

Minutes of the:

Ecma TC39, ES3.1WG

Phone conference

held on:

29 April 2008

1 Roll call and logistics

1.1 Participants

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

2 Agenda

String/Array generics

Getters/setters (from Kris)

Meta

3 Minutes

String/Array generics

Can't think of a reason to have static generics on String - is it too late to put things on the prototype? - if someone is already putting trim on the prototype you don't want to collide - frameworks have been adding methods to the prototype and we don't want to collide - are these readonly? Would things fail silently if you override? - fail-silently is the current semantics for readonly; unless you are in strict mode, in which case it would throw - collision-avoidance cannot be the motivation; frameworks typically test for the existence of such methods before adding their own - this looks like a case of false economy - one possible use case would the for-each used in conjunction with generics - how about convenience? Having them as statics that take an additional "stringLike/ArrayLike" parameter makes it more convenient to apply/call on other types of objects - if you really wanted to mixin such methods on objects other than Strings you would want to define a mixin prototype for this purpose - that seems like the JavaScript way of doing things - that seems more like the 'prototype language' way of doing things - the convenience argument is not convincing if it involves adding a whole new set of such parallel functions - check with Mozilla for the motivation.

Array generics may be more useful - some of the Array methods cause mutation while others do copying - only the copying methods make sense on strings.

Strong preference to add trim as a method on the prototype - ensure its definition of whitespace matches the definition being used by RegExps - RegExps use the same definition for whitespace as the rest of the language - that is a good thing - the trim proposal should be modified such that it relies on the regexp definition of whitespace - need to work out the handling of \x00, \u2028, and \u2029 - also, it was unclear whether the other String methods have READONLY status in the spec or only in Mozilla, and how that would affect our proposal.

We should add String.quote too - simple, useful, and easy to implement - apps can use this to defend against XSS attacks (provided the function does proper escapement).

Add trim and quote to the proposal (as methods on the String prototype) and send it out on the discuss lists.

Decimal

Mark's quirky interesting idea on integrating Decimal into ES3.1 - the main motivation for decimal is to represent currency - a peculiar fact is that \$23 (for example) is a valid identifier in JavaScript - when the engine sees \$23, it can trigger a catch all that gets the identifier (in this case \$23); the catchall can then invoke the decimal library and set the decimal value 23 to the identifier \$23 - how would you handle the decimal point - you could use the underscore for the decimal point - does \$ indicate currency world-wide? - well, this is just a quirky interesting idea - Mark's creative decimal literal idea was unanimously rejected, including by Mark himself :-)

Getters/Setters

Add this as the first item to discuss for the Thursday call.

Meta

Not discussed.

Mark Miller will not be able to make it for the Thursday call.

Meeting adjourned.