Ecma/TC39/2010/003

Module Primordials Man of Straw

Kris Kowal

For ECMA TC-39, January 27-28 2010

- grammar
- primordials
- why

Grammar

import *id* [from *pkg*]$_{opt}$ [with *scope*]$_{opt}$

GRAMMAR

```
(import id)
```

```
require(id)
```

DESUGAR: MEMOIZED MODULE EXPORTS FROM WITHIN THE CURRENT PACKAGE

```
(import id from pkg)
```

```
require(id, pkg)
```

DESUGAR: MEMOIZED MODULE EXPORTS FROM AN EXTERNAL PACKAGE

```
(import id with scope)
```

```
load(id)(scope)
```

DESUGAR: CALL A MAKER FROM THE CURRENT PACKAGE

(import *id* from *pkg* with *scope*)

load(id, pkg)(scope)

DESUGAR: CALL A MAKER FROM AN EXTERNAL PACKAGE

Primordials

```
new Context()
```

CREATE A NEW EXECUTION CONTEXT; THIS IS THE INTERFACE

```
let RhinoContext =
    Packages.org.mozilla.javascript.Context;
let rhinoContext = new RhinoContext();
rhinoContext.initStandardObjects();
```

```
new Module(
    text,
    fileNameopt,
    lineNoopt
)
```

MODULE CONSTRUCTOR

```
new Function(
    ...args,
    text,
    fileNameopt,
    lineNoopt
)
```

```
new Module(
    ...args,
    text,
    fileNameopt,
    lineNoopt
)
```

COMPARE AND CONTRAST

```
new Module(
    text,
    fileName_opt,
    lineNo_opt
)
```

```
module(
    scope_opt,
    load_opt,
    require_opt,
    context_opt
)
```

MODULE CONSTRUCTOR

MODULE FUNCTION

```
let f = new Function(“x”, “y”, “return x + y”);
let z = f(10, 20);
```

```
let m = new Module(“return x + y”, “m”, 1);
let z = f({“x”: 10, “y”: 20});
```

EQUIVALENT PROGRAMS

```
let m = new Module(“import ‘m’”);
m.dependencies[0][0] == “m”;
```

STATIC ANALYSIS OF MODULE DEPENDENCIES

```
array [
    array {
        string id;
        any pkg;
    };
] dependencies;
```

“ORDERLY” JSON SCHEMA OF DEPENDENCIES

```
[
    [id, pkg],
    [id, pkg],
    [id, pkg]
    ...
]
```

EXAMPLE

Roadmap

- `require`, `exports`, `module`
  - Migration: `<script>`able modules
- `Context`, `Module`, `import`, `export`
  - Migration: `require`, `exports`, `module`

Why

- frozen globals vs.
  mutable globals vs.
  isolated worlds

- singleton modules vs.
  e-makers

- require, exports, module vs.
  import, export

- return promises vs.
  alter exports object

FALSE DILEMMAS

HTTP://WIKI.ECMASCRIPT.ORG/DOKU.PHP?ID=STRAWMAN:MODULES_PRIMORDIALS