

Notes - 12/14

53

Wednesday, December 14, 2011 8:10 AM

[In 41/1587 today at 10:00am]

Agenda:

- Update on engineering and hiring
 - Progress toward 12/16 drop
 - Tutorial
- Make calls on current spec/implementation divergences:
 - Private
- Question: Should we consider including some targeted ES6 features? (destructuring perhaps?)
- Design work:
 - Generics
 - [ES6 modules](#)
 - /r and #reference and other meta-directives
 - modulefaces
 - Mixins and traits
 - Inference for lambdas
- Anything else?

Partial interfaces:

- In C#, accretive, or optional but must match
- Should be accretive

Modules:

- What happens with 'var math = 3;'
- Can we add module

```
(function(global) {
  global.Math.pi
})(this)
```

// compiler flag which requires a single top level module, then codegens it to write onto this (as above), and statically emits an interface:

```
// input.str
Export module Server { // or package Server
  property x = 5;
  class C { ... }
}
```

RegExp literals are a todo

Ambiguity in type grammar - array associates to return type.

ASI for Strada

Design topic: Can static bodies include classes

- For now, change static to ClassSourceElements

Remove imports for now

Do we need global.M1.M2?

Static sections in exgterns

'self'

Remove 'class expression' from current grammar

Lambdas currently require 'function'

'as' expression - where in precedence?

- ComparisonExpression
- Currently seems to be a mix "var z = x as number + x as number"

```
//.str
Interface Server {
  property x = 5;
  interface C { ... }
  property C : { new () : C }
}
```

```
//.js
(Function(Server) {
  server.x =
  server.y =
})(exports)
```

```
///<reference path='node.str'/>
```

```
"use strict";
var http: IHttp = require("http");
class Greeter() {
  function start(port: number, message: string) {
    http.createServer(function(request: IServerRequest,
response: IServerResponse) {
      response.writeHead(200, {"Content-Type":
"text/html"});
      response.end("<h1>" + message + "</h1>");
    }).listen(port);
    console.log("Server running at http://localhost:
"+port);
  }
}
```

```
exports.Greeter = Greeter;
```

```
///<reference path='server.str'/>
```

```
//module server = 'server.str'
```

```
var server = require("server");
```

```
var hellogreeter: Server.Greeter = new server.Greeter();
hellogreeter.start(4000, "hello");
```