

Metaprogramming JavaScript

Dan Webb (dan@danwebb.net)

WARNING
Code .ahead();

Overview

- ★ What is metaprogramming?
- ★ Why JavaScript?
- ★ What have we got to work with?
- ★ Real examples (from DOM Builder, Low Pro and Sumo)

Metaprogramming...



The writing of programs that write or manipulate other programs or themselves.



...JavaScript

- ★ Vastly underrated
- ★ Perceived as a 'dumb language'...actually dumb programmers
- ★ Very dynamic (Functional programming, prototype inheritance etc...)
- ★ The very nature of JavaScript makes metaprogramming very valuable...

Why?

- ★ More usable APIs
- ★ Compact
- ★ Good clean fun

Metaprogramming Building Blocks

Everything* is a hash

* well, nearly everything


```
pres.js
1 location['href'] === location.href;
2
3 window['randomVariable'] === randomVariable;
4
5 var a = new Array;
6 a['thing'] = 5;
7 a.thing; //=> 5
8
9 Array.prototype.thong = 47;
10 a.thong; //=> 37
11 a['thong']; //=> 37
```

Line: 11 Column: 20 Prototype & Script.ac... Soft Tabs: 2 thong

Functions can be used
to create objects

```
pres.js
36 function Car(speed, passengers) {
37     this.speed = speed;
38     this.passengers = passengers;
39     this.horn = function() {
40         return "BEEP!";
41     }
42 }
43
44 var vw = new Car(100, 4);
45 vw.speed; //=> 100
46 vw.horn(); //=> 'BEEP!'
47
48
49
50
51
52
```

Line: 52 Column: 1 Prototype & Script.ac... Soft Tabs: 2 Car

```
pres.js
54 function Tank(speed, passengers, guns) {
55     this.speed = speed;
56     this.passengers = passengers;
57     this.guns = guns;
58     this.destroy = function (target) {
59         return 'KABOOM!' + target;
60     }
61 }
62 var Tank.prototype = new Car;
63
64 var sherman = new Tank(50, 3, 2);
65 sherman.horn(); //=> 'BEEP!'
66 sherman.destroy('Dan'); // 'KABOOM! Dan'
67
68
69
70
71
72
```

Line: 67 Column: 1 Prototype & Script.ac... Soft Tabs: 2 Tank

Functions ARE objects

```
pres.js
15 function ghostbusters(p) {
16     alert("I ain't afraid of no " + p);
17 }
18
19 typeof ghostbusters; //=> 'function'
20
21 var ghostbusters = function(p) {
22     alert("I ain't afraid of no " + p);
23 }
24
25 each(['ghost', 'ghoul', 'spook'],
26     ghostbusters);
27
28 location.useless = ghostbusters;
29 location.useless('Dan');
30
31
```

Line: 31 Column: 1 Prototype & Script.ac... Soft Tabs: 2 ghostbusters

```
pres.js
94 function size() {
95     return this.length;
96 }
97
98 size.call([1,2,3,4]); //=> 4
99 size.call(document.getElementsByTagName('LI')); //=> 3
100
101
102
103 var slice = function(array, length) {
104     return Array.prototype.slice.call(array, length);
105 }
106
107 var links = document.getElementsByTagName('A');
108 var firstFive = slice(links, 5);
109
110
111
112
113
114
Line: 99 Column: 55 Prototype & Script.ac... Soft Tabs: 2 size
```

The arguments array


```
pres.js
98 function xhrGet(url, options) {
99     options = options || {};
100     // ...
101 }
102
103
104 function sum() {
105     var total = 0;
106     for (var i=0; i < arguments.length; i++)
107         total += arguments[i];
108     return total;
109 }
110
111
112
113
114
115
```

Line: 113 Column: 1 Prototype & Script.ac... Soft Tabs: 2 sum

```
pres.js
135
136 function returnMe() {
137     return arguments.callee;
138 }
139
140 returnMe(); //=> function returnMe()
141
142 function boo() {
143     return arguments.callee.caller;
144 }
145
146 function bang() {
147     return boo();
148 }
149
150 bang(); //=> function bang()
151
152
153
154
Line: 148 Column: 2 Prototype & Script.ac... Soft Tabs: 2 bang
```

Functions remember
the scope they were
created in

```
pres.js
74 function adder(num) {
75     return function(a) {
76         return a + num;
77     }
78 }
79
80 var add5 = adder(5);
81 add5(10); //=> 15
82
83 function Doodad(num) {
84     this.getNum = function() {
85         return num;
86     }
87 }
88
89 var d = new Doodad(10);
90 d.getNum(); //=> 10
91 d.num; //=> undefined
92
```

Line: 81 Column: 18 Prototype & Script.ac... Soft Tabs: 2 adder

And if all else fails...
`eval()`

then mix it all about...

Dynamic functions: DOMBuilder

```
pres.js
5
6
7
8 var ul = document.createElement('ul');
9 ul.setAttribute('id', 'my_list');
10 var li1 = document.createElement('li');
11 var text1 = document.createTextNode('Nigel Havers');
12 li1.appendChild(text1);
13 ul.appendChild(li1);
14 // ... and on and on and on
15
16
17
18
19
20
21
22
23
Line: 203 Column: 1 Prototype & Script.ac... Soft Tabs: 2 function
```



```
untitled
1 DOMBuilder.create('ul', { id : 'my_list'}, [
2   DOMBuilder.create('li', null, ['Nigel Havers']),
3   DOMBuilder.create('li', null, ['Rambo']),
4   DOMBuilder.create('li', null, ['Jesus'])
5 ];
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
```

Line: 4 Column: 42 JavaScript Soft Tabs: 2

```
dom.js — lowpro
dom.js
63     }
64     return DOM.Builder.create(tag, attrs, children);
65 };
66 },
67 create : function(tag, attrs, children) {
68     attrs = attrs || {}; children = children || [];
69     var isIE = navigator.userAgent.match(/MSIE/);
70     var el = document.createElement(
71         (isIE && attrs.name) ?
72         "<" + tag + " name=" + attrs.name + ">" : tag
73     );
74
75     for (var attr in attrs) {
76         if (typeof attrs[attr] != 'function') {
77             if (isIE) this.ieAttrSet(attrs, attr, el);
78             else el.setAttribute(attr, attrs[attr].toString());
79         }
80     }
81
82     for (var i=0; i<children.length; i++) {
83         if (typeof children[i] == 'string')
84             children[i] = document.createTextNode(children[i]);
85         el.appendChild(children[i]);
86     }
87     return $(el);
88 }
89 };
90
```

Line: 84 Column: 9 Prototype & Script.ac... Soft Tabs: 2 tagFunc

```
1 var linkList = $ul({ id : 'my_list' },
2   $li('Nigel Havers'),
3   $li('Rambo'),
4   $li('Jesus')
5 );
6
7 element.appendChild(linkList);
```

```
dom.js — lowpro
dom.js
77     if (typeof attrs[attr] != 'function') {
78         if (isIE) this.ieAttrSet(attrs, attr, el);
79         else el.setAttribute(attr, attrs[attr].toString());
80     };
81 }
82
83 for (var i=0; i<children.length; i++) {
84     if (typeof children[i] == 'string')
85         children[i] = document.createTextNode(children[i]);
86     el.appendChild(children[i]);
87 }
88 return $(el);
89 }
90 };
91
92 // Automatically create node builders as $tagName.
93 (function() {
94     var els = ("p|div|span|strong|em|img|table|tr|td|th|thead|tbody|t
95             "h1|h2|h3|h4|h5|h6|ul|li|ol|li|form|input|textarea|legend|f
96             "select|option|blockquote|cite|br|hr|dd|dl|dt|address|a|
97             "script|link|style|bdo|ins|del|object|param|col|colgroup
98             "label|dfn|kbd|samp|var").split("|");
99     var el, i=0;
100    while (el = els[i++])
101        window['$' + el] = DOM.Builder.tagFunc(el);
102 })();
103
104
Line: 101 Column: 5 Prototype & Script.ac... Soft Tabs: 2 tagFunc
```

```
dom.js — lowpro
dom.js
47     var trans;
48     if (trans = this.IE_TRANSLATIONS[attr]) el[trans] =
49     else if (attr == 'style') el.style.cssText = attrs[
50     else if (attr.match(/^on/)) el[attr] = new Function
51     else el.setAttribute(attr, attrs[attr]);
52 },
53 tagFunc : function(tag) {
54     return function() {
55         var attrs, children;
56         if (arguments.length>0) {
57             if (arguments[0].nodeName ||
58                 typeof arguments[0] == "string")
59                 children = arguments;
60             else {
61                 attrs = arguments[0];
62                 children = [].slice.call(arguments, 1);
63             };
64         }
65         return DOM.Builder.create(tag, attrs, children);
66     };
67 },
68 create : function(tag, attrs, children) {
69     attrs = attrs || {}; children = children || [];
70     var isIE = navigator.userAgent.match(/MSIE/);
```

Line: 58 Column: 13 Prototype & Script.ac... Soft Tabs: 2 tagFunc

Domain specific languages: Sumo

```
hcalendar.js — sumo
x hcard.js • hcalendar.js
1 HCalendar = Microformat.define('vevent', {
2   one : ['class', 'description', 'dtend', 'dtstamp',
3         'dtstart', 'duration', 'location', 'status',
4         'summary', 'uid', 'last-modified',
5         { 'url' : 'url' }],
6   many : ['category']
7 });
8
9
10 var cals = HCalendar.discover();
11
12 cals[0].description; //=> RailsConf 2007
13 cals[0].url; //=> 'http://www.railsconf2007.com'
```

Line: 13 Column: 19 Prototype & Script.ac... Soft Tabs: 2

```
hcard.js — sumo
x hcard.js hcalendar.js
    }
    return fn;
}
}],
many : ['label', 'sound', 'title', 'role', 'key', 'mailer', 'rev'
        'url' : 'url', 'logo' : 'url', 'photo' : 'url', 'email' : 'ur
    }, {
    'adr' : {
        one : ['post-office-box', 'extended-address', 'street-address
              'postal-code', 'country-name']
    },
    // implied org
    'org' : function(node) {
        var org = this._extractData(node, {
            one : ['organization-name'],
            many : ['organization-unit']
        });

        if (!org.organizationName)
            org.organizationName = this._extractData(node, 'simple');

        return org;
    }
}
]);
);
Line: 21 Column: 53 Prototype & Script.ac... Soft Tabs: 2 function
```


Dynamic 'classes': Low Pro

```
pres.js
161 Tab = Behavior.create({
162   initialize : function(options) {
163     // ...
164   },
165   onclick : function(event) {
166     // ...
167   }
168 });
169
170 Tab.apply($$('div.tab'), { startOpen: true });
171
172 Event.addBehavior({
173   'a.external' : function() {
174     open(this.id, this.href);
175     return false;
176   },
177   'div.tab' : Tab({ startOpen : true })
178 });
179
```

Line: 177 Column: 38 Prototype & Script.ac... Soft Tabs: 2 function

```
behavior.js — lowpro
* dom.js * behavior.js * draggable.js
93 // MyBehavior.attach(el, values, to, init);
94 //
95 Behavior = {
96   create : function(members) {
97     var behavior = function() {
98       if (this == window) {
99         var args = [], behavior = arguments.callee;
100         for (var i = 0; i < arguments.length; i++)
101           args.push(arguments[i]);
102
103         return function(element) {
104           var initArgs = [this].concat(args);
105           behavior.attach.apply(behavior, initArgs);
106         };
107       } else this.element = $(arguments[0]);
108     };
109     behavior.prototype.initialize = Prototype.K;
110     Object.extend(behavior.prototype, members);
111     Object.extend(behavior, Behavior.ClassMethods);
112     return behavior;
113   },
114   ClassMethods : {
115     attach : function() {
```

Line: 101 Column: 11 Prototype & Script.ac... Soft Tabs: 2 unload

Wha!?!?