

Notes - 4/9

68

Monday, April 09, 2012 10:08 AM

[In 41/4731 at 10:00 today.]

Agenda:

- Review all remaining “Open” issues on the [design items spreadsheet](#)
- Input from Steve on Mankala app building
 - Cross-declaration module privates
- Others?

()=> any = () => void

- Should be allowed
- Assignment compatibility

```
// a.str
module "utils" {
}
var x = 10;
module "hello" {
}
```

```
//b.str
var y = 20;
```

```
//c.str
module "hello" {

}
```

stradac a.str b.str c.str

=> utils.js

=> hello.js

=> a.js

=> b.js

// => c.js (not emitted because it's blank)

```

stradac a.str b.str c.str -interfaces
=> utils.js
=> hello.js
=> a.js
=> b.js
// => c.js (not emitted because it's blank)

```

```

1)
=> a.i.str
=> b.i.str
=> c.i.str

```

```

2. // *.js => *.i.str
=> a.i.str
=> b.i.str
=> hello.i.str
=> utils.i.str

```

```

stradac a.str b.str c.str -out:abc.js
=> utils.js
=> hello.js
=> abc.js

```

```

//a.str
Var x = 10;
Module "utils" {
  var y = x;
}
var z = 30;

```

```

=>
//a.js
var x = 19;
var z = 30;
//utils.js
var y = x;

```

```

enum Something {
  x: 0
  y
}
Something.x

```

```

enum Something {
  "hello",
  "goodbye"
}
var x: Something = "hello";

```

```

var x: Something = Something.hello;
var x: Something = Something["hello"];

```

```
Enum Weekday {  
    Monday: 0,  
    Tuesday  
} // 0 and 1
```

```
Enum Weekday {  
    "Monday",  
    "Tuesday"  
}
```

```
Enum Weekday {  
    Monday: "Monday",  
    Tuesday: "Tuesday"  
}
```

```
Var d: Weekday = Weekday.Monday  
Var d: Weekday = "Monday"
```

```
enum Weekday {  
    Monday: 0,  
    Tuesday: 1  
}
```

```
Var d: Weekday = Weekday.Monday  
Var d: Weekday = 0
```

```
// maybe a bridge too far  
enum Weekday {  
    Monday: 0,  
    Tuesday: "Tues",  
    "Wednesday",  
    Thursday  
}
```

```
Luke to propose:  
// rule: each enum must have a consistent base type  
// RHSs must be literals  
// Everything gets inlined
```

Concrete things:

- Move current enums to object literal syntax

```
class Todo2 {  
  
    foo: string;  
    constructor(foo: string) {  
        this.foo = foo;  
    }  
  
}  
  
class Todo3(public foo: string) {  
}  
  
class Todo4 {  
  
    constructor(foo: string) {  
        public foo = foo;  
    }  
}
```

```
var foo = {  
    name: "Bob",  
    toString: function(this: string) {  
        this.name;  
    }  
}  
  
function Foo(b: bool): string {  
    if(b) {  
        return "hello";  
    }  
    return;  
}  
  
var bar = new Foo();
```

```
}  
  
}  
  
class Todo5 {  
  
    public foo: string[] = ["hello"];  
  
    constructor() {  
    }  
  
    bar() {  
        this.foo[0] = "bye";  
    }  
  
}
```