

# ECMAScript 6 Draft Hisotry Repo

ES6 DraftのGitリポジトリを作る

# 自己紹介

azu

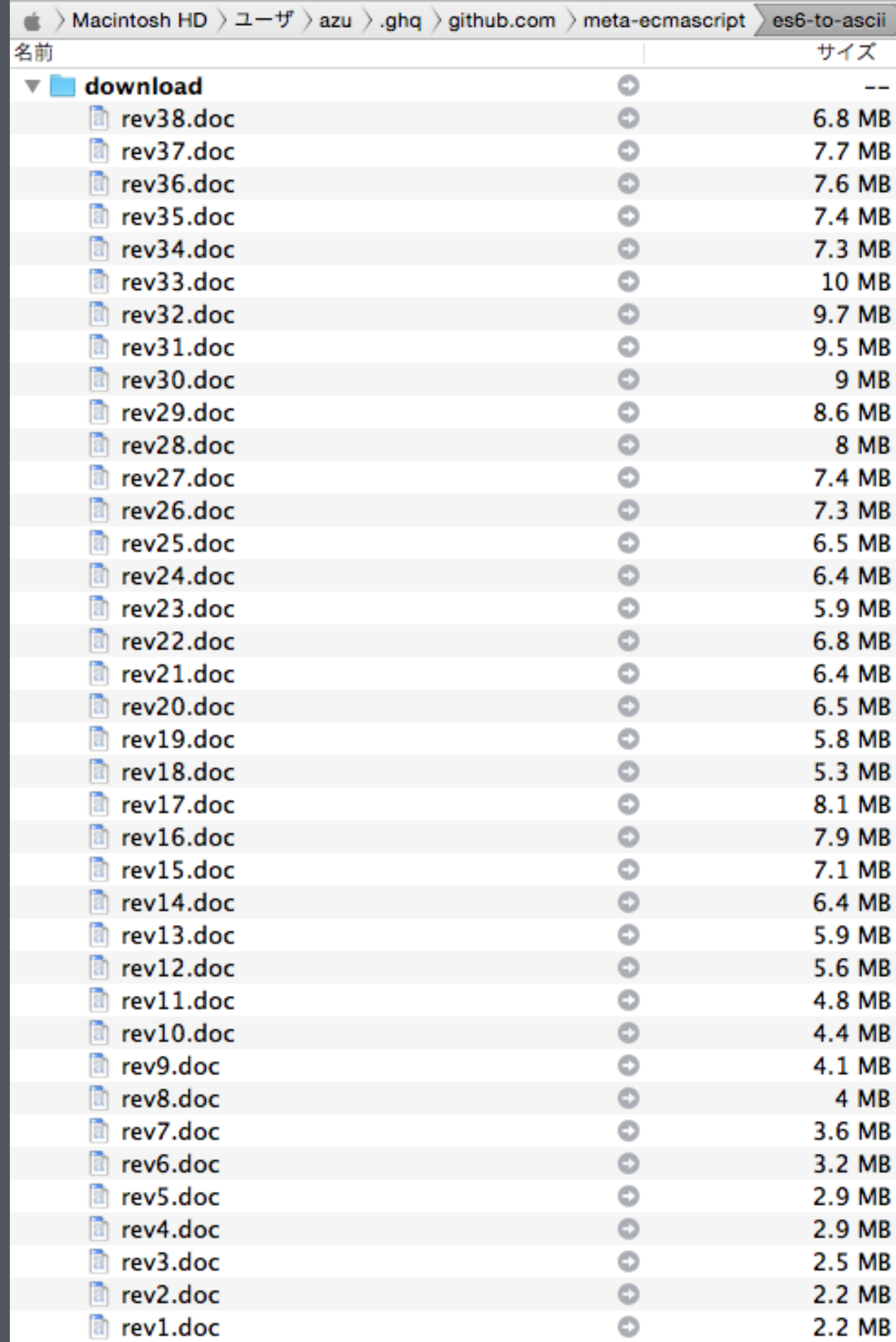
@azu\_re

Web scratch, JSer.info



# ECMAScript 6 ドラフト

- Draft Specification for ES.next  
(Ecma-262 Edition 6)
- Revision 38までである
- doc、pdfで公開されている
- 非公式のHTML版も公開されてる
  - [jorendorff/es-spec-html](http://jorendorff/es-spec-html)



Macintosh HD > ユーザ > azu > .ghq > github.com > meta-ecmascript > es6-to-ascii

名前	サイズ
▼ download	--
rev38.doc	6.8 MB
rev37.doc	7.7 MB
rev36.doc	7.6 MB
rev35.doc	7.4 MB
rev34.doc	7.3 MB
rev33.doc	10 MB
rev32.doc	9.7 MB
rev31.doc	9.5 MB
rev30.doc	9 MB
rev29.doc	8.6 MB
rev28.doc	8 MB
rev27.doc	7.4 MB
rev26.doc	7.3 MB
rev25.doc	6.5 MB
rev24.doc	6.4 MB
rev23.doc	5.9 MB
rev22.doc	6.8 MB
rev21.doc	6.4 MB
rev20.doc	6.5 MB
rev19.doc	5.8 MB
rev18.doc	5.3 MB
rev17.doc	8.1 MB
rev16.doc	7.9 MB
rev15.doc	7.1 MB
rev14.doc	6.4 MB
rev13.doc	5.9 MB
rev12.doc	5.6 MB
rev11.doc	4.8 MB
rev10.doc	4.4 MB
rev9.doc	4.1 MB
rev8.doc	4 MB
rev7.doc	3.6 MB
rev6.doc	3.2 MB
rev5.doc	2.9 MB
rev4.doc	2.9 MB
rev3.doc	2.5 MB
rev2.doc	2.2 MB
rev1.doc	2.2 MB

# ECMAScript 6 ドラフトの問題点

- 一般的なバージョン管理がされていない
  - Wordの編集履歴とChangeLogのみ
- Revision同士の差分が取得できない
  - 例) Rev 16とRev 30を比較できない

# ES6ドラフトの差分を見たい

- Gitリポジトリとして更新されていけばなー
  - Rev毎に変更点がコミットされたものがあれば
- ないなら作ってしまおう !!
- [meta-ecmascript/es6-draft-revision](https://github.com/meta-ecmascript/es6-draft-revision)

# ということ?

- ES6ドラフトをRev毎にテキストファイルに変換
- 1つのテキストファイルの単一リポジトリとしてES6ドラフトの履歴を表現
- [meta-ecmascript/es6-draft-revision](https://github.com/meta-ecmascript/es6-draft-revision)

## Gitリポジトリ





09:38 azu o [master] <rev38> rev38

09:38 azu o <rev37> rev37

09:38 azu o <rev36> rev36

09:38 azu o <rev35> rev35

09:38 azu o <rev34> rev34

09:38 azu o <rev33> rev33

09:38 azu o <rev32> rev32

09:38 azu o <rev31> rev31

09:38 azu o <rev30> rev30

09:38 azu o <rev29> rev29

09:38 azu o <rev28> rev28

09:38 azu o <rev27> rev27

09:38 azu o <rev26> rev26

09:38 azu o <rev25> rev25

09:38 azu o <rev24> rev24

09:38 azu o <rev23> rev23

09:38 azu o <rev22> rev22

09:38 azu o <rev21> rev21

09:38 azu o <rev20> rev20

09:38 azu o <rev19> rev19

09:38 azu o <rev18> rev18

09:38 azu o <rev17> rev17

09:38 azu o <rev16> rev16

09:38 azu o <rev15> rev15

09:38 azu o <rev14> rev14

09:38 azu o <rev13> rev13

09:38 azu o <rev12> rev12

09:38 azu o <rev11> rev11

09:38 azu o <rev10> rev10

09:38 azu o <rev9> rev9

09:38 azu o <rev8> rev8

09:38 azu o <rev7> rev7

09:38 azu o <rev6> rev6

09:38 azu o <rev5> rev5

09:38 azu o <rev4> rev4

09:38 azu o <rev3> rev3

09:38 azu o <rev2> rev2

09:38 azu I <rev1> rev1

commit 5d27842717cd0604a2f561c269bcad369fdc646c

Refs: <rev34>

Author: azu <azuciao@gmail.com>

AuthorDate: Thu May 28 09:38:24 2015 +0900

Commit: azu <azuciao@gmail.com>

CommitDate: Thu May 28 09:38:24 2015 +0900

rev34

---

es6-draft.txt | 11054 +-----

1 file changed, 4701 insertions(+), 6353 deletions(-)

diff --git a/es6-draft.txt b/es6-draft.txt

index 24b7a18..88a7b07 100644

--- a/es6-draft.txt

+++ b/es6-draft.txt

@@ -85,20 +85,19 @@ \_7.1.3\_ \_ToNumber ( argument )\_

\_7.1.4\_ \_ToInteger ( argument )\_

-\_7.1.5\_ \_ToInt32 ( argument ) - Signed 32 Bit Integer\_

+\_7.1.5\_ \_ToInt32 ( argument )\_

-\_7.1.6\_ \_ToUint32 ( argument ) - Unsigned 32 Bit Integer\_

+\_7.1.6\_ \_ToUint32 ( argument )\_

-\_7.1.7\_ \_ToInt16 ( argument ) - Signed 16 Bit Integer\_

+\_7.1.7\_ \_ToInt16 ( argument )\_

-\_7.1.8\_ \_ToUint16 ( argument ) - Unsigned 16 Bit Integer\_

+\_7.1.8\_ \_ToUint16 ( argument )\_

-\_7.1.9\_ \_ToInt8 ( argument ) - Signed 8 Bit Integer\_

+\_7.1.9\_ \_ToInt8 ( argument )\_

-\_7.1.10\_ \_ToUint8 ( argument ) - Unsigned 8 Bit Integer\_

+\_7.1.10\_ \_ToUint8 ( argument )\_

# 作り方

1. ドラフトのdocファイルを全部ダウンロード
2. docファイルをテキストに変換する
3. GitにRevごとにコミットとして突っ込む
4. `git log -S -i "search"`とかで検索できる!



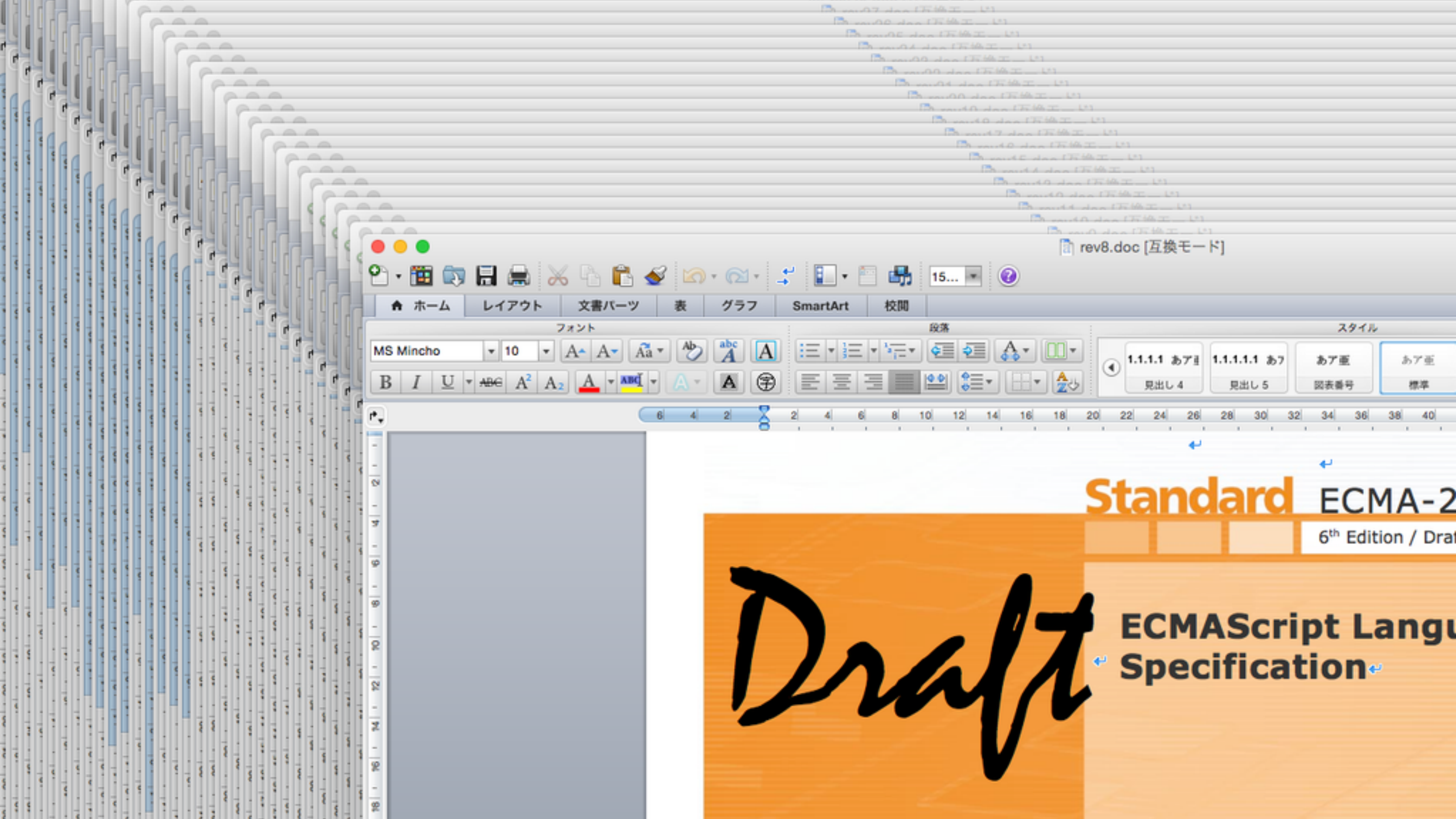
# ドラフトのdocファイルを全部ダウンロード

- ダウンロードするURLの一覧
- [meta-ecmascript/es6-spec-changelog](#)
- ダウンロードするツール
- [meta-ecmascript/download-es6-spec](#)

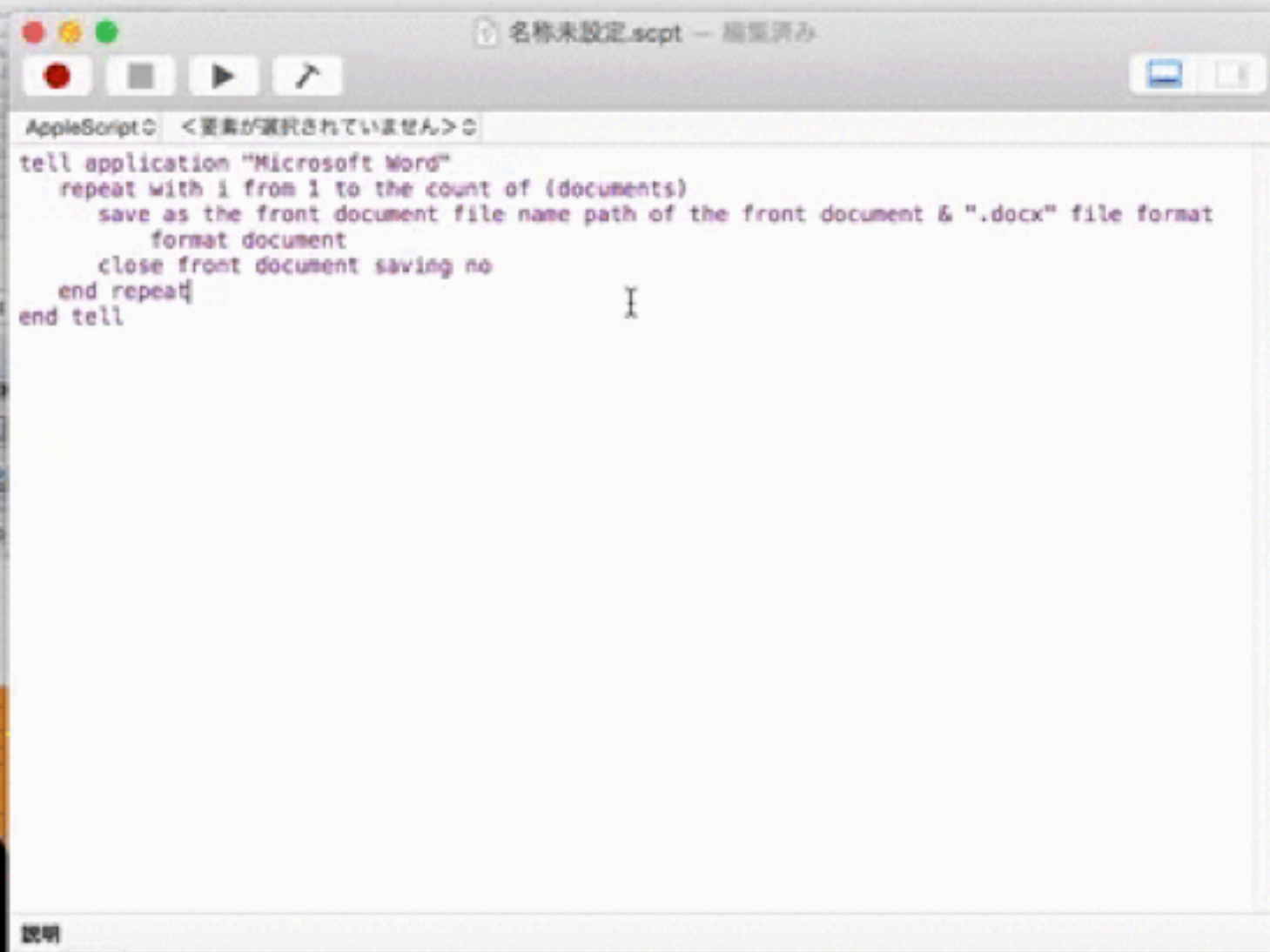
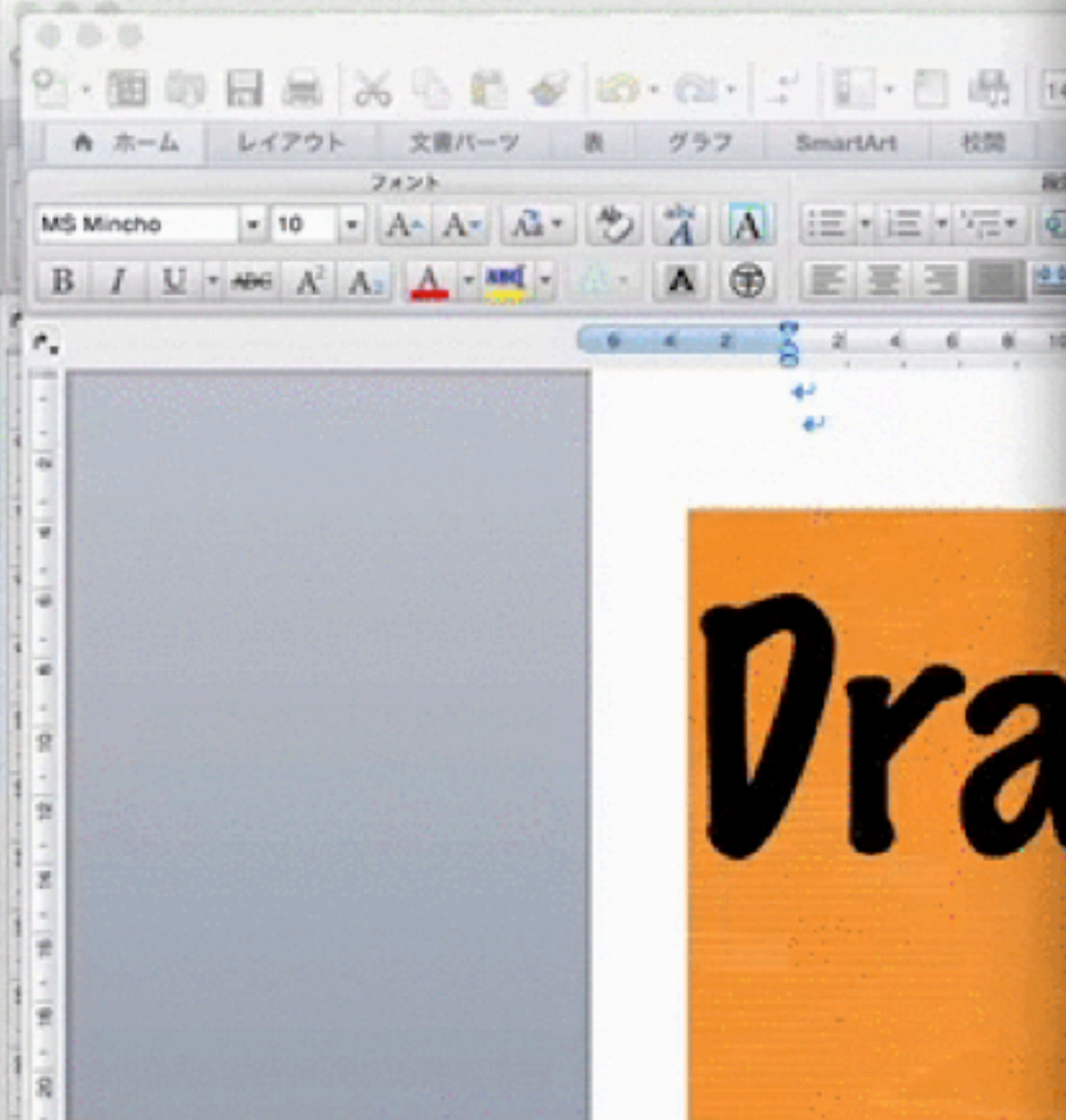
名前		サイズ
▼ <b>download</b>	→	--
rev38.doc	→	6.8 MB
rev37.doc	→	7.7 MB
rev36.doc	→	7.6 MB
rev35.doc	→	7.4 MB
rev34.doc	→	7.3 MB
rev33.doc	→	10 MB
rev32.doc	→	9.7 MB
rev31.doc	→	9.5 MB
rev30.doc	→	9 MB
rev29.doc	→	8.6 MB
rev28.doc	→	8 MB
rev27.doc	→	7.4 MB
rev26.doc	→	7.3 MB
rev25.doc	→	6.5 MB
rev24.doc	→	6.4 MB
rev23.doc	→	5.9 MB
rev22.doc	→	6.8 MB
rev21.doc	→	6.4 MB
rev20.doc	→	6.5 MB
rev19.doc	→	5.8 MB
rev18.doc	→	5.3 MB
rev17.doc	→	8.1 MB
rev16.doc	→	7.9 MB
rev15.doc	→	7.1 MB
rev14.doc	→	6.4 MB
rev13.doc	→	5.9 MB
rev12.doc	→	5.6 MB
rev11.doc	→	4.8 MB
rev10.doc	→	4.4 MB
rev9.doc	→	4.1 MB
rev8.doc	→	4 MB
rev7.doc	→	3.6 MB
rev6.doc	→	3.2 MB
rev5.doc	→	2.9 MB
rev4.doc	→	2.9 MB
rev3.doc	→	2.5 MB
rev2.doc	→	2.2 MB
rev1.doc	→	2.2 MB

# docファイルをテキストに変換する

- docファイルを直接テキストに変換は難しい
- doc -> docx -> txt としてやる
- doc から docx への変換を安定してやるにはWordを使うしかない...
- wordで開いてdocxで保存するマクロで変換
















































# GitにRevごとにコミットとする

- pandocでdocx -> txtとして変換
- revごとのテキストファイル作成
- es6-draft.txtにリネーム->コミット->次のRevでes6-draft.txtを上書き...
- 繰り返してGitのコミット積み上げる



グラフ	説明	
	 rev38	 master rev38
	 rev37	rev37
	 rev36	rev36
	 rev35	rev35
	 rev34	rev34
	 rev33	rev33
	 rev32	rev32
	 rev31	rev31
	 rev30	rev30
	 rev29	rev29
	 rev28	rev28
	 rev27	rev27
	 rev26	rev26
	 rev25	rev25
	 rev24	rev24
	 rev23	rev23
	 rev22	rev22
	 rev21	rev21
	 rev20	rev20
	 rev19	rev19
	rev18	rev18
	rev17	rev17

完成

meta-ecmascript/es6-draft-revision



the `NewPromiseCapability` abstract operation.

`PromiseCapability` Records have the fields listed in Table

Table 57 – `PromiseCapability` Record Fields

_____	_____	_____
_FIELD NAME_	VALUE	MEANING
[[Promise]]	An object	An object that is u
[[Resolve]]	A function object	The function that i
[[Reject]]	A function object	The function that i
_____	_____	_____

`IfAbruptRejectPromise ( value, capability )`

`IfAbruptRejectPromise` is a short hand for a sequence of  
that use a `PromiseCapability` record. An algorithm step o

1. `IfAbruptRejectPromise(value, capability)`.

# (Gitリポジトリにすると)何が良いのか

- テキストファイルだから検索しやすい
- Gitを使った変更履歴の検索パターンが応用できる
  - われわれは、いかにして変更点を追うか
- ドラフトのどのバージョンで変更があったのかが追しやすい
  - 関連するログを調べて何故そういう変更があったのかを調べるのに役立つ

# DEMO

- `Reflect.construct`が何時入ったのかを探してみよう
- 26.1.2 `Reflect.construct ( target, argumentsList [, newTarget] )`
- `new target` + 内部処理もできるReflect API

# Class - TypeScriptとBabel

- Classで定義したmethodの違い
- TypeScriptはmethodがenumerable
- Babelはmethodがnon-enumerable
- どっちが正しい?



## TypeScript

Walkthrough: Classes

Share

Run

## JavaScript

```
1 class Greeter {  
2   greeting: string;  
3  
4   greet() {  
5     return "Hello, " + this.greeting;  
6   }  
7 }
```

```
1 var Greeter = (function () {  
2   function Greeter() {  
3   }  
4   Greeter.prototype.greet = function () {  
5     return "Hello, " + this.greeting;  
6   };  
7   return Greeter;  
8 })();  
9
```

```
1 class Greeter {  
2   greeting: string;  
3  
4   greet() {  
5     return "Hello, " + this.greeting;  
6   }  
7 }
```

```
1 "use strict";  
2  
3 var _createClass = (function () { function defineProperties(target, props) { for (var i  
4  
5   function _classCallCheck(instance, Constructor) { if (!(instance instanceof Constructor)  
6  
7   var Greeter = (function () {  
8     function Greeter() {  
9       _classCallCheck(this, Greeter);  
10    }  
11  
12    _createClass(Greeter, [{  
13      key: "greet",  
14      value: function greet() {  
15        return "Hello, " + this.greeting;  
16      }  
17    }]);  
18  
19    return Greeter;  
20  })();
```

# ClassMethodはnon-enumerable

- 現在の仕様ではClassのmethodはnon-enumerable
- Babelの方が仕様に沿ってる
- なぜこうした違いが生まれているのかを変更履歴から見る
  - DEMO:(やり方はこちら)



**Allen Wirfs-Brock**

@awbjs



Following

OK, methods defined within an ES6 class definitions are now specified as being non-enumerable. Object literal methods are enumerable.



FAVORITES

4



3:50 AM - 1 Feb 2015



# コミットを見つけたら

- Rev前後のミーティングノートやMLを探してみる
- [ES Discuss](#)
- [rwaldron/tc39-notes](#)
- [Bugzilla](#)

# 参考

- われわれは、いかにして変更点を追うか
- インデントコミットで真犯人がわからなくなった場合の git blame - Qiita
- git logでコミットの差分の中身で絞り込む - Qiita

# 類似研究

- [dspinellis/unix-history-repo](#)
  - Unixの歴史のgitレポジトリ

# まとめ

- (Gitの)歴史は捏造できる
- Git向けのツールがそのまま転用できる
  - [ECMAScript 6ドラフトのDiff検索用リポジトリを作った | Web Scratch](#)
  - [meta-ecmascript/es6-draft-revision](#)