dev8D Javascript workshop

Juliette Culver

While you wait

- Check you have a command-line javascript interpreter e.g. JSDB
- http://www.jsdb.org/download.html or on the memory sticks
- In your javascript interpreter try:
- 1) 0.1 + 0.2
- 2) a = '0'; b = 0; c = '';
- try each of the following: a==b; b==c; a==c;

Objects

1. An object is a collection of name-value pairs ('attributes').

2. A method is an attribute which is a function.

3.You can create an object in two ways (directly or indirectly):

a) Using an object literal (i.e. {...})

b) Using the new keyword and a constructor function

Functions

1. You can create a function with a function literal.
2. Functions are first-class objects.
3. Functions can be anonymous.

Creating multiple objects

1. Javascript has no native concept of classes.

Instead:

a) Inheritance via the object tree

b) Instantiation via constructor functions and functions that return objects

The object tree

1. Every object has a parent object (apart from Object.prototype which is the root of the object tree).

2. Attributes are inherited down the object tree

3. If a = new F() then the parent of a is F.prototype

4. Most recent versions of Javascript have a ‘create’ function

Things to try

See handout part 1.

The global object

1. There is a global object. In the browser, ‘window’ is the global object.
2. Anything defined in global scope is an attribute of the global object.
3. Major problem: the global object is the last-resort value for ‘this’

What is ‘this’?

- ‘this’ is a keyword. You cannot directly assign to ‘this’.
- The object referred by ‘this’ in foo() depends on the execution context:
  - foo() — ‘this’ refers to the global object
  - bar.foo() — ‘this’ refers to bar
  - bar = new foo() — ‘this’ refers to bar
  - foo.call(bar) or foo.apply(bar) — ‘this’ refers to bar

Scope

1. A new scope is created every time a function is called.

2. Variables declared with var in the function are added to its scope.
WARNING: If a variable is not declared with var and it is not in the scope of one of the functions above, it will be created with global scope.

3. A function can access variables in its scope and in scopes above it.

A closure is a function that retains a reference to the scope of another function which would otherwise have been destroyed.

The Module Pattern

- Global namespace pollution and collisions are a problem
- We can use closures to create a namespace with private and public attributes and methods.
- Self-invoke a function that returns an object. Use this return value as a namespace.
- Put private attributes and methods in the self-invoked function.

Things to try

- See handout part 2

Warnings and gotchas

1. Calling a function intended as a constructor function without the ‘new’ keyword corrupts the global object.

2. The fact that a.b === f does not mean a.b() === f() (‘bind is transient’)

Books and Websites

- Javascript: The Good Parts by Douglas Crockford
- Professional Javascript for Web Developers by Nicholas Zakas
- JSLint: http://www.jslint.com
- Javascript Garden: http://bonsaiden.github.com/JavaScript-Garden/
- js4py: http://js4py.readthedocs.org
- wtfjs: http://wtfjs.com/