Class construction examples

Notes

In this section we'll describe in general terms the different ways you would use to "create" a "class".

Prototype approach

```
var Foo = (function() {
    var _proto, Foo;

    // class-level methods and variables go here
    // Accessible by all other class and instance methods


    // The constructor.
    function makeNew(initials) {
        var o = Object.create(_proto);
        // Initialize object o;
        return o;
    }
    // Could have other constructors as well


    // prototype object. Could also use Object.create(...)
    _proto = {
        // Your instance methods go here
    };

    // The returned object. This is what your users see.
    Foo = {
        new: makeNew,      // Could name it something other than "new"
        .....     // Exported class methods and constructors
    };

    return Foo;
}());

// Call with
var foo = Foo.new(...);
```

So let us see what goes on:

anonymous function Wrap everything in an anonymous function invocation, to create private scope.

local variables Any local variables to this function are visible by all object methods and class constructors/methods. Use this space for helper methods and objects.

instance methods Define what is to become your instance properties in the _proto object.

class methods Define class methods and constructors as properties of the returned object.

return We return an object with those "class methods" we want the world to be able to use, in particular the constructor(s).

Constructor approach

```
var Foo = (function() {
    var Foo;

    // class-level methods and variables go here
    // Accessible by all other class and instance methods


    // The constructor.
    function Foo(initials) {
        // Initialize object this;
        return this;     // Optional, done automatically
    }

    // Put class methods and variables here
    Foo.aClassMethod = function() { ... }
    Foo.aClassVariable = 5;

    // prototype object.
    Foo.prototype = {
        // Your instance methods go here
    };

    return Foo;
}());

// Call with
var foo = new Foo(...);
```

So let us see what goes on:

anonymous function Wrap everything in an anonymous function invocation, to create private scope.

local variables Any local variables to this function are visible by all object methods and class constructors/methods. Use this space for helper methods and objects.

instance methods Define what is to become your instance properties in this object in the Constructor.prototype object.

class methods Other class methods are properties of this function.

return We return the constructor function itself.