From: **Waldemar Horwat** waldemar@google.com
Subject: Meeting notes
Date: November 5, 2009 at 6:21 PM
To: e-tc39@ecma-international.org

Here are my partial notes from today's meeting. I wasn't taking notes during other parts of the meeting.

Waldemar

Fix my name in last meeting's minutes.

Interpolation flattens everything and delays context until the final use. This causes the trouble case:

```
d = interpolate("AT&T < $price");
page = interpolate(<p>$d</p>");
```

Make interpolation compile into closures that reference properties of an object instead of using eval each time?

E quasi-literals require naming of context on each interpolation.

Feedback on object initializers strawman:

These look like classes. The thing that's missing is unforgeable nominal typing, which should be there as well.

The constructor declaration that defines a constructor that can't be called as a function should be more flexible: allow both behaviors.

Much unhappiness about constructor declarations as currently proposed. The constructor declaration doesn't let one interpose scopes between the constructor and the created object -- no way to use scopes to create instance-private environments. Also no way to do some local, preliminary calculation that is then used to initialize several const properties (same problem as with C++'s field initializers).

Brendan: use "class" as the keyword for enhanced object initializers.

Problem with private: Many don't like the syntax.
Also the proposal defines private things as lexical bindings while the public things are object properties. They're used and invoked differently, which would prevent people from refactoring code to turn public things private or vice versa.

const implying nonenumerability is weird and gets in the way of refactoring from mutable to const members and back.

Mark and Tom will do traits on top of classes-as-sugar.

Hash brouhaha:

Allen's proposal doesn't address Mark's covert channel concerns; the multiple hash functions just introduce more covert channels.

Observation: are bare hashes commonly used, or are they mostly hiding inside hash tables, ephemeron tables, etc.? A bunch of us think it's inside hash tables, ephemeron tables, etc.? A bunch of us think it's the latter, reducing design pressure on getting bare hashes fast.

Judy Brewer - Directory of Web Accessibility Initiative at W3C
Michael: Aria is an attribute system for defining semantic meaning of things like user-developed checkbox chrome to tag them as being checkboxes.
No relevance to the core language that TC39 is working on.

Internationalization:
http://www.w3.org/International/wiki/JavaScriptInternationalization
Locales: Recommend providing just locale parameters to various ECMAScript methods without defining what exactly they do.
Mark Miller would prefer a completely specified behavior. Some others don't.
Locale data is about 5 megabytes.
Line endings are a non-issue. This item was spurious based on 0x0085.
ECMAScript requires Unicode 3 or later. Some implementations use exactly version 3. Implementations have the freedom to use later versions if they wish. OK.