Minutes of the: ***Ecma TC39, ES3.1WG***

held in: ***Phone conference***

on: ***10 July 2008***

1 Roll call and logistics

1.1 Participants

Mike Cowlishaw (IBM), Doug Crockford (Yahoo!), Pratap Lakshman (Microsoft), Mark Miller (Google), Adam Peller (IBM), Sam Ruby (IBM), Allen Wirfs-Brock (Microsoft) and Kris Zyp (The Dojo Foundation)

2 Agenda

Decimal

3 Minutes

In the process of adding Object.keys - if the first parameter is not an Object we should throw an exception - only in the cautious mode ? - no, throw irrespective of the mode - look at it on a case by case basis and make sure it applies in all situations.

Decimal

Should we be defining an object model for decimal values and use OO style method invocations ? - or should we just define static methods - all operations should be specified as static methods on the Decimal constructor - these methods should be sealed and the global binding of Decimal should also be sealed and readonly - just expose basic decimal capabilities in a way that can be supported efficiently by implementations.

What do we see as the primary use case for decimal - do we expect it to be used by user code (as in the parking meter case cited early on) or by code generators? - primary motivation is for applications to be able to do precise arithmetic.

Lets make a subroutine library that can be wrappered - code generators could directly call the static methods - methods would take all the necessary arguments - ugly - and inconsistent with the rest of the global library - why not be able to lock it down - its about what you can do on the implementation side - target for a layered language - in ES4 you cannot rebind a global name - but you can monkey patch the members - would like to be as consistent as the style of the existing library it is being added to - can we have the statics as well as the non-statics ? - which is more primitive? lets add only the more primitive ones for this iteration - ideally you want to cleanly integrate the capability into the foundations of the language - for ES3.1 we cannot get it in that deeply - lets provide it as library - parking meter folks would use decimal arithmetic using either the primitive operations or by writing a wrapper - nothing convenient about the static method proposal - create wrappers if you want convenient - use objects to implement a primitive numeric type

For now lets have a decimal by adding to the spec from the DecNum package as well as have static methods - post Oslo we can, if required, yank the higher level stuff.

ES4 non-strict too is conditioned by opt-in

Not enough clarity around this till now - without the opt-in, non-strict ES4 behaves like ES3 - in the case of ES3.1 certain semantic differences are observable even outside of the cautious subset - how do we handle function declaration in blocks ? - spec needs to call out non-opt in

behaviour - specifically function declaration in blocks must be specified - caja implements an intersection of current browser semantics; for function declaration in blocks, scoping of the var hoisted to containing function but its initialization gets turned into an assignment that gets hoisted to the top of the block of the containing function - no! not representative of intersection semantics; won't work in IE.

use subset cautious prohibits var declarations inside blocks - why not say if you are in cautious code functions are properly block scoped - what if you take out the cautious code; now it would behave differently - how about a new directive 'use nested blocks' - ES3 grammar anyway does not function declarations within blocks; why not enforce that in cautious mode - in ES3.1 treat functions like vars - if you are in cautious mode they are prohibited in blocks - you can still get block scoped functions using const foo = function () {}.

Updating to draft

Do we keep posting updates to the draft - yes, important updates should be circulated on the discuss lists and the draft must be kept updated - if people get the time to readup on the latest drafts that would be good; if not we must be ready to review the "Oslo" draft and call out which sections have evolved.

Meeting adjourned.