

Javascript - Avoiding the bad parts

Mikko Ohtamaa

mikko@opensourcehacker.com

<http://opensourcehacker.com>

<http://twitter.com/moo9000>

Lighting talk of 5 min

Turn Python gurus to JS gurus

Plone Conference 2011

Javascript has become better

Don't write Javascript like it was 1996

What your every JS file
should look like

```
/*global require,define,window,console*/  
// Main  
require(["jquery", "domready!",  
"submodule"], function($, domready,  
submodule) {  
    "use strict";  
  
    $("a").click(function() {  
        var val = submodule.foobar();  
        window.alert(val);  
    });  
});
```

```
/*global require,define,window,console*/
// Submodule
define(["jquery", "myothermodule"], function($, myothermodule) {
    'use strict';

    // Export public module API
    return {
        foobar : function() {
            return 1+1+myothermodule.func();
        }
    };
});
```

Use require.js

- AMD (Asynchronous Module Definition)
- a.k.a. import for Javascript
- Dependency solving
- Automatic compression and merging of needed and only needed JS files using require.js optimizer

Use ECMAScript5

- Goodies: `this.function.bind()`,
`Array.forEach()`
- Backwards compatibility (Internet Explorer): <http://bit.ly/es5shim>

“use strict”;

Enforces no global variables by default and fixes some other bad language features

Enforce background JSLint checking in your text editor

... or JS*Lint* modern fork, JSHint

```
var x = ix * segment_width - width_half;
var y = iy * segment_height - height_half;

//x += ix * fixFactor;

//y += 1;

var vx = new THREE.Vertex( new THREE.Vector3( x, -y, 0 ) );
```

⚠️ Use '===' to compare with '0'.
y == gridY1-1) {
HACK HACK

```
vx.expandY = false;
```

```
}
```

⚠️ if(ix == 0 || ix == gridX1-1) {

```
// HACK HACK HACK
```

```
vx.expandX = false;
```

```
}
```

```
this.vertices.push(vx);
```

```
}
```

```
}
```

```
// Some JSLint hinting
```

```
/*global window,console*/
```

Never use Javascript
inline in HTML

~~<a href="#"
onclick="aargh()">~~

No

```
// On page load  
$("#handler").click(function() {  
    // do stuff  
});
```

```
// For dynamic content  
// (AJAX loaded)  
$("#handler").live(function() {  
    // do stuff  
});
```

Passing a bound
method to an event
handler using
`function.bind()`

```
function FooClass(){
}
FooClass.prototype.myfunc = function(arg1,arg2) {
}
var f = new Foo();

// Bind arguments are this, arg1, arg2
setTimeout(f.myfunc.bind(f, "arg1", "arg2"), 10);
```

Finding Javascript info

- Use Mozilla Developer Network: “MDN Javascript bind”
- Put w3school.com on ban-list - only **bad** information there

Conclusion

- Worst JS farts have been fixed / can be worked around with discipline
- Javascript still lacks syntactic sugar, is not wrist friendly (boo!)
- Mozilla et. al. are working on to fix that (but Philip didn't promise to drop curly brackets or semicolons)