

Minutes of the:

Ecma TC39, ES3.1WG

on:

Phone conference

13 May 2008

1 Roll call and logistics

1.1 Participants

Doug Crockford (Yahoo!), Collin Jackson (Stanford), Pratap Lakshman (Microsoft), Mark Miller (Google), Adam Peller (IBM), Sam Ruby (IBM) and Kris Zyp (The Dojo Foundation)

2 Agenda

- (1) http://wiki.ecmascript.org/doku.php?id=es3.1:richer_reflection_capability_on_function
- (2) http://wiki.ecmascript.org/doku.php?id=es3.1:inner_functions_and_this
- (3) http://wiki.ecmascript.org/doku.php?id=es3.1:targeted_additions_to_array_string_object_date - specifically what additions do we want on Function and Object.

3 Minutes

Richer reflection capability on function

http://wiki.ecmascript.org/doku.php?id=es3.1:richer_reflection_capability_on_function

The one we are now calling "parameters" operates the same in strict and non-strict mode. It always returns an array of the declared parameter variable names, i.e., an array of strings - this had been misnamed "arguments" on the ES3.1 wiki pages. This (as well as "name" and isBoundedArity (name TBD) become own properties of each individual function object.

The Function "arguments" property implemented by all current major browsers, but not specified by any standard. This is the one whose defacto behavior we'll be codifying in annex B (along with Function.caller and arguments.callee), but specifying to throw on strict functions.

Function.caller behaves the same way - honoured in a non-strict module; will throw an exception in a strict module - add these to non-normative AnnexB in the spec - note that the "name" property shall return the empty string in the case of anonymous functions.

Inner functions and this binding

http://wiki.ecmascript.org/doku.php?id=es3.1:inner_functions_and_this

The proposal as worded on the wiki is not compatible with the proposed ES4 behaviour - in a non-strict module, the behavior is as in ES3; in a strict module an attempt to access "this" will throw an exception - this is compatible with proposed ES4 behaviour (when 'this' is bound to null or undefined)

Targeted additions to Function

http://wiki.ecmascript.org/doku.php?id=es3.1:targeted_additions_to_array_string_object_date

Need to add a "bind" method - the bind method (Function.prototype.bind) creates a new function that carries thisValue and the optional arguments. this will be faithfully bound to thisValue, even if thisValue is null or undefined - is it too late to add properties on the prototype - the Prototype framework already does this - we can add it to the prototype - it will not be 'readonly' - an existing Ajax library that wants to override it to provide an incompatible implementation can do so.

If we are going to provide a 'bind' do we still want the Array generics to take the optional second parameter? - in the case of the Array generics is the callback invoked with the collection bound to the 'this'? - actually, if the second argument is not provided or is null, the global object associated with "callback" is used instead - since we are going to add a 'bind', the Array generics need not take the optional second parameter to indicate the thisObject - so these Array generics will take only one parameter which will be the callback - the callback shall be invoked as a 'function' - if there is a 'this' used within the callback it will have the same semantics as any function - an attempt to access it in a strict module will throw an exception while in a non-strict module it will behave as in ES3.

Is the defineProperties sketch ready for the proposed-ES4 folk to evaluate? - they are compatible with the getters/setters proposal Kris is working on - what about procedural properties? Do they have a getter and a setter? - the getter of a procedural property `_defaults_` to a function that returns undefined. In other words, that's what you get if you define a procedural property by defining only a setter. - we can have the proposed-ES4 folk evaluate the sketch.

pratapL to upload the working ES3.1 doc onto the wiki so that others might mark it up.

Meeting adjourned.