

Q

A

JavaScript情報って
なんだっけ?

JSer.info 5周年記念イベント

アジェンダ

- 情報の定義
- JavaScript情報は欲しい、でも溢れてる
- 選択肢が多すぎる問題を切り分けする話
- ライブラリコミュニティの読み方

最初に書いておくが答えはない



「情報」って何?

「情報」って何?

『知る』ということの実体化。われわれが、あるものについて『知る』ということは、何かしらを得たこと、何かを頭の中に取り込んだことである。その『何かしら』を、われわれは情報と呼ぶのである

— 高橋秀俊

データ != 情報

- データ 자체はデータでしかない
 - ただのバイト列のデータ
 - (人間が)データを解釈して情報となる
- 意味あるデータ = 情報
- 同じデータでも人によってそれが**情報**なのか?は異なる



なんで情報が欲
いの?

何で情報が欲しいの?

- 新しい技術が面白そうだから
- 開発で楽をしたいから
- 話題だから
- 選択肢を持ちたいから 



情報って既に沢山
あるのでは?



ありすぎて困る

情報の整理が必要

情報が整理されないと何が問題なっ

てくるのか

少し過去を振り返る

2011年

世界のJavaScript情報を読もう

- 古いものと最近のものが混ざって見つけにくい

Be the first to clip this slide

現在を知るためには...

- 現実的な問題
 - JavaScript情報が溢れすぎ
 - 古い情報が混じっている
 - jQuery語がじわじわと
- ごちゃごちゃしてる中から見つける必要性

- 情報の取捨選択が必要になる



2012年

世界のJavaScriptを読もう @ 2012

- 古いものと最近のものが混ざってる
- 最近のものを整理するWeekly、ニュースサイトが増えた

乱立するまとめサイト

- ブラウザのアップデート頻度が増加
- Webの変化が急速化
- Webの情報量の増加
- SBMがまとめ役を保てなくなった
 - まとめ役が求められた
 - => Weekly系、小さくまとめるソーシャルニュースサイトが乱立

QUENCY OF UPDATE	Website	Updated	Keywords	Twitter
Daily	JavaScript Weekly	weekly	JavaScript Library Tools	@JavaScriptDaily
Weekly	Dart Weekly	weekly	Dart JavaScript	
Monthly	HTML5 Weekly	weekly	HTML5	
Other	Node Weekly	weekly	Node JavaScript	
GITHUB	Web Tools Weekly	weekly	HTML CSS JavaScript Tools	@WebToolsWeekly
new on Github	Open Web Platform Daily Digest	daily	Web Spec HTML CSS JavaScript	@simevidas
on Github	JS Central	other	Twitter JavaScript	@jscentral
	The Morning Brew	daily	NET JavaScript	
	JSter	monthly	JavaScript Library	
	Web Design Weekly	weekly	Design CSS	@wdweekly
	angular-js.in	daily	AngularJS JavaScript	@ng_directives
	ng-newsletter	weekly	AngularJS JavaScript	@ngnewsletter
	The WebKit Blog	weekly	Webkit Safari Browser	
	JSer.info	weekly	HTML CSS JavaScript Browser Tools	@jser_info
	JavaScript Jabber	weekly	JavaScript Podcast	@JSJabber
	CSS WG Blog	weekly	CSS Spec	
	W3C News	other	Spec	@W3C
	Echo JS	daily	JavaScript HTML5 Library Tools	@echajs

Meta Weekly



古いものと最近のものが混ざってる

- 自然の摂理
- しょうがない

2014年

世界のJavaScriptを読もう @ 2014

- 急速にライブラリの数が増えてきた
- ユーザーがライブラリの選択で悩むようになってきた
- 現在に対する選択肢の増加

ウェブの変化の早さ

- ウェブの変化に対応するためにブラウザのリリース速度は上がった
- ライブラリも変化/トレンドに対応するためリリース速度は上がっている

250000

- CPAN
- npm (node.js)
- Packagist (PHP)
- Rubygems.org

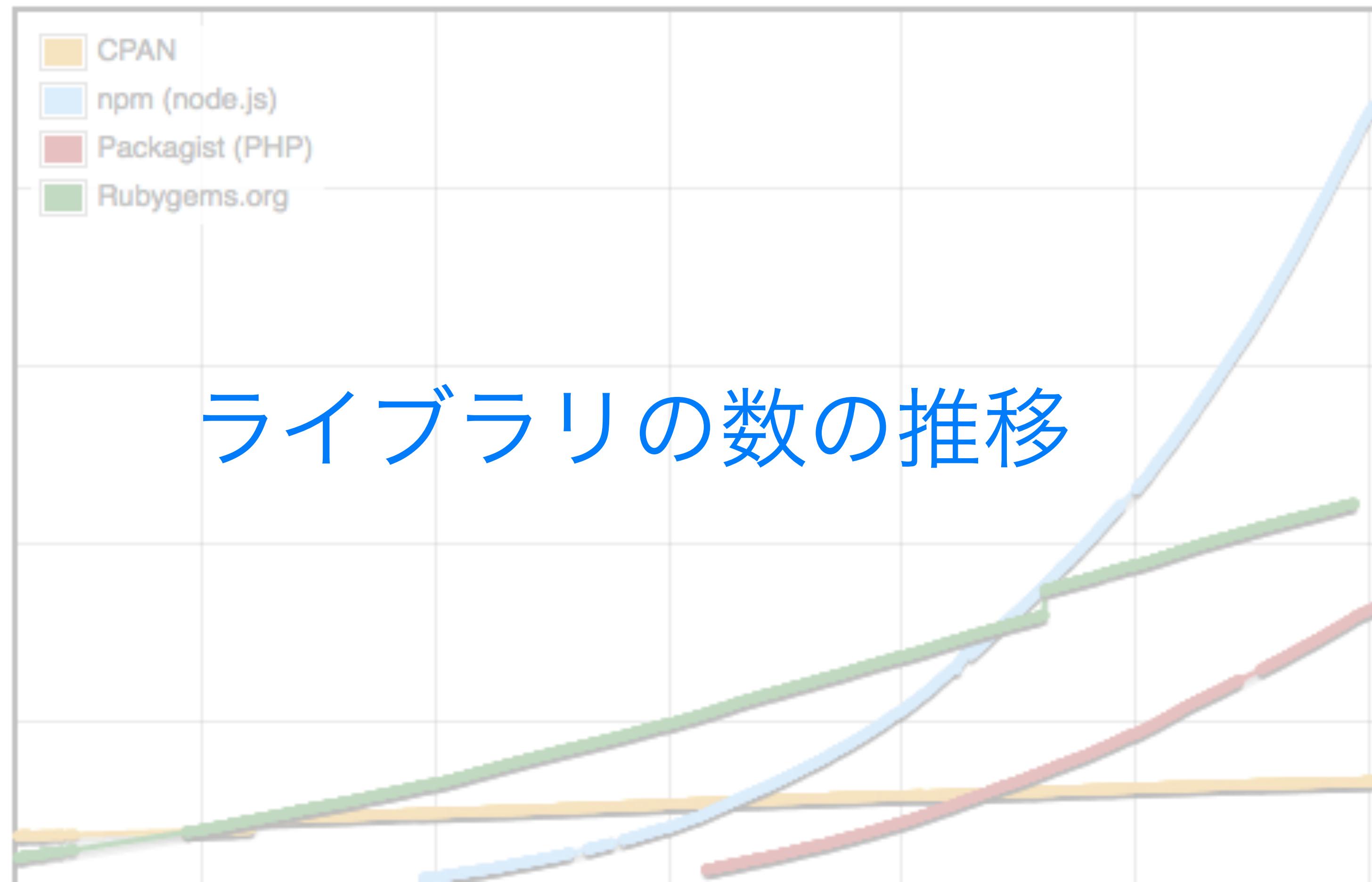
200000

150000

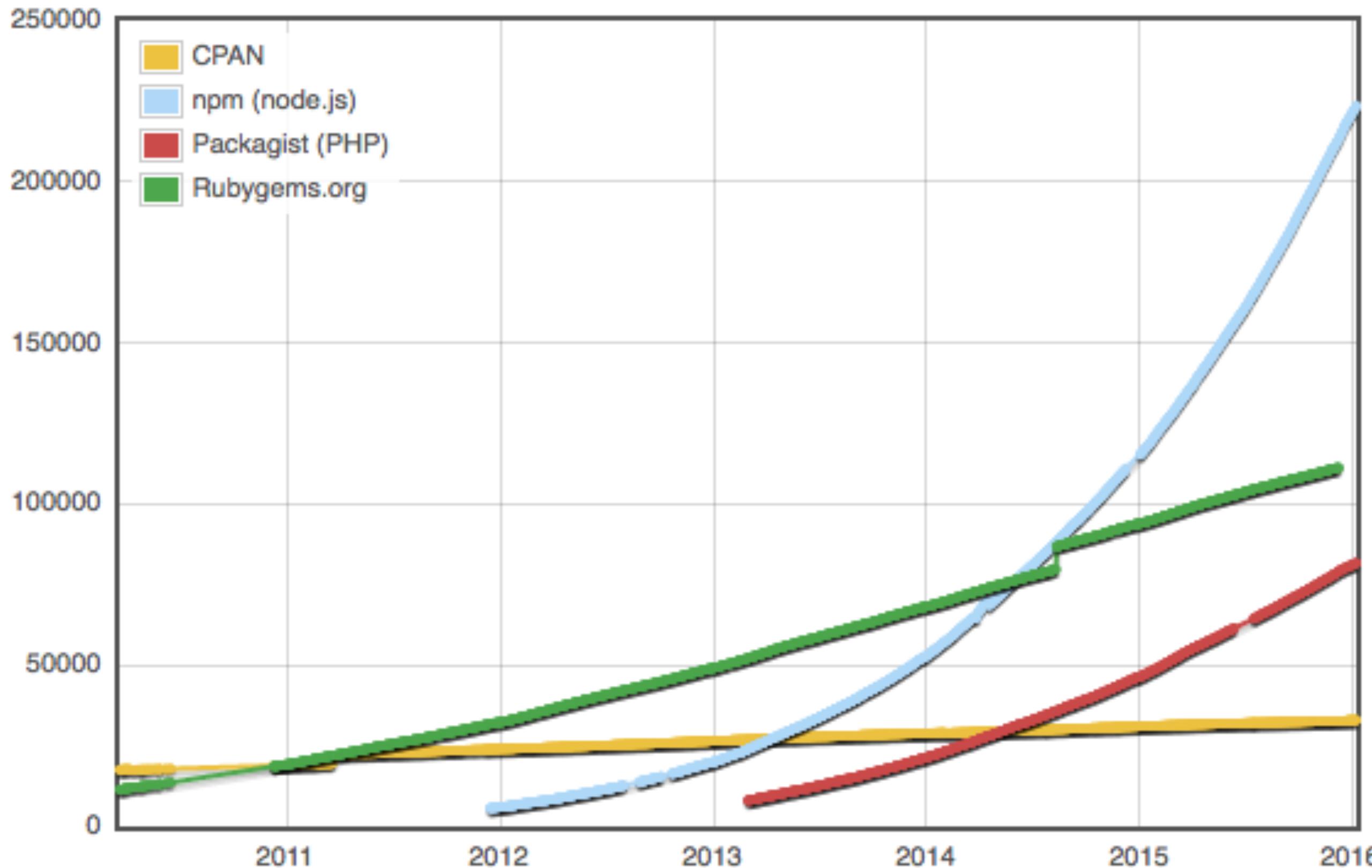
100000

50000

ライブラリの数の推移



Module Counts





そもそも何でこんなにライブラリって多いの?

色々な要因

- 単純にユーザーが多い
- 標準ライブラリがない
- Node.jsもコアAPIは少ない
- UIKitのようなUIデザインの元が薄い
- 小さなライブラリを組み合わせて使うのが良いとされてる
- etc...

Node.js

少ないほうが豊かであるという考え方

Less is more

@mikeal -- [Less is more · Issue #18 · nodejs/NG](#)

ウェブの変化が激しいのは決められてないことが多い

決定版がないというのは、変化が早いということ

@yosuke_furukawa -- [The Node.js Future | CodeLunch.fm](#)

2015年 - 去年

- 古いものと最近のものがまざってる
- 現在に対する候補が多い
- 未来に関する悩みが増えた ★
- ブラウザの情報が整理された



未来に関する悩み
って?

未来に関する悩み?

- 不確実な新しすぎるものが増えた
- Transpilerによって未来の仕様が身近になった

例えば、将来

- [ECMAScript Proposal](#)の機能が全て使えるわけではない
- [cssnext](#)にある機能が全て使えるわけではない



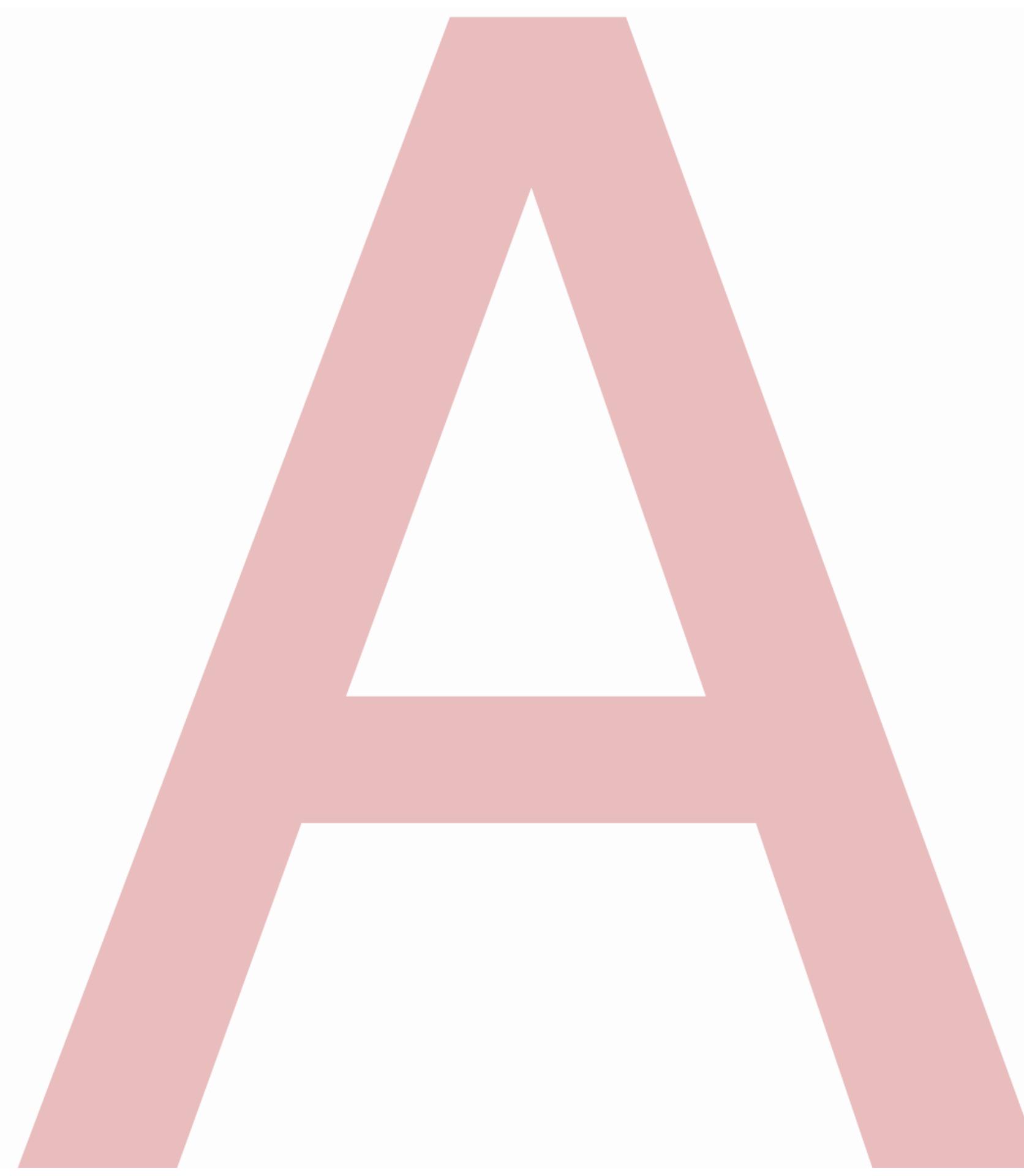
未来の悩みって今
悩まなくともいい
のでは?

未来と現在の混同

- BabelがES6の仕様だと思ってる人がいる
 - まだProposalの機能をES6だと思って使ってる
- AngularJSやReduxのように不確定な仕様を推奨するものがある
-  安定した仕様とそうでないものを見分ける必要がある



安定した/してない
仕様の見分け方は?



仕様の策定プロセ
スを知るのが近道

仕様 - ECMAScript

- 大きく分類
- **Stable** - ECMAScript 20XX
 - ES6、ES2016... 1年ごとリリース
- **Unstable** - Proposal
 - 仕様のProposal、ドラフト

仕様 - Proposal

- ES 2016からは機能毎のProposal
 - [ES.nextと策定プロセス | ECMAScriptとは何か？](#)
- Proposalには5段階のStageがある
 - Stage 4以外は基本的にUnstableと考える

仕様 - Stage

- Stage 0 Strawman - アイデア
- Stage 0.5  - Championが付きアイデア
- Stage 1 Proposal - 提案
- Stage 2 Draft - ドラフト
- Stage 3 Candidate- 仕様書と同じ形式
- Stage 4 Finished - 策定完了 => 仕様書へマージ

Stage 0.5 🚀

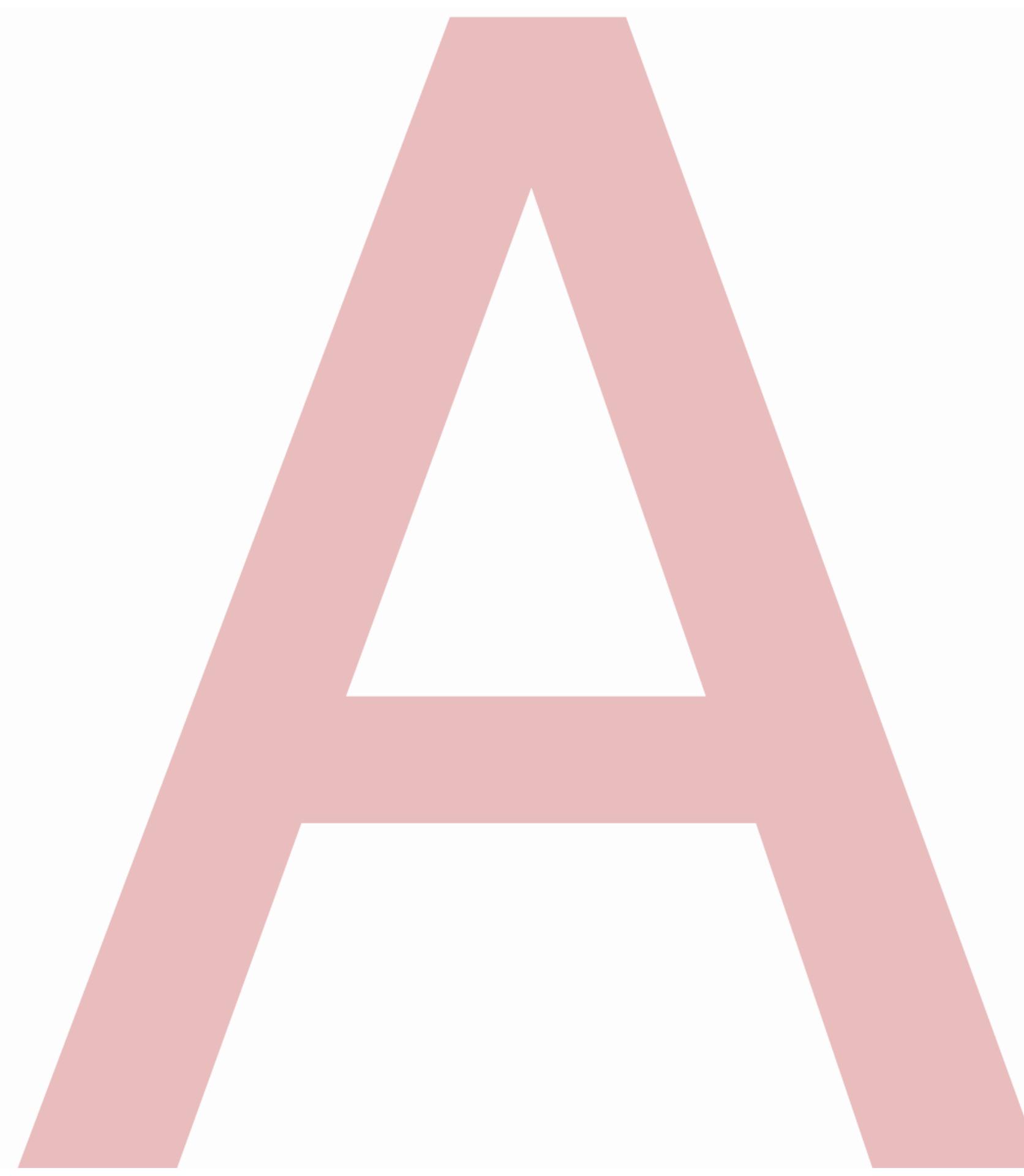
- (公式用語ではない)
- Stage 0 ProposalsにはChampionが付いてるかどうかの違いがある
- Champion 🚀 : 仕様を最後まで進めるTC39のメンバー
- 例) BabelはChampion 🚀についてないStage 0は実装しない
 - New plugin for pipeline operator proposal #3159

仕様の策定プロセスまとめ

- ECMAScriptはProposalを見れば安定したものなのか分かる
 - [tc39/ecma262](#)
- まだProposalなもの = 安定してない
- 正式に仕様に入ったもの = 安定してる
- 仕様はGitHub上で管理されている



仕様っていつブラ
ウザでいつ使える
の?



ステータスページ
を見ればわかる

ステータスページ

- [azu/browser-javascript-resource](#)

- 主要なブラウザ(エンジン)のステータスページが用意された!

- [status.modern.IE](#)

- [Firefox Platform Status](#)

- [Chromium Dashboard](#)

- [WebKit Web Platform Status](#)

Browser Platform Status Tracker

2016-01-08

Chromium

- **Changed:** Reflect (ES6)
Status: In development -> Behind a flag
ShippedAndroidMilestone: (n/a) -> 49
ShippedMilestone: (n/a) -> 49
ShippedOperaAndroidMilestone: (n/a) -> 36
ShippedOperaMilestone: (n/a) -> 36
- **Added:** addEventListenerOptions.passive
Status: In development

2015-01-07

Browser Platform Status Tracker

Chromium

- **Changed:** WebAssembly
Status: Proposed -> In development
- **Changed:** Deprecate some nonstandard uses of RTCPeerConnection legacy methods.
Status: Enabled by default -> Deprecated
ShippedMilestone: 23 -> 49
ShippedOperaAndroidMilestone: (n/a) -> 36
ShippedOperaMilestone: (n/a) -> 36
- **Changed:** Destructuring (ES6)
Status: In development -> Enabled by default
ShippedAndroidMilestone: (n/a) -> 49
ShippedMilestone: (n/a) -> 49
ShippedOperaAndroidMilestone: (n/a) -> 36

話を戻して  **BACK**



(ライブラリ)選択肢
が多いことはいい
ことなのでは?

基本はいいこと

- 選択肢が多い != 選択肢を持つてる
- 状況に応じて選択できないと意味ない



ライブラリはどう
やって選択するの?



必要なのは整理さ
れた情報

壊れやすい

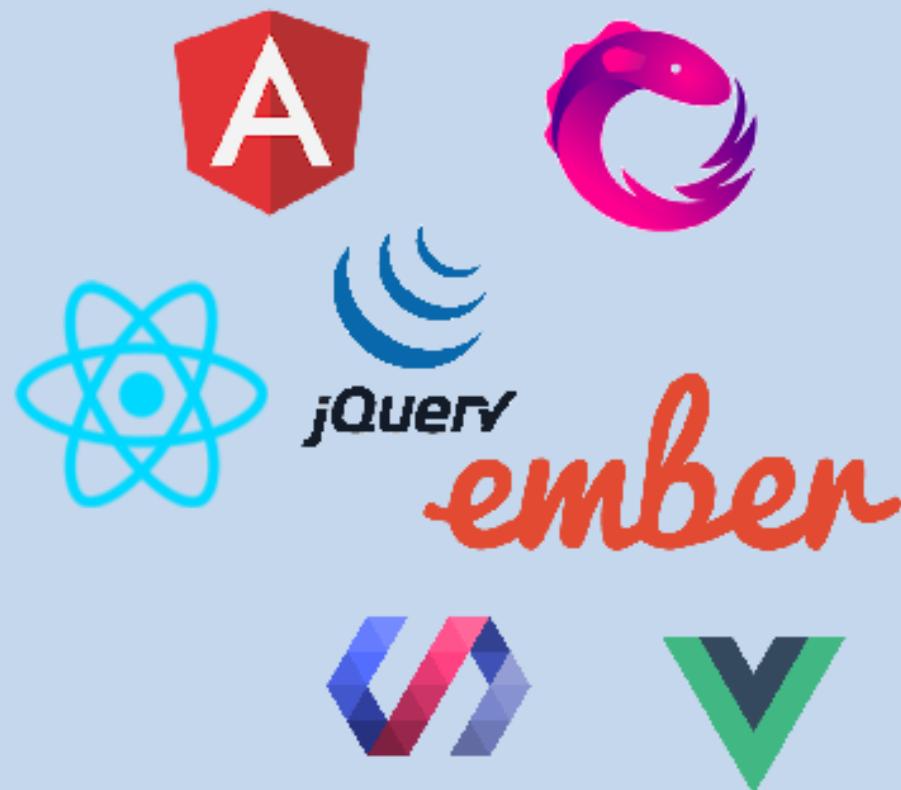
壊れにくい

壊れやすさ

ライブラリ

ブラウザ

仕様



ECMAScript

DOM

HTML

CSS

ライブラリ/ツールの選択は難しい

- ライブラリの種類による
 - JavaScriptが扱えることが増えた分色々なコミュニティがある
 - 例えばフロントエンド、音声、サーバ、自然言語系(NLP)
etc..
- 仕様のようにプロセスが一意ではない
- ライブラリごとのコミュニティがある



そのライブラリ安定してる?

ライブラリのバージョン種類

- ライブラリのバージョニングは基本Semver(例外あり)
- Suffixが付いてるのはUnstableなバージョン
 - β
 - α
 - RC
- 何もついてない(Stable)

ライブラリのリリースバージョン

ライブラリバージョンの安定度かけ算(嘘)

$$\alpha * 0.1 + \beta * 0.3 + RC * \text{Math.random()} + (\text{Version} - 1)$$

```
var α = 0;  
var β = 0;  
var RC = 1;  
var Version = 1;  
α * 0.1 + β * 0.3 + RC * Math.random() + (Version - 1)
```

ライブラリ/ツールの選択は難しい

- 自分なりの基準を設けて、複数の情報を組み合わせて使う
- 少なくとも死にそうなものは避けたい 
- 後は状況に応じて選べるようにしよう



ライブラリの生死
ってどうやって見
るの?



PULSE ❤

ライブラリのPlus

- ライブラリがアクティブにメンテナンスされているか?
- 小さなライブラリはAPIがフリーズしていることがあるので例外
 - 小さなライブラリとは自分で書き直せるなという規模
- 大きなライブラリ/フレームワークはアクティブなのかは重要
 - 特にエコシステムを持つものは重要 

[Code](#)[Pull requests 25](#)[Pulse](#)[Graphs](#)

December 8, 2015 – January 8, 2016

Period: 1 month ▾

Overview

97 Active Pull Requests

 76
Merged Pull Requests 21
Proposed Pull Requests

0 Active Issues

 0
Closed Issues 0
New Issues

GitHub Pulse

Excluding merges, 24 authors have pushed 123 commits to master and 130 commits to all branches. On master, 385 files have changed and there have been 5,011 additions and 707 deletions.

 13 Releases published by 2 people**Published** v6.3.16 a month ago**Published** v6.3.17 29 days ago**Published** v6.3.18 26 days ago**Published** v6.3.19 25 days ago

Welcome to Issue Stats!

Analyze and compare how long it takes for [Github](#) issues to be closed.

Check out some examples:

[rails/rails](#)

pull requests closed in about 5 hours

issues closed in 2 days

[nodejs/node](#)

pull requests closed in 1 day

issues closed in 2 days

[twbs/bootstrap](#)

pull requests closed in 1 day

issues closed in 3 days

Issue Stats

Github Repository	Time to Close Pull Requests ▾	Time to Close Issues	Closed Issues	Stars	Forks	Size	Select Language ▾
tenjava/website	less than a minute	about 1 hour	169	8	6	11829	PHP
ucsf-ckm/amalgamatic	less than a minute	about 6 hours	65	5	0	661	JavaScript
jared-schmidt/brown-bag	less than a minute	8 days	50	0	1	12438	JavaScript
tenjava/website	less than a minute	3 days	143	8	6	11829	PHP
Invoke-IR/Uproot	less than a minute	about 1 hour	34	39	4	689	PowerShell

Contributor(s) 

Contributor(s) △

- 現在がアクティブでもContributorが1人(本人のみ)というケース
- コミットが止まる => プロジェクトが止まる
- 単一障害点(SPOF)

单一障害点(SPOF)

- SPOFに対してどういう対策をしているかというのも一つの基準
- これに対する答えはアクティブなコントリビューターを増やす
- それのためにどういう工夫をしているかを見る

FILE

- LICENSE.md
- CONTRIBUTING.md
 - ガイドラインがあるか
- CODE_OF_CONDUCT.md
 - 行動規範、ポリティカル・コレクトネス問題
- ORGANIZATION
 - リリースを出来る人が複数いるかどうか

Add CODE_OF_CONDUCT by azu ·
Pull Request #83 · jser/jser.info

コミュニティの問題

- Babel
 - ~2015 in review – Medium
- C3.js
 - Create an organisation? • Issue #1440 • masayuki0812/c3
 - そのプロジェクトがContributeしやすくできるかが大事

プラグインアーキテクチャ

- ライブラリコアに全ての機能を持たせると、要望が全てIssueになる
 - Issueの処理ばかりになって開発に支障がでることがある
- 回避するためにプラグインアーキテクチャを持たせて分散
- [azu/JavaScript-Plugin-Architecture](#)

dear-github

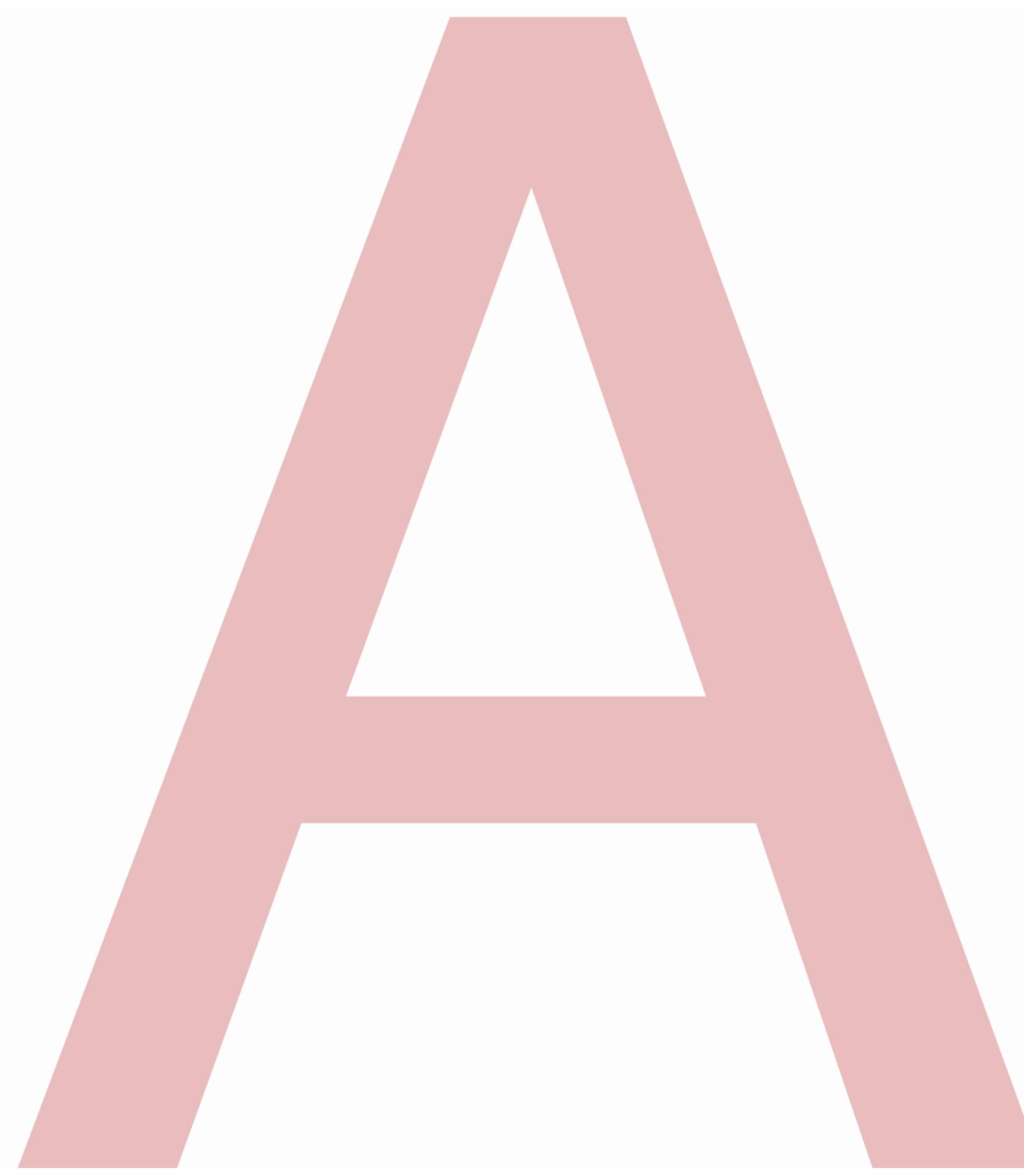
An open letter to GitHub from the maintainers of open source projects

The screenshot shows a GitHub repository page for 'dear-github/dear-github'. The repository has 69 commits, 1 branch, 0 releases, and 31 contributors. The README.md file contains the following text:

```
Dear GitHub,  
  
You have done so much to grow the open source community and make it really accessible to users. Somehow you have us chasing stars and filling up squares, improving the world's software in the process.  
  
However, many of us are frustrated. Those of us who run some of the most popular projects on GitHub feel completely ignored by you. We've gone through the only support channel that you have given us either to receive an empty response or even no response at all. We have no visibility into what has happened with our requests, or whether GitHub is working on them. Since our own work is usually done in the open and everyone has input into the process, it seems strange for us to be in the dark about one of our most important project dependencies.
```



じゃあ結局どうや
って選択すればい
いの?



一つの情報でダメ
なら複数の情報で



結局、詳しい人に
聞けばいいのでは?

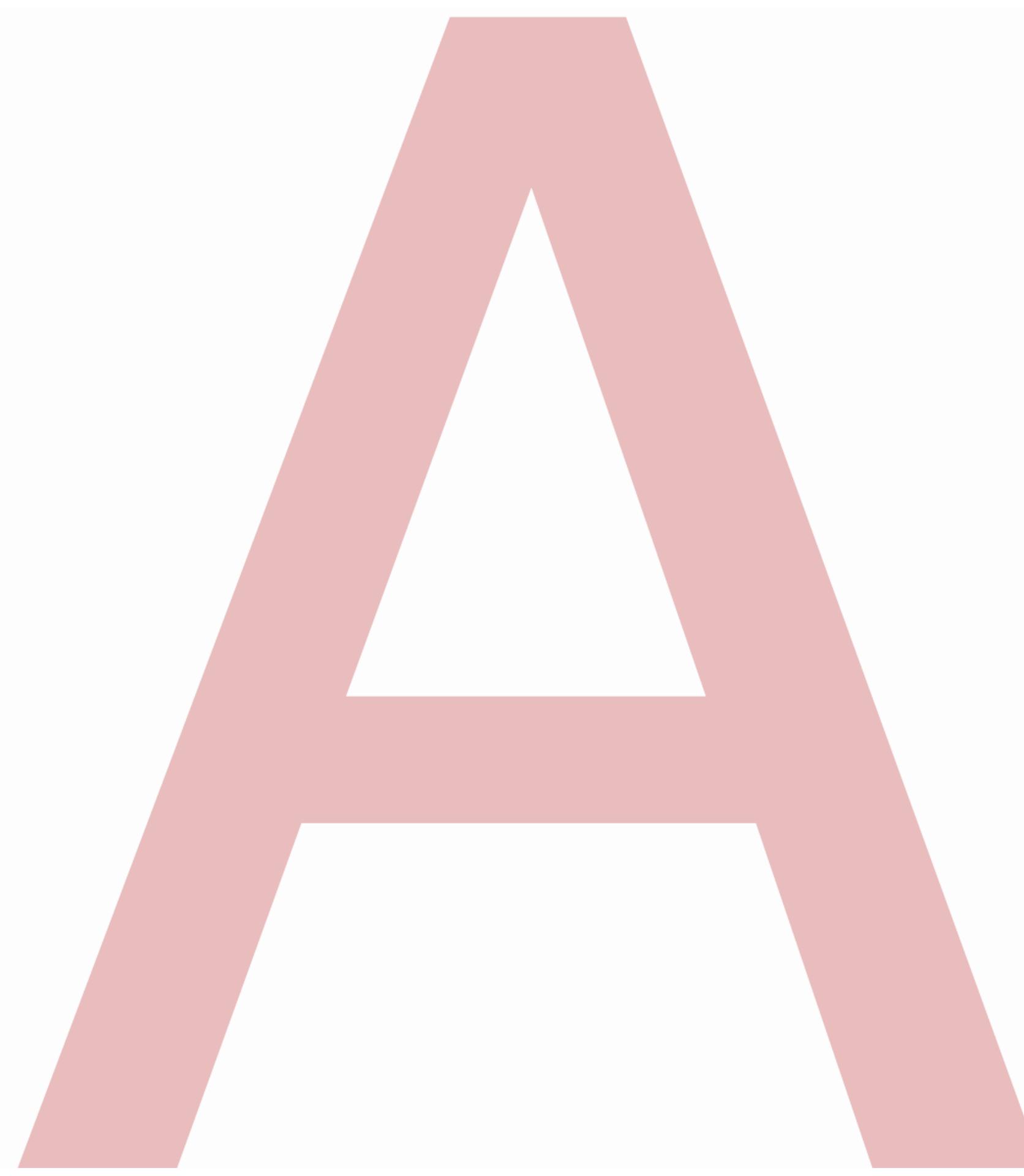


正解

誰が何に興味を持って
いるかはそういう意味
でも大事



選択できても1年後
には変わるんでし
ょ？



はい

一度選んで終わりでは
ない！

JavaScriptのエコシステムへようこそ！



まとめ

- データが多いと選択は難しくなる
- 情報にも種類があるのでちゃんと区別を付ける
 - StableとUnstableからはじめてみる
 - ライブラリの選択は難しい
 - コミュニティの問題も関連する
 - 一つの情報ではなく複数の情報から判断しよう

おわり