Minutes for the: *25th meeting of Ecma TC39*

held in: *Cupertino, CA, USA*

on: *16-17 November 2011*

1 Opening, welcome and roll call

1.1 Opening of the meeting (Mr. Neumann)

The TC39 meeting (hosted by Apple in San Cupertino, CA) was opened by **Mr. Neumann**, Chair of TC39 at approximately 10:15 AM on 16th November 2011 (TC39/2011/047 - Venue for the 25th meeting of TC39, Cupertino, CA, November 2011).

It was noted that before the TC39 meeting, on the 15th of November 2011 the TC39 ad-hoc group on internationalization (i18n Ad Hoc group) has also met. The report of that meeting is given under 4.3 below. However, detailed technical report on i18n Ad Hoc group work was a major part of the technical discussion on the afternoon of November 16, 2011.

1.2 Introduction of attendees

John Neumann – Ecma International

Istvan Sebestyen – Ecma International

Josée Auber – HP (President of Ecma International) – part-time

Isabelle Valet-Harper – Microsoft (Vice President of Ecma International) – part-time

Alex Russell - Google

Waldemar Horwat - Google

Allen Wirfs-Brock - Mozilla

Sam Tobin-Hochstadt - Northeastern University

Douglas Crockford - Yahoo!

Brendan Eich - Mozilla

Mark Miller - Google

Luke Hoban - Microsoft

Dave Herman - Mozilla

Eric Arvidsson - Google

Oliver Hunt – Apple

Gavin Barradough – Apple

Edward O'Connor – Apple

Bill Ticehurst – Microsoft

Nebojsa Ciric – Google

Tom van Cutsem - VUB

Norbert Lindenberg - guest – no affiliation

1.3 Host facilities, local logistics

Oliver Hunt welcomed on behalf of Apple the delegates and provided logistical information. It was announced that Ecma international would host the usual social event on November 16th evening.

2 Adoption of the agenda (2011/049-Rev2)

Ecma/TC39/2011/049-Rev2 contained the Agenda for the 25th meeting of TC39, San Cupertino, November 2011. This was agreed without changes.

However, **Waldemar Horwat** has observed that although ES6 has been "frozen" at the May 2011 meeting of TC39, several new proposals have found their entry into the agenda, and their status raises concerns. It was agreed to discuss this general policy point during the general discussion.

The relevant Ecma TC39 contributions for the meeting are the following:

- Ecma/TC39/2011/044 TC39 chairman's report to the CC, October 2011
- Ecma/TC39/2011/046 Draft minutes of the 24th meeting of TC39, San Francisco, September 2011 (Rev. 1)
- Ecma/TC39/2011/047 Venue for the 25th meeting of TC39, Cupertino, November 2011
- Ecma/TC39/2011/048 Draft "ECMAScript Globalization API Specification", 31 October 2011
- Ecma/TC39/2011/049 Agenda for the 25th meeting of TC39, Cupertino, November 2011 (Rev. 1)
- Ecma/TC39/2011/050 Extension of experimental software copyright policy
- Ecma/TC39/2011/051 ECMAScript Globalization API
- Ecma/TC39/2011/052 Test262 Status Report, November 2011
- Ecma/TC39/2011/053 Harmony Proxies: update by Tom Van Cutsem and Mark S. Miller
- Ecma/TC39/2011/054 Notes of the ad hoc meeting on Internationalization, 15 November 2011

Other documents are mentioned via their URL to the ES Wiki.

The more detailed technical notes by **Mr. Horwat** are attached to this report.

3 Approval of minutes from September 2011 (2011/046)

The minutes of the 24rd TC39 meeting in San Francisco in July 2011 have been unanimously approved with minimal editorial changes.

Mr. Sebestyen inserted a "Secretariat's Note" on the planned update of the TC39 website. That note contains the reflection of the CC at their October meeting on the planned update of the TC39 website. The new version of the minutes have been stored as Ecma/TC39/2011/046 Rev2.

4 Status Reports

4.1 Report from Geneva (incl. IPR issues)

Mr. Sebestyen gave a short report, basically concentrating on the outcome of the CC meeting in October and what the meeting had to say about TC39 activities.

Note: On the ECMAScript Trademark matters verbally in the meeting he did not specifically report that it was on its way in Switzerland, the EU, USA, Korea and Japan. In Switzerland we have the mark, in the EU, US, Korea and Japan it is being progressing. As soon as there is significant new information he will let TC39 know.

Mr. Sebestyen points out about the internal Ecma TC39 documentation and archival is still not up to the requirements as they should, but since the last meeting some progress has been made.

At the last TC39 meeting it was decided:

- The Secretariat will get in static HTML format copies (snapshots) of the ES Wiki on a regular basis. **David Fugave** has volunteered in the September 2011 TC39 meeting to do it (maybe with some help within Mozilla). The requirement is that no special software should be needed to present the content. **(This has not happened yet…David is on paternity leave.)**
- Also the "ES Discuss" list should be archived. **Patrick Ch.** from the Ecma Secretariat has subscribed to it. "ES5 Discuss" also exists (maintenance), and also a "Test262 discuss". The "Discuss" lists are being archived by Patrick. The archive is private to the Ecma Secretariat. Information from it to any Ecma member can be provided on request. Since the information is not part of the official Ecma archives it is not part of the Ecma internal website.
- Archival is needed from the practical point of view at least until possible patent issues may come up (around expiration dates of the essential patents in the standard). This is about 20 years.

Mr. Sebestyen then reported about the TC39 relevant discussions in the October 2011 CC meeting:

He said that **Ms. Valet-Harper** brought up the point that in the draft TC39 minutes from the September meeting she has found a formulation that may cause problems: "It was also agreed that the TC39 RANDZ standardization goal should be put onto the TC39 website, when the update is being done."

There was a discussion in the CC how this point should be reformulated. The CC basically agreed that it was ok to state that the "TC has the intension (or goal) to develop a royalty free standard, such as for important web standards", but of course no guarantee by Ecma can be given that this has been actually achieved. In other words the word "practice" should be avoided. The CC then requested that the new text should be first shown to the CC, before putting it up on the TC39 web page. As result of this "practice" in TC39/2011/046 rev2 was replaced by "standardization goal".

Ms. Valet-Harper added that the CC is ready to help TC39 to find the best formulation.

Mr. Sebestyen reported about the CC discussion on TC39 request to extend the TC39 experimental software copyright policy:

"Work in this area has exposed a potential problem "*How do we deal with the test contributor who is not a member of Ecma nor employed by an Ecma member*"?

The CC requested clarification from TC39 if the current draft TR already contains such modules, or if this is an issue for the development of new test modules after approval or also for ES 6? **Mr. Sebestyen** said, that to his knowledge there are no such 3rd party contributions in the current TR, but in future work it may. Anyway, this will be verified at the next TC39 meeting and reported back to the CC."

TC39 has confirmed that the current version of the TR on 262Test does not include 3rd party software copyrights. But in future developments it may.

Another question that was raised by TC39*: Can "open source components" with no identified source ("freely available software") become part of an Ecma standard*? There is some thought by TC39 members to include such components in the new work (e.g. In ES 6).

Mr. Sebestyen said that: "*according to his reading it appears that the current software contributors' license could also be applied to identify third party contributors, who could fill in and sign such a form with the contributors*. He thinks that the "open source components", when nothing is known from the origin, no body is taking any responsibility for the code etc. is more problematic.

TC39 has confirmed that the understanding of **Mr. Sebestyen** was correct. For future version of both the ES Standards and the ES262 Test 3rd party and open source software

contributions are possible. Regarding urgency in the matter, certainly the publication of the ES5.1 TR – after the planned GA approval in December 2011 – is the highest priority.

Mr. Auber then invited TC39 representative(s) (**Mr. Horwat** ???) to the upcoming GA meeting, when the above should be reported and discussed, though detailed discussions on the GA level are not expected, but more in the Software Copyright ad-hoc Group. There TC39 members' technical and legal representatives are welcome.

"*The CC has discussed this matter. A first observation was that it would be better if the 3rd party contributors became Ecma members, not only because we have a policy already for that, but Ecma needs new active members. It was agreed that contributions from non-members should be an exception. It was also agreed that such non-member contributors should not have less obligations than Ecma members.*"

This last point was briefly discussed in the meeting. It was brought up that Ecma may look into the matters of having individual memberships. It was also mentioned that fees for that have to be minimal. It was also mentioned that W3C has a "Community Group" process (for pre-standards) that might be looked at.

4.2 Report of the status for a Technical Report on interoperability/conformance tests

4.2.1 Prototype Website (http://test262.ecmascript.org and http://test.w3.org/html/tests/reporting/report.htm

Ecma/TC39/2011/052 "Test262 Status Report, November 2011" by **Bill Ticehurst**, November 2011 was given. Further good progress since the September meeting, some bug removed, a bunch of new tests have been added (11,016 tests so far). Microsoft and Google had a good co-operation on Test262. But there are some items still open - those that are not covered in the current tests yet. Microsoft declared that unfortunately their budget has been cut to put further work into the tests and therefore they have requested other TC39 members to jump on with further tests.

Mr. Sebestyen reported that the 2 page TR has been prepared for the December GA, now according to the Ecma TR format. Successful correspondence took place between the Ecma Secretariat (**Patrick**) on the TR template and the Editor. For the actual URL link a note points out that this has to be done in the final editing and publication process. That will be done after the approval of the TR. The document is now on the Ecma private web site for approval in the December 6, 2011 GA.

Ms. Auber pointed out that Ecma has an approved experimental software copyright policy that needs to be applied to the TR. In practice, the GA will approve the document with the principles how the final editorial changes should be done, but those actual changes (e.g. correct URL in the TR, change of copyright notice in the code) in line with the policy will be done after the approval in the publication process.

It is still the plan that with the approval of the TR the work will continue and also ES6 Tests will be gradually added to the current set.

4.3 Report from the ad hoc on Internationalization standard.

4.3.1 Review of proposed draft standard

On November 15, 2011 there was a separate ad-hoc group meeting on internationalization. **Norbert Lindenberg** and **Nebojša Ćirić** reported on it. The results of that meeting had been published as Ecma/TC39/2011/051 "ECMAScript Globalization API" Ecma/TC39/2011/054 "Notes of the ad hoc meeting on Internationalization, 15 November 2011".

As reported in the September meeting the spec is not ready for a GA approval in December 2011. The group are still working on a draft document. The latest revision is available at:

http://wiki.ecmascript.org/doku.php?id=globalization:specification_drafts

It was mentioned that additional testing was needed. The minutes of the November 15 meeting will be added to this minutes.

4.3.2 Multi-system prototype testing

4.3.3 Next steps

Next steps:

In the TC39 meeting the following has been decided:

Go for an Ecma General Assembly approval of the standard in June 2012. This delayed the original plan by about 6 months.

An immediate fast-track submission to JTC 1/SC 22 is planned.

^^^^^^^

Nebojša Ćirić's Notes related to Globalization Ad Hoc group items:

- *Final document has to be ready by the end of February 2012*
- *Comments from TC39 meeting incorporated into a final document by mid April (hopefully the comments are minimal)*
- *ES 262 test harness should be used - we should talk to the test harness maintainer (he is on parental leave as of yesterday, but Bill Ticehurst, Microsoft, is a good alternative contact)*
- *Plan for test 262 should be presented at the March meeting (TR for separate spec, but TC39 needs to work it out - what's exactly needed?)*
- *Add an Annex to the draft with informal/explanatory part of the doc (again, by February)*
- *Norbert gave Internationalization group status update as item 4.3*
- *Norbert created slide deck based on our 11/15/2011 meeting at Google, which he is going to present as item 5.3 of the TC39 meeting agenda*

Technical presentation and discussion (item 5.3)

- *setDefaultLocale on Globalization (implement a way to change default locale)*
- *Move localeList into options parameter?*
- *Sync up with HTML5 time element (http://css.dzone.com/articles/html5-time-element-returned, http://lists.w3.org/Archives/Public/public-html/2011Nov/0011.html)*
- *Globalization namespace maybe taken by jQuery plugin*
- *Use modules? (may be done ahead of standard) - or use other namespace - they are proposing Object.System but will have to hash it out on the list*
- *API too Java like? Maybe implement shortcuts without user needing to new DateTimeFormat to format*
- *We need to initiate discussion on es-discuss about namespace and API*
- *Supplementary characters - some things are in UTF-16 some are not - specify that our algorithm uses UTF-16 encoding*
- *Introduce ValueError to core language? (maybe use TypeError, or use .error property of exception object to detect/change)*

^^^^^^^^

From: cira@google.com
To: brendan@mozilla.com
CC: es-discuss@mozilla.org
Sent: 11/17/2011 2:41:11 P.M. Eastern Standard Time
Subj: Re: Nov 16 meeting notes

> Internationalization presentation.
>
> Allen: Can a web developer reasonably depend on his webapp working the same in a given locale on any conforming browser?
> Answer: No.
> MarkM: Are there specific areas where it's possible to pin implementations down more?
>
> Alex: Wants a way to globally set a default localeList based on application-specific data.
> Long debate. Not possible as defined currently. Have the Globalization object default to getting a localeList from one of its user-settable properties if not provided in a function call?
> MarkM: Doesn't want mutable globals (other than ones that are set up once and then frozen).
> Alex: If you don't allow mutating the default, applications will still store the locale in a global and just pass the value of the global to every method that takes a locale, resulting in a more wordy program.
> Waldemar: Sometimes users will want to change their locale from within the application.
> Waldemar: While having either a mutable default locale or having no default locale would be ok, having a default immutable and implementation-dependent locale would be a problem, as it would be hard to usefully rely on such a thing.
>
> Brendan: Looks like the identifier "Globalization" is used in existing JS libraries. Not sure in what capacity yet.

I was looking a jquery-ui, whre Globalization appears to be a local var name. But there are other hits via codesearch.google.com to consider.

I also objected to gunning for early standardization with under-specification. ECMA-262 tends to specify for interoperation because web content can't choose the browsers it runs in, and fall-back is not always graceful or even possible.

Maybe we can get two distinct (not both based on ICU) implementations done enough to interop-test by next March. I'm sceptical, and we should commit to doing the testing and removing under-specification that will bite developers and browser implementers, even if that takes us past next Spring's GA.

The issue of the G11N API being too Java-esque came up. We agreed to address it on es-discuss, with TC39ers beyond Allen participating.

I've started two threads on es-discuss, one about the namespace the other about the API concerns. Thanks for the input.

--
Nebojša Ćirić

4.4 Update of TC39 Web Page at Ecma home page

From the September 2011 minutes: "*It was decided to update the TC39 webpage.* ***Mr. Neumann*** *will send the update instructions to the Ecma Secretariat. But first he wants to know what has happened to his earlier update instructions*"

The Secretariat has checked this, and the earlier update instructions were apparently not received.

This was discussed with **Mr. Neumann** directly but was not part of the TC39 meeting.

4.6 Doug Crockford presentation on feedback from talks at Silicon Valley Code Camp and RegExp `quasi literals`

See **Waldemar Horwat'**s attached notes.

4.7 Status of ES 6 Draft Specification

Mr. Wirfs-Brock has presented the latest draft from the Wiki. It has the date November 7, 2011. The draft is marked with "Rev. 4". He explains all the changes he has made to the previous version and requests TC39 members for feedback.

5 Discussion of ES harmony (technical contributions are available and can be found on the ES wiki)

There was a general discussion of what should happen with the new contributions that are coming in to TC39 meetings, but which were out of scope of the "frozen" ES 6 specification – that was agreed by TC39 at its May 2011 meeting. There was a general agreement that there is a difference between "ES6" (the next official ECMAScript version) and "ES Harmony" which may also includes new functions that may go beyond the scope of the agreed ES6. It was agreed that those should also be discussed (maybe with a lower priority) but under a different agenda point like "ES Next". Of course care has to be taken about possible interaction of functions in the two above categories.

For more technical details see the attached note of **Waldemar Horwat** below.

In what follows is a list of technical items discussed.

The colour code means the following:

"red": discussed and solved

"yellow": discussed but not solved yet

"no signal color": the item has become irrelevant.

5.1 review of i18N technical content of proposed standard

See 4.3

5.2 Math and String extensions: Quick update on library additions

5.3 Modules and Binary Data:

5.4 fine points of maps and sets

http://wiki.ecmascript.org/doku.php?id=harmony:simple_maps_and_sets

5.5 Array extras and pushAll

http://wiki.ecmascript.org/doku.php?id=strawman:array_extras

http://wiki.ecmascript.org/doku.php?id=strawman:array.prototype.pushall

5.6 batch-assignment operator

http://wiki.ecmascript.org/doku.php?id=strawman:batch_assignment_operator

5.7 fine points of for-of ((ES-discuss))

http://wiki.ecmascript.org/doku.php?id=harmony:iterators

5.8 Direct Proxies

http://wiki.ecmascript.org/doku.php?id=strawman:direct_proxies

5.9 Proxy.attach/Proxy.startTrapping

<http://wiki.ecmascript.org/doku.php?id=strawman:direct_proxies>..

5.10 Prototype implementation that builds upon FF7 proxies is available

<http://code.google.com/p/es-lab/source/browse/trunk/src/proxies/DirectProxies.js>

5.11 function proxy prototypes

http://wiki.ecmascript.org/doku.php?id=strawman:function_proxy_prototype

5.12 derived getPropertyDescriptor and getPropertyNames traps

http://wiki.ecmascript.org/doku.php?id=strawman:proxy_derived_traps

5.13 handler access to proxy

http://wiki.ecmascript.org/doku.php?id=strawman:handler_access_to_proxy

5.14 defineProperty reject behaviour

http://wiki.ecmascript.org/doku.php?id=strawman:defineproperty_reject_behavior

5.15 refactoring reject behavior (Mark's idea of introducing a boolean-valued [[Set]] internal method)

http://wiki.ecmascript.org/doku.php?id=harmony:proxy_defaulthandler#alternative_implementation_for_default_set_trap

5.16 proxies with fixed (non-configurable) properties

http://wiki.ecmascript.org/doku.php?id=strawman:fixed_properties

5.17 New class literal alternatives (Brendan)

5.18 Object exemplars (Allen)

5.19 Decoupling [ ] and property access (Allen)

5.20 Array Destructuring and length

6 Date and place of the next meeting(s)

January 18-19, 2012 hosted by Yahoo (Silicon Valley, CA)

March 28-29, 2012 Google (Silicon Valley)

May 23-24, 2012 North-Eastern University (Boston)

July 25-26, 2012 Microsoft (Redmond)

September 24-25, 2012 Mozilla (Silicon Valley)

November 28-29, 2012 Apple (Silicon Valley)

7 Closure

The TC39 Meeting ended at 4:30 PM on 17 November 2011. **Mr. Neumann** has thanked the meeting participants for their good contributions, constructive discussions and the co-operative spirit of the group.

The group expressed appreciation to Apple and to **Oliver Hunt** for hosting the meeting and Ecma international for hosting the TC39 dinner on the 16th November in Cupertino, CA.

Item 5 Attachment

Waldemar Horwat's Meeting Notes from November 16:

To: es-discuss@mozilla.org
Sent: 11/16/2011 8:19:29 P.M. Eastern Standard Time
Subj: Nov 16 meeting notes

Here are my rough notes from today's meeting.

Waldemar

IPR discussions

Test262 status

Internationalization: due by March meeting if we want to try for June GA

Doug Crockfords's presentation at Silicon Valley Code Camp. User feedback gathered by Doug:

* Users like quasis.

* Half of the users vehemently opposed large syntax changes (paren-free etc.) that change syntax merely to add a different way of doing largely the same thing. They know that they don't have to use the new syntax, but they also don't want to see it in code they read.

* Virtually all users hated block lambda.

* Users not looking forward to new class syntax but not as opposed.

* Users liked the ... operator.

* Users would like some functionality to gather stack traces to do things like remote application diagnosis.

* Users generally liked modules.

* Confusion about map. Particularly for people who work on (geographic) maps.

* Requests for Object.flatten (collapse prototype chain into one object), copy, and deep copy.

* Users support elective use of Unicode characters as alternate forms of syntax tokens. Examples: ≤, λ, etc.

* Some users wanted a NoSuchMethod trap on Object.prototype.

Allen's update on ES 6 Draft Specification

Allen: Wanted to allow an identifier named "arguments" only in the case where it's the ... parameter: function f(x, y, ... arguments) {...}
Objections: Unnecessarily complex. This isn't the same as the deprecated meaning of arguments because it doesn't include x and y (and also is an array instead of an arguments object).
Consensus: Don't make a special exception for use of the identifier "arguments" here.

```
function f(a, b = g()) {
  function g() {return a}
}
```
What happens here?
Waldemar, DaveH, and several others: b's initializer doesn't see the internal definition of g because the call to g is outside the braced scope. A forward reference here is counterintuitive.

Oliver: What about these?

```
var x = 1;
function f(a = x) {
  var x = 5;
}

function g() {return "g1"}
function f(a = g) {
  return a();
  function g() {return "g2"}
}
```

MarkM: Two-scope model (one scope for parameters, an independent inner scope for let, const, and var bindings). This model is upwards-compatible with ES5.1 strict because the latter disallows shadowing of a parameter by a binding in function scope.

Allen: Two-scope model, but with inner scope prohibited from shadowing parameters.
MarkM: Likes this. Any program that is accepted works with either of the two-scope model intuitions.

Waldemar: What about this?

```
var d = 12;
function f(a, b = a, c = d, d = 0) {...}
```

Do we have let, let*, or letrec semantics for a, b, c, and d?

Consensus: letrec with temporal dead zone, so the above would be an error when evaluating c's initializer because d hasn't been bound yet.

DaveH: Concerned about let temporal dead zone semantics (in general for let).

Waldemar: Dead zone very unlikely to come up for parameters in practice. The compiler can readily see if parameters are forward-referenced in uses. Of course, it is possible to come up with obscure cases that demonstrate the dead zone:

```
function f(a, b = function(){return d}, c = b(), d = 42) {...}
```

Waldemar: Point of order. Concerned about a number of items not approved in May appearing on the ES Harmony agenda for this meeting (section 5 of the meeting agenda). While Waldemar likes some of them and is as eager as anyone to discuss them, it would be better to separate them into a separate heading such as section 6 to make it clear to us and anyone watching whether the proposals are for ES.Next or for a future revision of ECMAScript. When posting agenda items, please indicate whether they're for ES.Next or for future revisions.

Internationalization presentation.

Allen: Can a web developer reasonably depend on his webapp working the same in a given locale

on any conforming browser?
Answer: No.
MarkM: Are there specific areas where it's possible to pin implementations down more?

Alex: Wants a way to globally set a default localeList based on application-specific data.
Long debate. Not possible as defined currently. Have the Globalization object default to getting a localeList from one of its user-settable properties if not provided in a function call?
MarkM: Doesn't want mutable globals (other than ones that are set up once and then frozen).
Alex: If you don't allow mutating the default, applications will still store the locale in a global and just pass the value of the global to every method that takes a locale, resulting in a more wordy program.
Waldemar: Sometimes users will want to change their locale from within the application.
Waldemar: While having either a mutable default locale or having no default locale would be ok, having a default immutable and implementation-dependent locale would be a problem, as it would be hard to usefully rely on such a thing.

Brendan: Looks like the identifier "Globalization" is used in existing JS libraries. Not sure in what capacity yet.

Timezone faithfulness across changes in the timezone laws issue.

Contains: Should the position parameter match IndexOf or lastIndexOf?

Array and string methods will continue to treat strings as pure sequences of 16-bit values.

randomInt is based on Math.random.

Will add erf/erfc/gamma if they're in the common math libraries.

Map/Set:
Size property should be a getter property with no matching setter. It's defined on the prototype.
What is its name? size, count, or length? Decide on es-discuss.

Default for-of iteration on Sets is clear. On Maps, should the default iteration return the keys or key-value pairs?
for-in iteration on Maps and Sets doesn't do anything useful.
You can ask for the other kinds of iteration on a Map by running a method (not a getter because it's not idempotent).

What is the order of enumeration? Insertion order (and without any exceptions for integral keys --
MarkM: who would define a language feature like that ;-)?

Does creating an iterator snapshot the object?
Brendan: If enueration order is insertion order, a Map or Set would make a useful worklist abstraction as long as new elements do appear in iteration.
Waldemar: What are the alternatives? Snapshotting would be heavy-weight.
DaveH: The alternative is to throw from the iterator if the object has been modified since the previous iterator call.
MarkM: Now in the worklist semantics camp, merely to avoid weird iterator state.

Argument over whether Map.prototype is a Map. We want to avoid the mess caused by Date.prototype being a Date (MarkM: It provides a hidden communication channel that cannot be frozen).
What happens when you extract Map.prototype methods and apply them to an object that is not a Map?
Type error.
Allen: Wants the prototype to be an instance of the class for consistency.

Waldemar, MarkM, DaveH: The prototype is not an instance of the class. Empirically, the only things that act as instances of user-defined classes are the things that inherit from the prototype, not the prototype itself.
Will continue on es-discuss. However, we agree that there should be no mutable private state in the Map/Set constructors.

Also need to fix Date.prototype. Its private state should be frozen as well.

DaveH: Use === or egal? Distinguishing ±0 as separate slots in a Map or Set will confuse users.
MarkM: What about defineOwnProperty? Same problem would appear there.

_____

es-discuss mailing list
es-discuss@mozilla.org
https://mail.mozilla.org/listinfo/es-discuss

^^^^^^^^^^

Waldemar Horwat's Meeting Notes from September 17:

From: waldemar@google.com
To: es-discuss@mozilla.org
Sent: 11/17/2011 7:40:26 P.M. Eastern Standard Time
Subj: Nov 17 meeting notes

Array destructuring and length:
let [a, b, c, d, ... r] = {2: 3} <| [1, 2]
Obvious: a is 1; b is 2.
What are c, d, and r?
c = 2.
d = undefined.
r = empty.

Fixed property destructuring doesn't rely on length.
Vararg r destructuring uses length.
The semantics of length will match that of slice.

Allen: We may upgrade ToUint32 to ToInteger in various array semantics.

What should the semantics be if we allow fixed properties in the
middle of a destructuring?
[a, ... r, b] = [42]
What are the values of a, r, and b?
a = 42
r = []
b = undefined

Brendan:
[a, ... r, b] = [, 43] <| [42]
What are the values of a, r, and b?
a = 42
r = []
b = 43 or undefined?

Array.from discussion: What happens if you subclass Array?

```
Subarray = Array <| function() {...}
Subarray.from(arraylike)

DaveH:
Array.from = function(x) {
  var result = new this();
  for (var i = 0, n = x.length; i < n; i++)
    result[x] = x[i];
  return result;
}

Array.of = function(... x) {
  var result = new this();
  for (var i = 0, n = x.length; i < n; i++)
    result[x] = x[i];
  return result;
}
```

The above should skip holes.

MarkM: Now these functions are this-sensitive and will fail if extracted and called indirectly.
DaveH: Replace 'new this()' with 'new (this || Array)()' above.
MarkM: Of all of the static methods in ES5, not one of them is this-sensitive. The simple extraction of a static method fails, thereby making static methods not be first-class. If Math.sin did this, you couldn't map it over an array. With this, you can't map Array.of over an array.
Doug: Concerned about the use of the word 'of'; confusion with for-of.
Wild debate over class hierarchies and class-side inheritance.
Deferred Array.from and Array.of due to concerns over this-sensitivity until we figure out a proper class-side abstraction mechanism.

Array.from(a) is superfluous because it's expressed even simpler as [... a]. DaveH withdrew it.

Array.pushAll:
Debate over whether this is a workaround for poor implementations of using Array.push with spread or apply, or whether we should directly have a second set of methods.
Brendan: Let's implement spread and optimize it first. Later we can always add pushAll if it's needed. "This isn't ... paving cowpaths; this is a mountain goat that went too high".

DaveH: Very opposed to .{ .

Cut 'fine points of for-of' from this meeting due to time.

Batch assignment:
Is this ES6 or ES7? This is new, not discussed in May.
Can't discuss batch assignment without also discussing .{ .
Was .{ part of the May object literal proposal?
MarkM: Two kinds of .{ collisions to worry about. The object literal just statically disallows them. .{ can have run-time property collisions.
DaveH: Like the functionality but not the .{ syntax.

Example from .= page:

```
let element = document.querySelector('...');
element.{
  textContent: 'Hello world'
}.{
  style.{
    color: 'red',
    backgroundColor: 'pink'
  }
}.{  // back on element
  onclick: alert
};
```

Waldemar: Can you replace }.{'s with commas? Brendan: Not in general.
}.{'s do property redefinitions on property name conflicts, while
commas produce errors on conflicts.
Waldemar: Can you distribute the middle section above into the following?

```
}.{
  style.{color: 'red'},
  style.{backgroundColor: 'pink'}
}.{  // back on element
```

Answer: Maybe.

DaveH: Bind operator syntax strawman.
softBind strawman.
[A bunch of different discussions going on simultaneously, which I
couldn't track.]

Direct Proxies slide show.

Discussion about what hidden or implementation properties are passed
from the target through a direct proxy and how a proxy handler would
find out about all of them. The author of a proxy needs to keep up to
date about picking the correct target as we add hidden properties.
For example, to make an Array-like proxy object, a proxy should start
with an Array instance as the proxy target. Same with Date, etc.
Allen: There's no way to bootstrap -- can't define an Array-like proxy
if you don't have an Array target to start with.
Discussion about proxying the [[class]] name.

No more fundamental vs. derived traps. (Almost) all traps default to
the target object's behavior if not overridden. An exception is the
construct trap, which by default calls the call trap instead of
forwarding to the target object.
Allen: Should just pass through to the target.
Allen worried about other derived traps.
Waldemar: Always defaulting to the target will prevent us from ever
defining new non-leaf traps in the future, as that would break
existing proxies. For example, if we have a current trap API where
the proxy defines only the trap GET, and we later wish to evolve the
language to refactor the API to call the derived trap HAS followed by
GET, where an object's HAS is defined in terms of GET, then defaulting
to the target will break proxies because HAS will invoke the target's
GET instead of the proxy's GET.
MarkM: This is forwarding vs. delegation. The issue applies to many

traps, not just call. All non-leaf traps should be resolved in the same way.
Allen: Get rid of non-leaf traps (i.e. make them nonoverridable, allowing proxies to override only the leaf traps into which the non-leaf traps decompose).
MarkM: Why?
Waldemar: Are there any derived traps that have additional intelligence not expressible via the leaf traps which they call?
Allen: Future-proofing is not important because we're unlikely to change the proxy API in the future.
Waldemar: Counterexamples: Classes (if they're not just sugar), guards, value proxies, ...
Sam, MarkM: The language provides a prototype handler with reasonable defaults for derived traps. Proxy handlers derive their handlers from that prototype. When the language evolves to create new traps, the language's prototype handler evolves in lockstep to keep existing proxies working.
Discussion about double-lifting (making the proxy handler also be a proxy so it can abstract over the proxy trap API) and future-proofing membranes.

Make target always be the first handler parameter instead of always being the last? That would make it match the Reflect API.
What does receiver do in Reflect.get/set/call(target, receiver, ...)?
Make target first and receiver last in the handler API. The Reflect API can drop the extraneous receiver parameter.
MarkM: That doesn't make sense for Reflect.call. The order should be Reflect.call(target, receiver, args), not Reflect.call(target, args, receiver). Note that args is an array, not a spread list of arguments.
Discussion about what Reflect.call should be called. The name "invoke" was invoked. "apply" might also apply here.

Object.preventExtensions is a leaf trap and calls handler.protect('preventExtensions').
Object.seal and Object.freeze: Should these call handler.protect('seal') and handler.protect('freeze')? Replace them with separate handler traps? Replace them with calls to a large number of separate traps that individually seal or freeze properties?

MarkM: Why not define call behavior as target function behavior instead of having a call trap?
Waldemar: That would break double lifting.

Nonconfigurability invariants:

Waldemar: get/set can still get and set own properties even on frozen objects with fully frozen prototype chains.
Sam, MarkM: True. Unfortunately it's too hard to enforce this invariant.

Tom: Use a null target to indicate a permanently "virtual" object.
Brendan: Proxy.DonJuan

Tom: Replace synchronization actions with throws if the synchronization actions would make any modifications to the target object.

Discussion of __proto__.
Brendan: It's a real use case in object literals.

Proxy.stopTrapping:
Can approximate this by deleting all handler bindings from the handler.
MarkM: Swap arrow by making the proxy be the target and target be a proxy?
Consensus: Dropped stopTrapping

Refactoring prototype climbing in the spec.
Refactoring:
Brendan: Don't get rid of Object.getPropertyDescriptor and
Object.getPropertyNames even if they aren't traps.

Bikeshedding Proxy.for and Proxy.create names.
Why isn't Proxy a module?
if it's a module, can't use Proxy.for unqualified because it's a keyword.

Tom: Direct proxy refactoring takes care of the set/put/canPut chaining problem.

Should we have a defaultValue trap?
Brendan: Leave it out for now. Revisit when we do value proxies.

Attach:
If target is a proxy, call a new trap.
Tie ability to attach to object extensibility.
Waldemar: Won't work for host objects that need to be both secure and
extensible.
MarkM: This is especially evil in capturing function arguments. If
Alice gives Bob and Cathy a plain ES5 function, Cathy can't peek at
the arguments Bob passes to the function when he calls it. With
Attach, Cathy can intercept Bob's calls to Alice's function and,
worse, make the function do something different for Bob.
Discussion about whether this should be done. A number of us want
value observers and feel that proxies are less useful without them.
DaveH: There are many things that are wanted that didn't make the ES.next cut.

Consensus: Direct proxies (other than startTrapping/attach) accepted
for ES.next to replace older proxies proposals and strawmen.

2012 meetings:
Jan 18-19 at Yahoo
Mar 28-29 Google
May 23-24 Northeastern University, Boston
Jul 25-26 Microsoft Redmond
Sep 24-25 Mozilla
Nov 28-29 Apple

_____