JavaScript and AJAX

JavaScript loops and more Jquery

Huw Davies
daviesah7@cardiff.ac.uk

JavaScript and AJAX

- Sometimes you need to execute the same code many times
  - Perform an operation a fixed number of times
  - Do something with each item in an array, or with each property of an object
    - Iteration

JavaScript and AJAX

The **while** loop

- The most basic type of loop in any programming language
- All other loops are derived from the **while** loop

JavaScript and AJAX

Structure of a **while** loop:

```
var loopvar = somevalue;
while (loopvar condition) {
    … do stuff …
    … update the loopvar
    … possibly other stuff
}
```

JavaScript and AJAX

Key points of the **while** loop:

- You must initialise a loop variable
- You must test the loop variable with a condition that will, at some point, return **false**
- You must update the loop variable in the body of the **while** loop
- Special case: **while** (true) { … }

JavaScript and AJAX

break and **continue**

- **break** jumps out of the enclosing loop
  - jumps to the next statement after the **while** loop
- **continue** starts a new pass through the loop
  - Jumps back top the **while** condition
- Can be used in all types of loop

JavaScript and AJAX

The **do while** loop

- The body of a **while** loop may never be executed
  - if the initial state of the loop variable makes the **while** condition **false** on the first pass
- The **do while** loop tests the condition at the end of the loop
  - so the loop is always executed at least once

JavaScript and AJAX

Structure of the **do while** loop

```
var myVar = someVal;
do {
    … some stuff …
    … update myVar
    … some other stuff …
} while (myVar condition);
```

JavaScript and AJAX

- You can always use a **while** loop
- Only use a **do while** loop if you are sure you want to execute the body of the loop at least once

JavaScript and AJAX

for loops

- More compact syntax than **while** or **do while**

for (startval; endval; increment) {

    … do stuff …

}

- startval is the loop variable
- endval is the condition
- increment is the updating of the loop variable

JavaScript and AJAX

Example of a **for** loop:

```
for ( i = 0; i < 10; i++) {
    … do stuff …
}
```

This is the same as:

```
var i = 0;
while (i < 10) { … do stuff … ; i++; }
```

JavaScript and AJAX

Special types of **for** loop

- **for ... in**
  - Iterates over the enumerable properties of an object
  - Do not use for arrays if index order is important
- **for** (**var** prop **in** obj) { … do stuff … }
- **Note**: in returns the property, not its value

JavaScript and AJAX

for ... of

- Iterates over the enumerable properties of an object, returning the property's value
- Use for arrays and DOM collections

```
arr = ['3', '5', '7', 'foo'];
for (var i of arr) { … do stuff … }
```

JavaScript and AJAX

JQuery selects an element/elements using the syntax: $(…)

The argument uses a CSS-like syntax

JQuery also has a number of special selector attributes…

JavaScript and AJAX

Selectors and filters

Selectors:

```
p
*
#identifier
.class
div.main
p#special
```

JavaScript and AJAX

Selectors and Filters

More selectors

p[lang]=en

[name=fred]

[name^=fred]

[name$=fred]

a[href]

JavaScript and AJAX

Selectors and Filters

Some filters

has(selector) – has a descendent that matches

not(selector) – remove elements from set

eq(index) – select an index from a match set

first(), last()

attr(‘:even’) – Get the value of an attribute for the first element in the set of matched elements or set one or more attributes for every matched element.

JavaScript and AJAX

Selectors and Filters

Special selectors

:even and :odd

:checkbox and :checked

:header

:first :last :first-child :last-child :nth-child(n)

:not

JavaScript and AJAX

JQuery Manipulation

.addClass(‘class’)

.css(‘backgroundColor’, ‘red’)

.html() – get or set the HTML of 1st element in set

.removeAttr() .removeClass() .toggleClass()

.val()

.hasClass(‘class’)

JavaScript and AJAX

- JQuery UI
- User interface widgets for JQuery
  - Calendar
  - Tabs
  - Accordions and much more

JavaScript and AJAX

- JQuery onload functions
- $(document).ready(function(args){…} OR
- $(function(args) {…});

JavaScript and AJAX

- Exercise: add a date picker to a form
  - Start a new web project
  - Add a new HTML page
  - Add a text input in the page body (in a p or div)
  - Add links to jquery and jquery ui
  - Add an onload jquery function that calls the date picker:

    ```
    $("#datepicker").datepicker();
    ```

JavaScript and AJAX

- Exercise 2: Using the documentation
- Change the datepicker date format to something acceptable in Europe

  http://api.jqueryui.com/datepicker/
- What is an altFormat?
- How would you initialize, get and set it?

JavaScript and AJAX

- JQuery UI tabs
- Tabs must be in an ol or ul
- Titles must be in a li and wrapped by an 'a' tag with an href attribute
- Tab content can be in any element, but the element must have an id corresponding to the title 'a' tag

JavaScript and AJAX

- Exercise: Create JQuery UI tabs
- Create new html5 page in your project
- Include JQuery and JQuery UI
- Add a list with 3 items in a div with the id 'tabs'
- Add 3 paragraphs with id tab-1, tab-2, and tab-3
- Link to the paragraphs in the list items
- Add a JQuery onload function that calls tabs() on the 'tabs' element

JavaScript and AJAX

- JQuery UI Accordions
- Convert headers into accordions
- Accordions support arbitrary markup, but each content panel must always be the next sibling after its associated header. See the header option for information on how to use custom markup structures.

JavaScript and AJAX

- Exercise: create an accordion
- Add a new html5 file to your project
- Add required links
- Create a div with id 'accordion'
- Add at least 3 h2 and p pairs in the div
- Add a JQuery onload function that calls accordion() on the div