will need to find the file jquery-1.4.1.min.js and place it in a folder.

Create two files yourself; home.html and tut01.js – these have the following content:

|  |  |
| --- | --- |
| home.html | tut01.js |
| <!DOCTYPE html>  <html>  <head>  <script src='jquery-1.4.1.min.js' type='text/javascript'></script>  <script src='tut01.js' type='text/javascript'></script>  </head>  <body>  </body>  </html> | $(document).ready(function(){  alert("Hello World");  }); |

Load home.html into a web browser (ideally Firefox with the FireBug extension) and view the results:



## 2) Adding Content

The ready function is called once the main page elements are loaded. You can build content in the ready function or you can examine the existing page when ready:

|  |
| --- |
| $('body').append($('<h1/>',{text:'Hello world'})); |

Or alternatively:

|  |
| --- |
| $('body').append($('<h2>This is grand</h2>')); |

## 3) Programming some content

The following example generates film titles in a list.

|  |  |
| --- | --- |
| $(document).ready(function(){  var ls = [["Indecent","Proposal"],  ["Beautiful","Creatures"],  ["Dangerous","Liasons"]];  for (var i=0;i<ls.length;i++){  for (var j=0;j<ls.length;j++){  $('body').append($('<div/>',  {text:ls[i][0]+" "+ls[j][1]}));  }  }  }); |  |

Change the code so that is generates an <li> list inside a <ul> element. You will need to create the <ul> element before the loop starts and add each <li> element to it.

|  |  |
| --- | --- |
| $(document).ready(function(){  var ls = [["Indecent","Proposal"],  ["Beautiful","Creatures"],  ["Dangerous","Liasons"]];  **var u = $('<ul/>');**  for (var i=0;i<ls.length;i++){  for (var j=0;j<ls.length;j++){  **u.append($('<li/>',**  {text:ls[i][0]+" "+ls[j][1]}));  }  }  **$('body').append(u);**  }); |  |

Change the code to generate a table. You will need to create a table element, several <tr> elements and several <td> elements.

|  |  |
| --- | --- |
| Use  **var t = $('<table border="1"/>');**  to create the table. Notice that you must use different quotes. |  |

## 4) Setting properties of elements

You can set various properties of your new elements. The properties are put in a hash map in curly braces {}. You have seen the **text** attribute being set; you can also set the **cssClass** the **style** and the **id** and pretty much all other html attributes.

Setting style – you can set css style using a string or a hash map.

|  |  |
| --- | --- |
| Set style string | Set css hash map |
| var p=$('<p/>',{text:'Hello World',  style:'border:solid;font-family:sans-serif'}); | var p=$('<p/>',{text:'Hello World',  css:{border:'solid',fontFamily:'sans-serif'}}); |

You can also set css style after you have created the element:

|  |  |
| --- | --- |
| Set one attribute | Set several attributes |
| var td=$('<td/>',{text:'ABC'});  td.css('border','solid thin'); | var td=$('<td/>',{text:'ABC'});  td.css({color:'red',fontWeight:'bold'}); |

## 5) Creating a grid

|  |  |
| --- | --- |
| $(document).ready(function(){  var t=$('<table/>');  for(var i=0;i<8;i++){  var tr=$('<tr/>');  for (var j=0;j<8;j++){  var td=$('<td/>',{text:i+' '+j});  td.css({border:'solid'});  tr.append(td);  }  t.append(tr);  }  $('body').append(t);  }); |  |

You can use conditions inside the inner loop to create patterns:

|  |  |
| --- | --- |
| if (i==j)  td.css('background-color','red'); | if (j%2==0)  td.css('background-color','red'); |

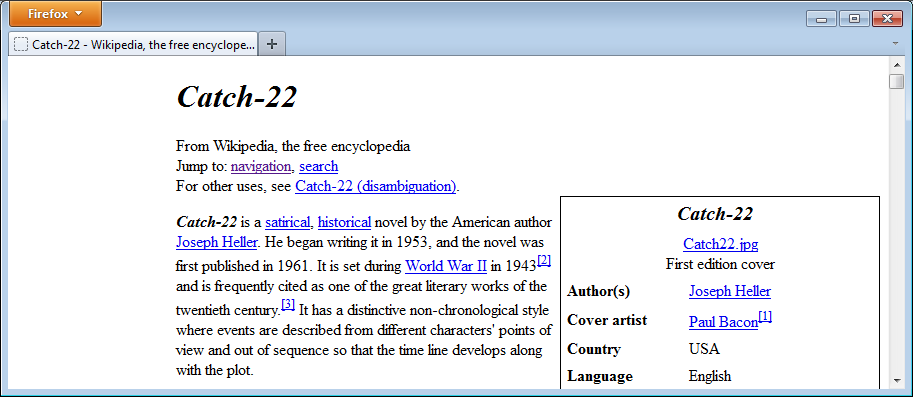
|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |

## 6) Identifying Elements

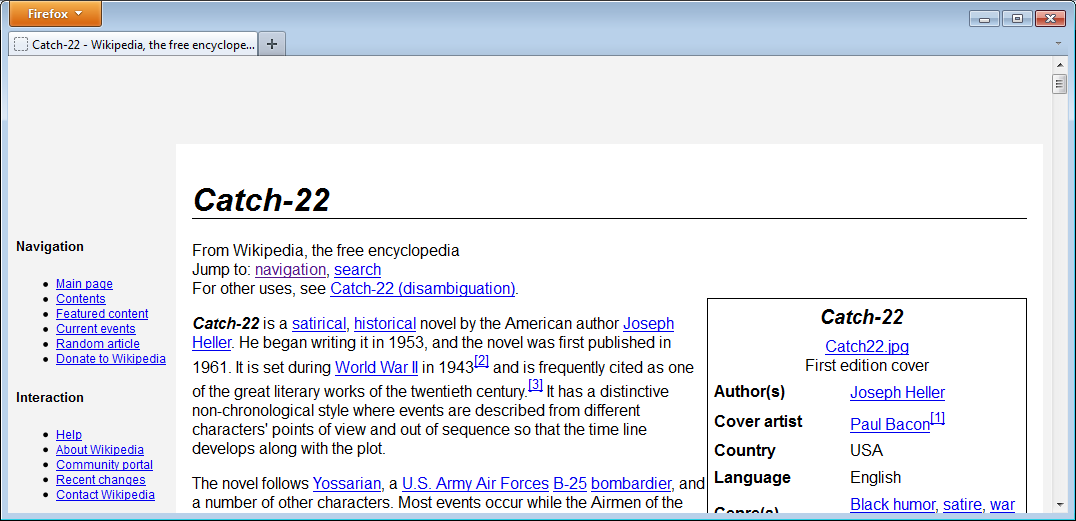
You can select elements using a variety of addressing techniques. The $ function is used again.

Get a copy of the files catch22.html and wiki.js

The file catch22.html is from Wikipedia you do not need to edit it. You will be changing wiki.js to change the appearance of the page.



|  |  |
| --- | --- |
| The box on the right has class **infobox**. The main content is in a div with id **content**. You can float the info box to the right and move the content away from the edge by adding the following lines to wiki.js | $(document).ready(function(){  $('.infobox').css({  'float':'right',  border:'solid thin black'});  $('#content').css({  marginLeft:'10em'  });  }); |

Make more changes to the css to achieve the appearance below. 

Change the **h1** elements to include a bottom border. Make **#mw-panel** position absolute.

* Change the background colour of the body to #F3F3F3.
* Change the background colour of #content to white.
* Change the height of #mw-page-base to 5em – that should move #content down.
* The #mw-panel content is near the end of the document. Bring it to the top by giving it the properties: { width:'10em', position:'absolute', top:'160px' }

## 7) Collapsing menus

The panel on the left has the following structure – we want to make each section collapse and reappear when the user clicks:

|  |  |
| --- | --- |
|  |  |

You can set the page so that the **div.body** is hidden when the page loads. When the user clicks on the **h5** then the body reappears.

|  |
| --- |
| $(document).ready(function(){  $('#p-navigation div.body').hide();  $('#p-navigation h5').click(function(){  $('#p-navigation div.body').show();  });  …  }); |

Try making the following changes:

* Include the parameter ‘slow’ so that there is animation when the items appear.

$('#p-navigation div.body').show('slow');

* Change the word **show** to **toggle** – that means the user can make it go and reappear.
* Do the same for the other elements of the panel
  + p-interaction
  + p-tb
  + p-coll-print\_export
  + p-lang
* Change the css of the h5 so that the ‘cursor’ is ‘pointer’

## Answers

|  |
| --- |
| $(document).ready(function(){  $('#mw-panel div.body').hide();  $('#mw-panel h5').click(function(){  $(this).next('div.body').toggle('slow');  });  }); |