

Notes - 10/26

Friday, October 26, 2012 10:30 AM

[In 41/2731 at 10:30 today.]

Agenda:

- Design backlog
 - Update on Fundules, clodules and lightweight mixins
 - What should rules be for functions assigned to object types
 - Rule for "var foo.data = 3;":
 - Permitted only if foo declared in same statement block
 - Any scope, even where a "module" isn't allowed - so this works:


```
var baz = (function() {
    function bar() { return 3;}
    var bar.data = 4;
    return bar;
})();
```
 - Examples:


```
var Foo.origin = ...;
=>
Foo.origin = ...;

function Foo.helper() {}
=>
Foo.helper = function helper() {}

class Foo.Bar {}
=>
Foo.Bar = (function() {
    function Bar() {}
    return Bar;
})();
```
 - Oth: Already supported, always
 - Module F.x {}
 - 1st: Class and function declarations F can, in the same scope, be augmented using:
 - var F.x;
 - class F.x {}
 - function F.x() {}
 - NOT module F.x {}
 - 2nd: Interfaces can have dotted names which
 - interface F.x {}
 - 3rd: interface + class is allowed (this is lightweight mixins, need to look at motivating examples)
 - class F {}; interface F {}
- Need to understand the scenarios for first class module types


```
Class C {}
Module M {}

C.Interface
C.static
M.static
staticof<C>
Staticof<M>
Typeof<C>
Typeof<M>
interface<C>
Interface<M>
```

 - ... for arguments
 - "boolean"? (#135)
 - Better typing for Function.prototype.bind (#133)
 - Static initialization (#74)
 - Generics, map out requirements (#185)
 - Thinking on how to approach async via generators (#38)
 - String interpolation
 - SkyDrive feedback – implications for future thinking?
- Others?