Minutes of the: Ecma TC39-TG1
held in: Phone conference
on: 27th September 2006

Attendees

- Jeff Dyer, Adobe Systems
- Lars Hansen, Opera Software
- Dave Herman, Northeastern University
- Brendan Eich, Mozilla Foundation
- Douglas Crockford, Yahoo!
- Dan Smith, Adobe Systems
- Michael O'Brien, Mbedthis
- Cormac Flanagan, UC Santa Cruz
- Graydon Hoare, Mozilla Foundation
- Steven Johnson, Adobe Systems
- Francis Cheng, Adobe Systems

Notes (Proposals walkthrough)

Splice syntax

Brendan: Is the grammar acceptable? Is it too LR? The problem is the : which is somewhat ambiguous, it can be a problem if the precedence of the expressions is not the right one.

Doug: could we use different punctuation?

Brendan: nice to be compatible.

Brendan: will convert to Expression operands for splice syntax.

Brendan: how do we feel about a third step expression?

Intrinsic

Pretty much every built-in method has an intrinsic counterpart. (This was semi-decided before and is implicit in the built-in classes but was not really stated anywhere.) Jeff will write it up.

Yield syntax

Jeff/Brendan: Detailed discussion about operator precedence, not quite resolved yet. Partly an issue of grammar form, partly an issue of what feels right and how to capture that.
Grammar form

Jeff: What should be the format of the grammar? LR is much easier to read than LL.

Brendan: We need something that can be converted to LL.

All: Discussion about language classes and inclusion, theorems of rewritability?

Brendan: We need to at least believe that a transformation is possible.

Jeff: the annotations on the nonterminals in the grammar can be expanded to yield a nonannotated grammar.

Trim

Java strips control characters (!) but not Unicode spaces.

We definitely want to strip Unicode spaces. But control characters?

Doug: can we have a generic clean method that strips control characters too? The use case is prepping data for sending to server via POST. Data to be stripped could be both accidental and malicious.

Brendan: `String.prototype.replace()`?

Doug: works, but ...

Dave: but will it clean all it needs to clean?

Brendan: we should probably keep this proposal simple, let regexp fixes cover the rest of the cases.

Doug: a lot of the frameworks do their own suites of functions to do this.

Ian cleans this up.

Regular expressions

(Digressing from trim discussion.)

Our character classes stink (notably for \w). We should consider fixing this. Apparently there are suggestions on http://www.unicode.org. Steven “volunteers” to look into it.

JSON

We allow DAGs but not cycles. (It’s cheaper, and it’s a convenience for the user.)

Prototypes and DontEnum properties are not serialized.

Unclear if the complexity of prettyprinting is worth it.

Doug: throw/nothrow is a nice convenience for the user, some users reference DOM from their nodes but don’t want to have to remove things before serializing.
Dave: prefer a lambda to filter rather than the array, but perhaps nice to support several possibilities here for convenience. Will propose something.

**Documentation**

Brendan: not sure of the utility of documenting initializers and function expressions, but probably OK.

No real objections to Doxygen-style text.

**Questions from Michael**

- Types are also objects
- You just name them in an expression: `int`
- All number classes derive from `Object`

**Recent changes**

Date/time: motivation for separate parse function is that all existing implementations would reject an ISO timestamp, it might be cleaner to have a new function. Motivation for +/- is to disambiguate in the full ISO grammar (that nobody implements).

Brendan/Lars lean toward just shoehorning it into `Date.parse`, it seems better to have just one function.

Brendan: capitalization of property names could be nicer.

•