JavaScript and AJAX

Functions and Objects

Huw Davies
daviesah7@cardiff.ac.uk

JavaScript and AJAX

- **What are functions?**
- Packages of behaviour
- Every time you want that behaviour, you can use the function
- A way of splitting programs up into manageable bits (decomposition)

JavaScript and AJAX

- **Parts of a function:**
- The keyword **function**
- (Optionally) a name for the function
- A list of parameters in parentheses
- The function body in braces, containing (optionally) a return value

JavaScript and AJAX

```
function myFunc(p1, p2) {
    … do stuff …
}

function addThem(toAddArray) {
    var sum = 0;
    for (i = 0; i < toAddArray.length; i++) {
        sum += toAddArray[i];
    }
```

JavaScript and AJAX

- Exercise
- function-exercise.html (See handout)

JavaScript and AJAX

- Functions and the scope of variables
- Variables declared using **var** (which might be implicit), at the top level of a script are global
- The top level means outside of any other JavaScript
- Global variables can be used anywhere
- But beware of “polluting the global namespace”

JavaScript and AJAX

- Variables declared with **var** inside a function are local to that function and replace any global variables with the same name
- Example (var_scope.html)
- Using or not using **var** inside a function is significant
- Using or not using **var** at the top level is not significant

JavaScript and AJAX

- A bit of fun (closure.html)
- It's probably best not to think to much about what's going on here!!!

JavaScript and AJAX

- Built-in functions (e.g. parseFloat and parseInt)
- parseFloat(string) converts the string to a floating point number
- parseInt(string) converts the string to an integer
- NOTE: integers are rounded down
- NOTE: if the string isn't a number, NaN is returned (Not a Number), but some browsers return 0!!

JavaScript and AJAX

- Objects: complex data types
- Properties hold data
- Methods perform actions, like getting and setting values for the data
- JavaScript provides some built-in objects like Math and Date

JavaScript and AJAX

- Demonstration: using the Date object
- dates.html
- Date object documentation:

  www.w3schools.com/jsref/jsref_obj_date.asp

JavaScript and AJAX

- JavaScript objects are class-less and based on prototypes
- By default all objects are based on the Object prototype
- Any constructor can be assigned as the prototype

JavaScript and AJAX

- Two ways to create objects
- Constructor
- Object literal

JavaScript and AJAX

- Using a constructor

```
function HRRecord(empname, dept, age, height) {
    this.empname = empname;
    this.dept = dept;
    this.age = age;
    this.height = height;
}
```

JavaScript and AJAX

- Using an object literal
- Will look in more detail in two week's time
- Have already used them, for example for options passed to jQuery functions

JavaScript and AJAX

- Creating an object instance (for objects created with a constructor):

```
var newEmp = new HRRecord('fred', 'IT', 34, 1.7);

var blankEmp = new HRRecord();
blankEmp.empname = 'fred';
blankEmp.dept = 'IT';
blankEmp.age = 34;
blankEmp.height = 1.7;
```

JavaScript and AJAX

- Demonstrations
- Simple object without methods (hrrecord_1)
- Adding a method the C++ way (hrrecord_2)
- Adding a method the direct way (hrrecord_3)

JavaScript and AJAX

- Using **var** in objects
- Since an object is just a function, if you use **var**, you make the variable invisible outside of the function (object), making the object's attributes private (hrrecord_4)

JavaScript and AJAX

- Exercise:
- Create an object (Handout)