Notes - 4/13

Friday, April 13, 2012 10:23 AM

69

We have a meeting today (10:15 in 4725), but Anders is out, and we've made good language design progress over the last several meetings. So today, I'd like to focus on some implementation and compiler topics:

- Address any outstanding questions from the dev/qa team on details of recently closed design decisions (in particular, attached)

```
        module Id.Id…; needs to be at the top of the file

        Anonymous modules also supported - slightly different semantic



module "foo";
export module Bar {
  module baz from "baz";
  export name = baz.name;
}
export module Bar2 {
  module baz2 from "baz";
  module {
    module baz2 from "baz";
    export var foo = baz2.name;
  }
}



define("foo", ["baz"], function(__baz__, exports) {

  var Bar;

    (function(Bar) {
        var baz = __baz__;
        Bar.name = baz.name;
    })(Bar);;


  var Bar2;

    (function(Bar2) {
        var baz2 = __baz__;
        (function(Bar2) {
            var baz2 = __baz__;
            Bar2.foo = baz2.name;
        })(Bar2);
    })(Bar2);;


    exports.Bar = Bar;

})



Strada bindings;
  • module: var
  • class: var
  • var: var
```

- property: var
- function: function

Use before def:
- Generally seems useful
- Couldn't identify any places where this is not wanted
- Some concern about implementation ordering - expected to be solvable

Reachability;
- yes

- Review status of “Closed” but “Not Yet Implemented” features in design items spreadsheet
- Raise any outstanding design concerns that are not yet captured in the design items spreadsheet
- Discuss compiler specific items from design items spreadsheet
  - Jshint-style static analysis

Should number errors/warnings so we can later turn on/off.

Semantic	Strada	Alex	Andre	Ben	Chris	Dirk	Erich	Joao	Johannes
Undeclared variable assignment	Done	*	*	*		*	*	*	*
Undeclared variable read	Done	*	*	*		*	*	*	*
Variable re-definition in same scope	Useful hint	*	*	*		*	*	*	*
Using a variable declared inside a previous block in same scope	Useful hint	*	*	*		*	*	*	*
Unused local variable	Useful hint	*	*	*		*	*	*	*
=== instead of ==	Lint not compiler	*	*	*		*	*	*	*
for in hasOwnProperty check	Lint not compiler	*	*	*					
Empty blocks	Lint not compiler								*
Warning if "use strict" is missing	Lint not compiler	*		*		*	*		
Missing new prefix when	Done			*		*	*		

invoking a constructor									
Missing semicolons	Lint not compiler	*		*			*		
Double semicolons	Lint not compiler	*		*					

Stylistic	Alex	Andre	Ben	Chris	Dirk	Erich	Joao	Johannes
Mixed tabs/spaces	*		*				*	
Curly enforcement	*		*			*		*

Pasted from <http://devdiv/sites/bpt/Zollikon/_layouts/OneNoteFrame.aspx?d=O76078548bd0f41e194ea2ca6bcd860bam4e7874b4641c45d0aaedc6dff06fc7a8m632ea5ebd9d34549833093fe9046e581m&wd=target%28Explorations%2Eone%7C694EFBB0%2D65CB%2D49CB%2DBE91%2D463E7E3B8583%2F%29>

- Integrated minification/optimization
- Others?

Nested classes:

```
class Parser {
  public class Token() {

  }

  static parse(): Token[] {
    var tok = new Token();
    return [tok];
  }
}

var toks: Parser.Token[] = Parser.parse()
```

```
// Alternative

module ParserStuff {

  var foo;

  export class Token() {

  }


export class Parser {
```

```
  static parse(): Token[] {
    var tok = new Token();
    return [tok];
  }
}

}
var toks: ParserStuff.Token[] = ParserStuff.Parser.parse()
```

```
// Renaud example:
walk(pre, post, ctx)

function Bar() {
  class C() implements IFoo {}
  return <{new (): IFoo}>C;
}
```

Want to propose - Brian and Renaud to send examples