

From: Wiltamuth S.

>Sent: dimanche, 26. janvier 1997 18:44

>To: e-tc39

>Cc: 'clayton@netscape.com'

>Subject: 1/24 ECMAScript working group notes

>

>

>In attendance

>-----

>Randy (Borland)

>Clayton Lewis (Netscape) -- clayton@netscape.com

>Shon (Microsoft)

>Scott (Microsoft)

>Andy, via phone (Silicon Graphics)

>

>

>Next meeting

>-----

>We tentatively agreed to meet in person at Borland this coming Friday from 1-6 pm. (Randy has a near-conflict that is also alternating Fridays, and he would like to interleave these meetings if possible.)

>This is fine by MS and Borland. Clayton can personally come; he will check Brendan's availability.

>

>

>Revision marks

>-----

>We started the meeting by examining revisions from the last working group meetings. Everyone agreed that from here on, we should use revision marks, since this will make this "checking to see that we got it right" process easier.

>

>

>Things that we agreed on that didn't yet make it into the doc

>-----

>* When errors are raised. The spec specifies the last possible moment that an error can occur. Implementations may err earlier. This will introduce some differences, due to side effects. If we do try?except in the future then trappable errors must be specified precisely.

>

>* Unicode escape sequences and end of source. We should make sure that a user-defined unicode sequence not be misinterpreted as specifying end of source. E.g., \u0000. The spec should say that there is a magic character/token that is placed at the end.

>* Embedded nulls are allowed in strings.

>* Need to fix the grammar for octal, as we discussed last time. Here's what we said: For numeric literals, if there is a leading 0 not followed by a dot, an x or an e. E.g., 001.5 generates two tokens -

>001

>and .5. This would be an error.

>* 7.2.2: the resolution of the issue listed there is that the type of i++ is always a number.

>* 7.5.2: Remove steps 5 and 6.

>* 8.4.2: The for loop case: for the "var in" case needs to include

>multiple vars, not just one.

>

>

>Comments on the things we have resolved that have not gotten into the doc

>-----

>-----

>* Adding form feed and vertical tab as white space. This got into one place in the doc, but it also needs to be in the simple whitespace productions -- 5.3.1 and 3.1.

>

>* Auto-semicolon insertion. This got into the doc, but there are some issues. This case:

> x = a + b

> ++i

>errs according to the rules in the doc. The programmer intent is clear, so we should try to handle this in the spec.

>

>Also, this case:

> for (a; b

>)

>will be interpreted as

> for (a; b;)

>

>Shon had some minor editing comments about how best to describe the rules.

>

>* Reserved words. Randy inadvertently left out "do" as a future reserved word. We added do...while to the extensions list. Reserved words that would be used for items on the extension list are listed as future reserved words.

>

>* Reserved words. The footnotes should be removed.

>

>* The 7.4 section was just added. Comments on it:

> * It probably makes sense to deal with NaN implicitly via the building

>blocks (e.g., *, /) rather than explicitly.

> * 7.4.1 and 7.4.2 -- "finite" should be "finite non-zero".

>

>* The 7.5 section was just added. The last bullet point looks like it is talking about all types, when it is really just talking about numeric types.

>

>* Hints for ToPrimitive. This got in, but there were editing comments

>about how best to describe this. Also, the description needs to say

>explicitly that the hint is passed through to DefaultValue.

>

>General comments on the doc

>-----

>We spent a significant amount of time on minor doc comments, both minor corrections and editorial comments. I didn't take notes on this part of the meeting.

>

>

>Resolved issues

>-----

>* Need to more rigorously define "anonymous code".

>

>This is now in the doc.

>

>* Will the style of the document be acceptable to ISO?

>

>Randy is working on style issues, and has existing language standards in hand.

>

>

>Open issues

>-----

>

>* Non-1970 dates. We will discuss this as an issue. We will not discuss a date type.

>

>* Versioning.

>

>* Prototype property attributes regarding get and put. Looking at 4.5.2 in v .3. If the prototype for an object has a ReadOnly or ErrorOnWrite member, and code attempts to do a put on the object with the same name as the prototype object member, then what is the result? That is, are the attributes of the prototype member respected? Brendan to define an example that justifies changing what is in the spec.

>

>* Order of evaluation for assignment. Randy and Shon now agree with the left-to-right logic. Brendan needs to think about this.

> // Example 0

> x = y = z = 1

- >
- > // Example 1
- > var x = 1
- > var o = new Object()
- > with (o)
- > x = o.x = 2
- > print(x) // Is x 1 or 2? It's 2 if we do left-to-right. This is what is in the spec today.
- >
- > // Example 2
- > o[i] = i++
- >
- >* Max string length. Proposal - implementations must support strings with at least 32000 characters. There was discussion about whether to have a minimum or not.
- >
- >* && and || semantics: like PERL or like C++, C and Java? E.g., (true && 10)
- >
- >* Eval function. Need to define object scoping with an eval block.
- >
- >Host-supplied members of scope chains vs. Implicit this.
- >
- >* Lifetime of Activation Record Object (has scope chain)
- >
- >* Should the arguments object include local variables?
- >
- >----- new issues below this line -----
- >
- >* What should division by zero result in? Shon thinks that division by zero should generate NaN rather than signed infinity, as specified in the doc today, and that we should not have the concept of positive 0 and negative 0.
- >
- >* Should comments nest?
- >
- >* According to the grammar, this: "\h" is an error. Maybe this should be interpreted as "h". General agreement that we should probably just match Java, though we don't know what Java does.
- >
- >* Does null coerce to 0 or NaN? We decided this but no one can remember the result.
- >
- >* What hints are used in which cases? The hint is optional, but we still need to examine these cases to make sure we're doing the right thing.
- >
- >* Should the Activation object and the arguments object be the same?

>If not, then two arguments share the same members.

>

>* 7.3.3: Shon would like to see us specify that the behaviors defined here are for native objects. We can either leave the external objects issue open, or we can specify that typeof(external object) returns "object" in all cases.

>

>

>Work division

>-----

>Shon has on his plate:

>* Working on the scoping section (Chapter 6). We will focus on this in the next meeting.

>

>Scott has on his plate:

>* Write versioning proposal/sketch..

>

>Randy has on his plate:

>* Working on object model section. We will focus on this in the next meeting.

>* Randy feedback from today's meeting

>* Proposal for extending the range of dates

>

>Brendan has on his plate:

>* ToInt32, ToUInt32

>* ToString(Number)

>* Arguments object and activation object. We will focus on this in the next meeting.

>

>

>Additions to the list of proposed extensions

>-----

><None>