

## Standardizing Information and Communication Systems

From: Wiltamuth S.

>Sent: samedi, 1. février 1997 18:11

>To: e-tc39

>Cc: Lenkov D.

>Subject: Notes from the 1/31 working group meeting

>

>

>In attendance

>-----

>Andy Palay from SGI

>Randy Solton from Borland

>Michael Gardner from Borland

>Clayton Lewis from Netscape

>Mike Ksar from HP

>Scott Wiltamuth from Microsoft

>Shon Katzenberger from Microsoft

>Guy Steele from Sun (by telephone)

>

>

>Next meetings

>-----

>2/7 1-2 pm by phone, arranged by Michael Gardner

>2/14 1-6 pm in-person at Netscape, organized by Clayton Lewis

>

>

>Review status of contributions

>-----

>Shon:

>\* DONE: Revise the scoping section (Chapter 6).

>

>Scott has on his plate:

>\* DONE: Versioning proposal/sketch

>

>Randy:

>\* UNDONE: Working on object model section.

>\* DONE: Revisions based on the last meeting

>\* Proposal for extending the range of dates.

>

>Brendan:

>\* `UNDONE`: `ToInt32`, `ToUInt32`

>\* `UNDONE`: `ToString(Number)`

>\* `UNDONE`: Arguments object and activation object.

>

>

>New work we parcelled out during the meeting

>-----

>Clayton:

>\* Draft section defining the semantics of numeric literals. E.g., the

>`0x11111111111111111111111111111111` issue.

>

>Randy:

>\* Incorporate changes from today's meeting

>

>Guy:

>\* More rigorously define the `Number` type.

>

>

>Add to the extensions list

>-----

>\* Unicode variable names. We do not believe there is any problem with adding htis in the future -- adding it would not break existing programs.

>

>

>Resolved issues

>-----

>\* Unicode. Shon points out that unicode escape sequences are only allowed in string literals anyway, so our statement about these should be more narrow; today this statement is quite broad. If we did this, then the end of source character could also be `/u0000`. Agreed on this point -- te statement regarding unicode escape sequence should move from the source description (generic) to the string literal description area (specific).

>

>\* Escape sequences -- when the escape character is used with a regular character, the escape character itself should just be ignored. E.g., `"/h"` just becomes `"h"`. Resolved.

>

>\* `5.3 ToNumber(Null)` generates `NaN`. We could not remember the resolution last time. Now, everyone remembers and agrees. This is good. Resolved.

>

>\* Can an implementation make `a["ab\x00cd"]` equivalent to `a["ab"]`?

>Agreed -- for internal objects, the answer is no. For external objects, this is beyond the scope of the standard. There should be a note about this in the doc. Resolved.

>

>\* Automatic semi-colon insertion. Shon and Brendan both agree that we should avoid having a specification in this area that essentially requires an implementation to lookahead more than one token. Resolved.

>

>Changes we agreed to:

> Guy's suggestion:

> In 7.2, say that Postfix operators cannot be separated from the MemberExpression by a newline.

>

> Shon's addition

> Automatic semi-colon insertion is turned off within a for(...)

>

>This case:

> (Example 1)

> x = a + b

> ++

>is an error.

>

>This case:

> (Example 2)

> x = a + b

> ++;

>is an error since it is interpreted as

> x = a + b;

> ++;

>

>This case:

> (Example 3)

> for (; x < 10;

> ++x)

>is interpreted as:

> for (; x < 10; ++x)

>

>This case:

> (Example 4)

> x = a + b

> ++c

>is interpreted as:

> x = a + b;

> ++c;

>

>This case:

> (Example 5)

> for(;;x

> ++)

>results in a parsing error. No one has a problem with this.

>

>

>\* 7.2.4 Shon proposes swapping steps 2 and 3. This would be consistent with the "left-to-right" decision that we made earlier. Resolved.

>

>\* 7.12.2 Swapping 2 and 3 again. Same issue. Resolved.

>

>\* 7.11 Resolved:

> LogicalOrExpression ? Expression : ConditionalExpression should be

> LogicalOrExpression ? Expression : AssignmentExpression

>So this is legal:

> a ? b : c = d

>

>And this example:

> a ? b : c, d

>is interpreted as

> (a ? b : c), d

>

>\* 4.6 A little wordsmithing -- "represents" should be "consists of".

>Resolved.

>

>\* 4.4 We need to define this type better. Guy will take a whack at this.

>

>\* 5.3.1 Resolved:

> +/- are missing from StrIntegerLiteral. Lengthy discussion about how best to describe this.

> We should allow additional whitespace characters before or after the number. The ones

> in question re: carriage return and line feed.

>

>\* 7.2.6 Return Result(3), not Return Result(32). Resolved.

>

>\* 7.4.3 As with the other sections, if either is NaN then the result is

>NaN. Resolved.

>

>\* D.1.15 Issue out of place.

>

>\* Resolved: nested comments (`/* xx /* */`) are not allowed, and will never be allowed.

>

>\* 7.5.1 and 7.7 Resolved: The hint for `ToPrimitive` should be `Number`.

>

>\* 4.5.2. Resolved: The `[[Call]]` one should be "Executes the object"

>

>\* 8.4 Resolved -- there is an error in the `(for var ; ; )` case -- "var

>Identifier = VariableDeclarationList" should be "var

>VariableDeclarationList".

>

>\* 7.3.3: What does `typeof` return for external objects? Resolved: we will leave this unspecified.

>

>\* Prototype property attributes regarding `get` and `put`. Resolution --

>`ReadOnly`, `ErrorOnWrite`.

>

>\* Minimum max string length. Resolved: we will not specify a minimum max-string-length. The length of the longest string that can be used is implementation-dependent.

>

>\* 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. Resolved -- `ToNumber` does number, `ToString` does string,

>`Addition` and `Relational` operators do `Number`.

>

>

>Review the new scoping section (from Shon)

>-----

>Shon walked people through that he wrote. I took no specific notes on this.

>

>

>Open issues that are blocked by Brendan's absence

>-----

>\* 7.4 Zeroes. Clayton wants to preserve the sign of zero. Shon believes this would be bad because it constrains implementations to use IEEE-64-bit for representing zero internally. This can be done today with an integer 0, and this is more efficient. No resolution. NS to consider Shon's reasoning.

>

>\* 7.8 Equality operators. Consider comparing `(null == undefined)`.



>Other open issues

>-----

>\* Non-1970 dates. We will discuss this as an issue. We will not discuss a date type. Waiting for a proposal from Randy. Shon raised an interesting sub-issue -- that you can do SetYear(96). Does this mean 96 A.D. or 1996?

>

>\* Versioning. Scott contributed a high-level proposal, which we need to review.

>

>\* 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. IEEE-64 does have these concepts. We are approximately evenly split on this issue. We agreed to think about this for one (and only one) more week. The next time we meet, we will settle this issue, even if settling it requires flipping a coin to determine the result.