

Web型グループウェア開発における技術標準動向と人材育成の課題

The standard trend in Web groupware development and training

竹迫 良範

Yoshinori TAKESAKO

サイボウズ・ラボ株式会社 〒112-0004 東京都文京区後楽 1-4-14 後楽森ビル 12F

E-mail: takesako@gmail.com

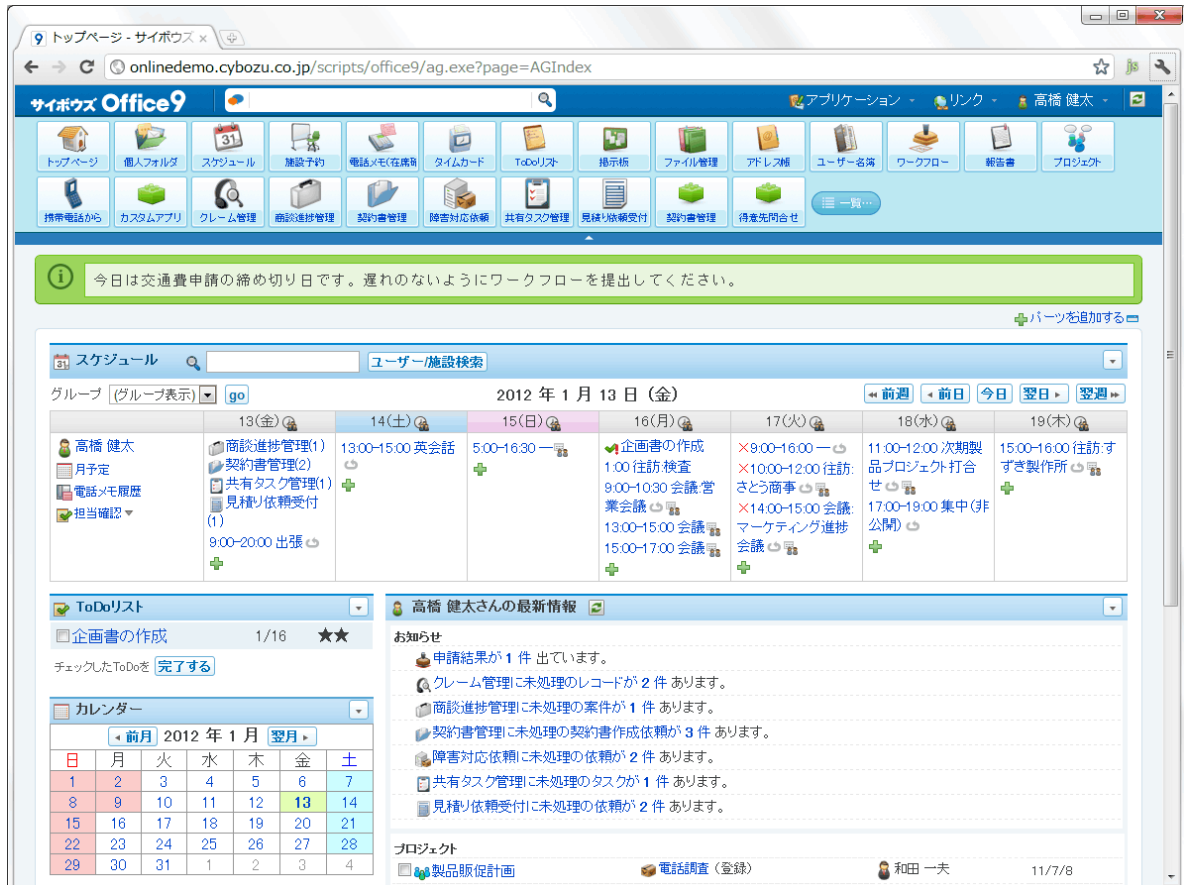


図 1: Web型グループウェア製品サイボウズ Office 9 のトップページ画面

1. はじめに

サイボウズ株式会社は、創業 1997 年から Web 型グループウェアを開発しているソフトウェア会社である。主力製品はすべて Web アプリケーションとして実装されており、専用のソフトウェアをインストールすることなく、Web ブラウザのみで利用できる (図 1)。

1.1. Web ブラウザの種類

現在、動作をサポートしている Web ブラウザの種類は、Internet Explorer、Firefox、Chrome、Safari の各バージョンと多岐に渡っており、各ブラウザの動作の違いが、開発上の大きな問題となっている (図 2)。

Windows版

- [Internet Explorer 7.0/8.0/9.0の設定方法](#)
- [Internet Explorer 6.0の設定方法](#)
- [Mozilla Firefox 7.0/8.0の設定方法](#)
- [Mozilla Firefox 2.0/3.0/3.5/3.6/4.0/5.0/6.0の設定方法](#)
- [Mozilla Firefox 1.5の設定方法](#)
- [Google Chromeの設定方法](#)

Macintosh版

- [Safari 5.0/5.1の設定方法](#)
- [Safari 2.0/4.0の設定方法](#)

図 2: サポート対象 Web ブラウザー一覧

2. Web ブラウザのバージョンアップの歴史

1994年にリリースされた Netscape Navigator では、テキストを点滅させる目的の<blink>を HTML 中に記載できるようにしていた。競争関係にあったライバルの Internet Explorer は、テキストをスクロール表示させる<marquee>や、音を鳴らす<bgsound>などを次々に導入した。当時ブラウザ間で全く互換性のない独自拡張がベンダー主導で積極的に行われる時期があった。

各ブラウザの挙動と見栄えの違いによって、多くの Web 制作者を悩ます結果となったが、W3C による HTML4.01 仕様の勧告と CSS の普及によって、各ブラウザ・ベンダーは標準準拠を意識するようになった。

Netscape は 2008 年に開発を終了したが、その後 Google Chrome の登場によって Web ブラウザのバージョンアップ競争が現在まで続いている (図 3)。

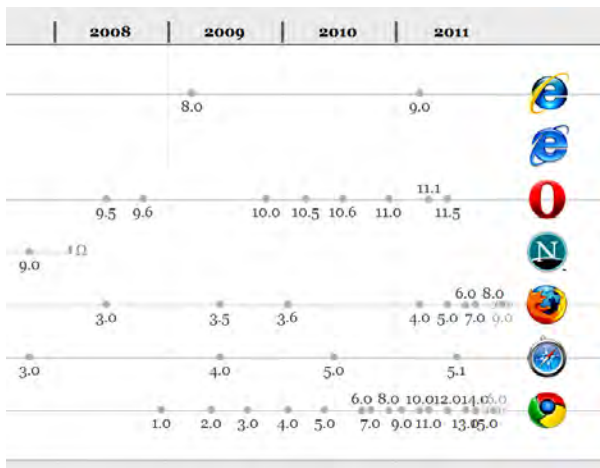


図 3 : Browser History Timeline

<http://meyerweb.com/eric/browsers/timeline-structured.html>

2.1. The Web Standards Project (WaSP)

WaSP は 1998 年に設立された Web 標準に沿ったブラウザの実装とコンテンツ制作を支援する団体である。

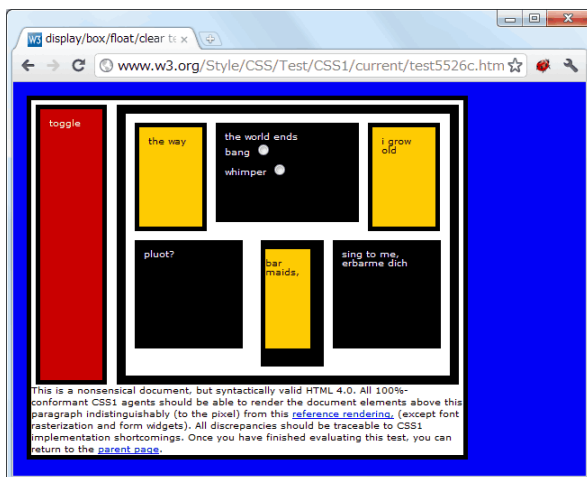


図 4 : 初めて作成された Acid1 Test

W3C が勧告した Web 標準にどれだけ準拠しているかを視覚的に確認できるテスト Acid1 Test (図 4) を初めて作成した。このテストは CSS1 Test Suite に取り込まれ、Web 標準普及への大きな前進となった¹。

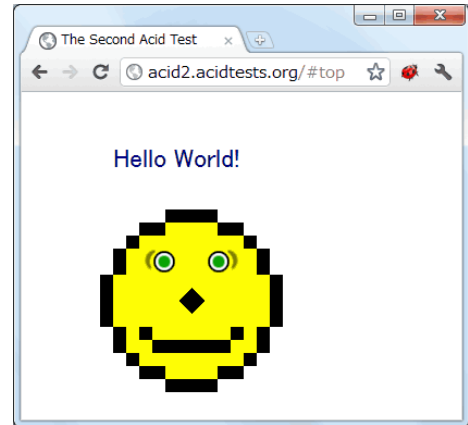


図 5 : Acid2 Test の表示結果

その他、主に CSS2.1 への準拠度を視覚的に確認できる Acid2 Test (図 5) も有名である。現在、デスクトップ PC 向けのブラウザではほぼ準拠しているが、Opera Mini 4 や iPod touch 2 などスマートフォン上のブラウザでは正しく表示されない実装がある。

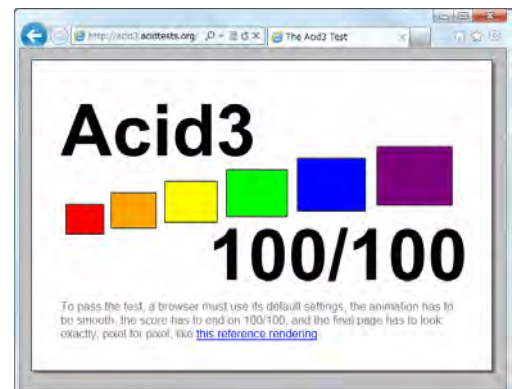


図 6 : Acid3 Test の 100 点満点結果

最近では Ajax 技術の普及によって、JavaScript や DOM を多用する Web アプリケーションが増えてきたため、主に HTML の DOM Level 2 や ECMAScript の挙動をテストする Acid3 Test (図 6) が 2008 年に公開された。

2.2. W3C CSS Working Group

このように視覚的に CSS のレンダリング結果を確認できるようにしたテストスイートは W3C の WG でも作成され、現在 9,600 件のテストが公開されている²。

¹ ES ウェブブラウザ通信 - Acid1 テスト

<http://shiki.esrille.com/2011/09/es-acid1.html>

² CSS2.1 Test Suite Results

<http://wiki.csswg.org/test/css2.1/results>

3. WHATWG と HTML5 の関係

WHATWG は Web Hypertext Application Technology Working Group の略で W3C とは別の団体である (図 7)。



図 7 : WHATWG の Web サイト

従来 W3C が進めていた XHTML2.0 の方向性と現場のニーズの無関心さに不満を持ったブラウザベンダー Opera、Mozilla、Apple の 3 社が集まり、独自に新しい HTML 関連仕様をまとめるため、2004 年に設立した。このときの仕様は Web Applications 1.0 と Web Forms 2.0 としてまとめられた。のちの HTML5 である [1]。

3.1. XHTML2.0 の終了と HTML5 の登場

結局 W3C は 2007 年に XHTML2.0 の策定をあきらめて、後方互換性を重視した新しい HTML の仕様を策定すると発表した。WHATWG は自分たちでまとめた仕様を HTML5 という新しい名前で作成してはどうかと W3C に対して提案し [2]、HTML Working Group が W3C 内に新設された。それに伴い W3C は HTML5 の仕様策定に注力し、XHTML2 Working Group を解散した³。

表 1 : HTML5 関連 API の実装状況

| 対応ブラウザ | Internet Explorer | | | Firefox | | Chrome | Safari | Opera |
|----------------------|-------------------|---|----|---------|---|--------|--------|-------|
| バージョン | 8 | 9 | 10 | 3.6 | 4 | 10 | 5 | 11.50 |
| Local Storage | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ |
| Session Storage | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ |
| Post Message | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ |
| Offline Applications | × | × | × | ○ | ○ | ○ | ○ | ○ |
| WebSockets API | × | × | × | × | × | ○ | ○ | × |
| Web Workers | × | × | × | ○ | ○ | ○ | ○ | ○ |
| Query Selector | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ |
| WebSQL Database (廃止) | × | × | × | × | × | ○ | ○ | ○ |
| IndexedDB Database | × | × | × | × | × | × | × | × |
| Drag and Drop | ○ | ○ | ○ | ○ | ○ | ○ | ○ | × |
| File API | × | × | ○ | △ | ○ | ○ | × | ○ |
| GeoLocation | × | ○ | ○ | ○ | ○ | ○ | ○ | ○ |
| History Management | × | × | × | × | ○ | ○ | ○ | × |

³ 2010-12-17: The XHTML2 Working Group is closed. <http://www.w3.org/MarkUp/>

3.2. HTML4 と HTML5 の違い

従来の HTML4 の文章 (図 8) を XML valid でない書き方が許容された HTML5 で書き直すと図 9 になる。

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN">
<html>
  <head>
    <meta http-equiv="content-type"
          content="text/html; charset=UTF-8">
    <title>HTML4.01 example</title>
    <script type="text/javascript">
      alert("Hello!");
    </script><style type="text/css">
      body{background:#decafe}</style>
    </head>
  <body>
    <p>HTML4.01 sample document<hr/></p>
  </body>
</html>
```

図 8 : HTML4.01 の文書サンプル

```
<!DOCTYPE html>
<html>
  <meta charset="UTF-8">
  <title>HTML5 example</title>
  <script>alert("Hello!");</script>
  <style>body{background:#eff}</style>
  <p>HTML5 sample document<hr></p>
</html>
```

図 9 : HTML5 の文書サンプル

従来の HTML の仕様では字句解析と構文解析のアルゴリズムが明記されておらず、HTML パーサの挙動がブラウザのバージョン毎に異なっていた (図 10)。

```
<!DOCTYPE html>
<title>HTML5 Syntax Checker</title>
<h2>Your Browser is...</h2>
<img
  :src = "opera8.gif"
  x=`src = "ie.gif"
  srcZ/"netscape6.gif"
  src#= "netscape7.gif"
  = /src = "html5.gif"
  'src^K= "safari.gif"
  /src = "firefox.gif"
  src = "others.gif"
  src = "lynx.gif"
/>
<hr/><!-- ^K is Vertical Tab '\x0B' -->
<p>http://www.w3.org/TR/html5/syntax.html</p>
```

図 10 : HTML5 ではパーサの動作も標準化

HTML5 は Web アプリケーションを記述するプラットフォームの仕様として JavaScript の API を多く定義している。いくつか API は元の HTML5 の仕様から分離した形で、個別にメンテナンスを行っている (表 1)。

4. ECMAScript の標準化

1996年3月にリリースされた Netscape Navigator 2.0 には Web クライアントサイドで動作するスクリプト言語 JavaScript の最初のバージョンが搭載されていた。当時 Netscape と Web 標準とブラウザシェアを争っていた Microsoft は JavaScript と互換性のあるスクリプト言語 JScript を搭載した Internet Explorer 3.0 を8月にリリースした。Netscape は JScript の発表に危機感を覚え、JavaScript の標準化を決意した。当初 W3C に JavaScript の標準化の話を持っていったが、断られてしまった[3]。Netscape は標準を買うことができる団体を求めて、ISO やその他標準化団体を回り、欧州の ECMA International に辿り着き、ECMAScript (ECMA-262) として JavaScript の言語仕様を標準化することとなった (表 2)。

表 2 : ECMAScript 各版の標準化動向

| 版 | 時期 | 概要 |
|---------|----------|---|
| - | 1996年3月 | JavaScriptの誕生 (Netscape Navigator 2.0) |
| 1 | 1997年6月 | ECMA-262 1st Edition Ecma Internationalで最初に規格化 |
| 2 | 1998年6月 | ECMA-262 2nd Edition (ISO/IEC 16262:1998) 以後、JIS X 3060:2000としてJIS規格化 |
| 3 | 1999年12月 | ECMA-262 3rd Edition ISO/IEC 16262:2002 (2nd Edition) 改訂 |
| 4 | 廃止 | ActionScriptとJavaScriptの統合と大幅な拡張を試みるが挫折 |
| 5 | 2009年12月 | ECMA-262 5th Edition を10年ぶりに策定 各ブラウザで実装されている仕様の追認 (内容的にはES3.1) |
| 5.1 | 2011年6月 | ECMA-262 5.1th Edition として Errata 等の修正 ISO/IEC 16262:2011 (3rd Edition) 改訂 |
| ES.next | 現在 | 新しいECMAScript HarmonyとしてTC39で仕様の拡張を検討中 |

Microsoft も標準化に参画し、JScript も ECMA-262 に準拠するようにバージョンアップされた。

ECMA-262 はその後 Fast-track 手続によって ISO/IEC 16262:1998 として国際標準として承認され、日本国内でも JIS X 3060:2000 として翻訳 JIS が制定された。

4.1. 消えた ECMAScript 4th Edition

2008年頃まで、改訂を目指し Adobe と Mozilla を中心に ECMAScript 4 の標準化に向けた準備が着々と進められていた。Adobe は ActionScript 3 の高速な JIT 処理系実装 Tamarin を公開し、Mozilla は JavaScript 2.0 に向けた言語仕様のバージョンアップを行っていた。

ECMAScript 4 は ActionScript と JavaScript の言語大統一を試みる野心的なプロジェクトであった。

ActionScript 由来の class 宣言や int 型、string 型、const 宣言が可能になり、type 宣言やテンプレート型も使うことができ、package、namespace、intrinsic 宣言など、ダイナミック言語の大規模開発で必要な要素はすべて ECMAScript 4 で取り入れられる予定であった[4]。

4.2. ECMAScript 5th Edition の登場

しかし、ECMAScript 4 の言語は後方互換性がなく、Web が壊れてしまうと懸念した他のブラウザベンダーから言語仕様の大幅な変更を受け入れてもらうことができなかった[5]。代わりに ECMAScript 3.1 をベースとした小さな仕様改訂 ECMAScript 5 が承認された。

4.3. ECMAScript5 の仕様忠実な処理系「lv5」

2011年、サイボウズ・ラボユースの鈴木勇介⁴ は[6]、最新の ECMAScript5 の仕様に忠実な処理系 lv5 を実装した。lv5 では Array.length に対する PropertyDescriptor の設定が可能になっている。これは他のブラウザで未実装な ECMAScript5 の新しい仕様の1つである (図 11)。

```
function testcase() {
  var arr = [0, 1];
  Object.defineProperty(arr, "1", {
    value: 1,
    configurable: false
  });
  try {
    Object.defineProperties(arr, {
      length: { value: 1 }
    });
    return false;
  } catch (ex) {
    var desc = Object.getOwnPropertyDescriptor(arr, "length");
    return ex instanceof TypeError && desc.value === 2 &&
      desc.writable && !desc.enumerable && !desc.configurable;
  }
}
```

図 11 : Test262, 15.2.3.7-6-a-112.js

TC39 では、ECMA-262 への仕様準拠度を確認するための Test Suite として Test262 を提供している。lv5 は、Test262 の 11029 件の全件テストのうち 11011 件 99.83% を pass している (表 3)。残り 18 件については Test262 自体のバグの可能性があり、Test262 の開発者にレポートを送り、数件のテストケースは修正の見込みである。

表 3: ECMAScript Test262 準拠度テスト結果

| | all | pass | fail | 仕様準拠度(%) |
|---------|-------|-------|------|----------|
| Opera | 11029 | 7276 | 3753 | 65.97 |
| Safari | 11029 | 10256 | 773 | 92.99 |
| Chrome | 11029 | 10611 | 418 | 96.21 |
| Firefox | 11029 | 10865 | 164 | 98.51 |
| lv5 | 11029 | 11011 | 18 | 99.83 |

4.4. ISO/IEC 16262:2011 の策定

日本では情報規格調査会の SC22 専門委員会内に ECMAScript アドホックを招集し、ISO/IEC 16262 の改訂に備えた。JSON 形式のサポート、Strict モードの追加、ISO8601 形式の日付対応、エラーの厳密化などの追加仕様部分を主に審議した。ECMA からの Fast-Track 手続により、JTC1 参加各国の SC22 mirror committee

⁴慶應義塾大学理工学部 3 年

が投票を行ったが、コメント付きの反対投票を行ったのは日本の JISC のみであった。投票コメントでは、セミコロン自動挿入や JSON の表記ゆれに対する許容について最近の実装を反映した柔軟性の改善、%演算子のゼロ除算時の振る舞いなど技術的な記述の明確化、URI の扱いの最新 RFC への準拠など技術的な改善を提案した他、出典として参照すべき仕様の追加、仕様の不整合箇所や誤記など編集上の指摘を行った。

5. Web 標準の課題

現在の Web アプリケーションの動作の根幹をなす JavaScript に関する言語仕様は、W3C ではなく、ECMA TC39 が ECMA-262 として仕様の策定を行っている。Web Socket の JavaScript 側 API⁵ は W3C が策定しているが、Web Socket の通信プロトコルは IETF が策定に関与している⁶。HTML5 や関連 API は基本的に W3C が策定しているが、WaSP や WHATWG のように別団体も過去に存在していた。現在の Web 標準の規格は一つの団体が主導して決めているものではない。実装の追認も含め、現実と理想のバランスを取りながら標準化を進めているのが現状である。複数の団体が将来のインターネットのビジョンを共有し、同じ目的を持って協調しながら仕様を策定していく必要がある。

5.1. Web 制作現場への標準化教育

Web 制作の現場では未だに CSS ハックと呼ばれている手法 (図 12) が使われている。ブラウザ毎に CSS の適用範囲や見栄えが異なるため、字句解析器の例外に依存した形で条件分岐を記述するという手法である。

```
body {
    color: red; /* all browsers */
    color: green¥9; /* IE6,7,8 */
    *color: yellow; /* IE6,7 */
    _color: orange; /* IE6 */
}
```

図 12 : CSS ハックと呼ばれる分岐手法

これはブラウザの実装に依存した形で、将来の動作が保証されたものではない。また、各ブラウザでの表示確認に余計な時間と手間がかかってしまう。今後 HTML5 と CSS2.1 の普及と制作現場への教育によって、このような CSS ハックが不要になることが望ましい。

5.2. Web セキュリティの問題

近年のサイバー攻撃でも Web システムが攻撃の対象となっている。Web ブラウザや Web アプリケーションのセキュリティはこれから益々重要となる。実装のミスだけではなく、CSRF のように仕様上の欠陥も脆

弱性となる。利用者が安心して Web アプリケーションを利用するためにも、仕様の策定段階からセキュリティを考慮した設計と確認作業が必要である。

5.3. 世界から見た日本のおかしな標準

日本の 12 時間表示の慣習で、午後 0 時というものがあるが、通常デジタル時計の表示は 12:00PM である。そのため、国際化を意識したソフトウェアでは、午前・午後ではなく AM/PM 表示を採用している (図 13)。

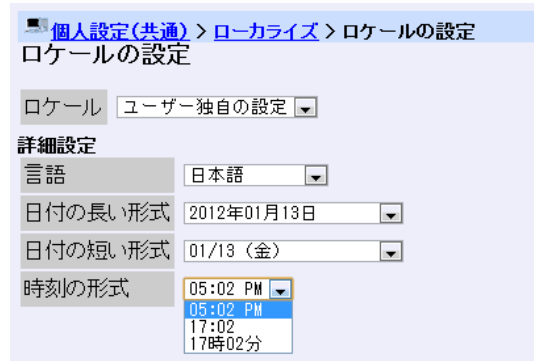


図 13 : 日時表示のロケール設定画面

日本の標準化の甘さが露呈した事件もあった。世界から見ると日本のローマ字の標準は ISO 3602:1989 である。これは「富士通」を「HUZITU」と表記することになり、へボン式に慣れている日本人には受け入れがたい。これは Web サイト所有者の法的実在性を証明する EV SSL の標準化において問題となった[7]。

6. さいごに

Web の技術は変化が早く標準策定のサイクルが速い。最新の標準動向に敏感な若い世代の人材を標準化団体の中に取り込んでいく施策も必要であると考えられる。

参考文献

- [1] W3C, 2011, HTML5 A vocabulary and associated APIs for HTML and XHTML (Working Draft) <http://www.w3.org/TR/html5/>
- [2] @IT, 2010, W3C は賞味期限切れ組織? <http://www.atmarkit.co.jp/news/analysis/201001/19/w3c.html>
- [3] Rob Larsen, 2011, ECMA-262 第 5 版 最新の JavaScript 仕様に関する調査 <http://www.ibm.com/developerworks/jp/web/library/wa-ecma262/>
- [4] 2007, Proposed ECMAScript 4th Edition <http://www.ecmascript.org/es4/spec/overview.pdf>
- [5] Chris Wilson, 2007, What I think about ES4. <http://blogs.msdn.com/b/cwilso/archive/2007/10/31/what-i-think-about-es4.aspx>
- [6] 情報処理学会, 2012, 第 53 回プログラミングシンポジウム予稿集 pp.171-176
- [7] 秋山卓司, 2009, EV SSL こぼれ話 (ISO 3602:1989 の罫) <http://yet2come.jp/2009/01/ev-ssl-iso-3602.html>

⁵ <http://www.w3.org/TR/websockets/>

⁶ <http://tools.ietf.org/html/rfc6455>