

Draft.jsによる

統制のとれたテキスト編集の世界

2017/05/27 HANATANI Takuma a.k.a. @potato4d

Biography

- 花谷拓磨 a.k.a. @potato4d
- 大阪出身→今年から東京生活
- 1998/04/13
- pixiv inc. エンジニア
- FRONTEND CONFERENCE 2017 実行委員長
- 📌のアイコンで社内ブログに載った
- 普段Vue.jsあたりの界隈にいます



Agenda

- ・ はじめに
- ・ Draft.jsとcontentEditable
- ・ 特長と選定するメリットについて
- ・ つらい話
- ・ まとめ

Agenda

- ・ はじめに
- ・ Draft.jsとcontentEditable
- ・ 特長と選定するメリットについて
- ・ つらい話
- ・ まとめ

はじめに

はなすこと・はなさないこと

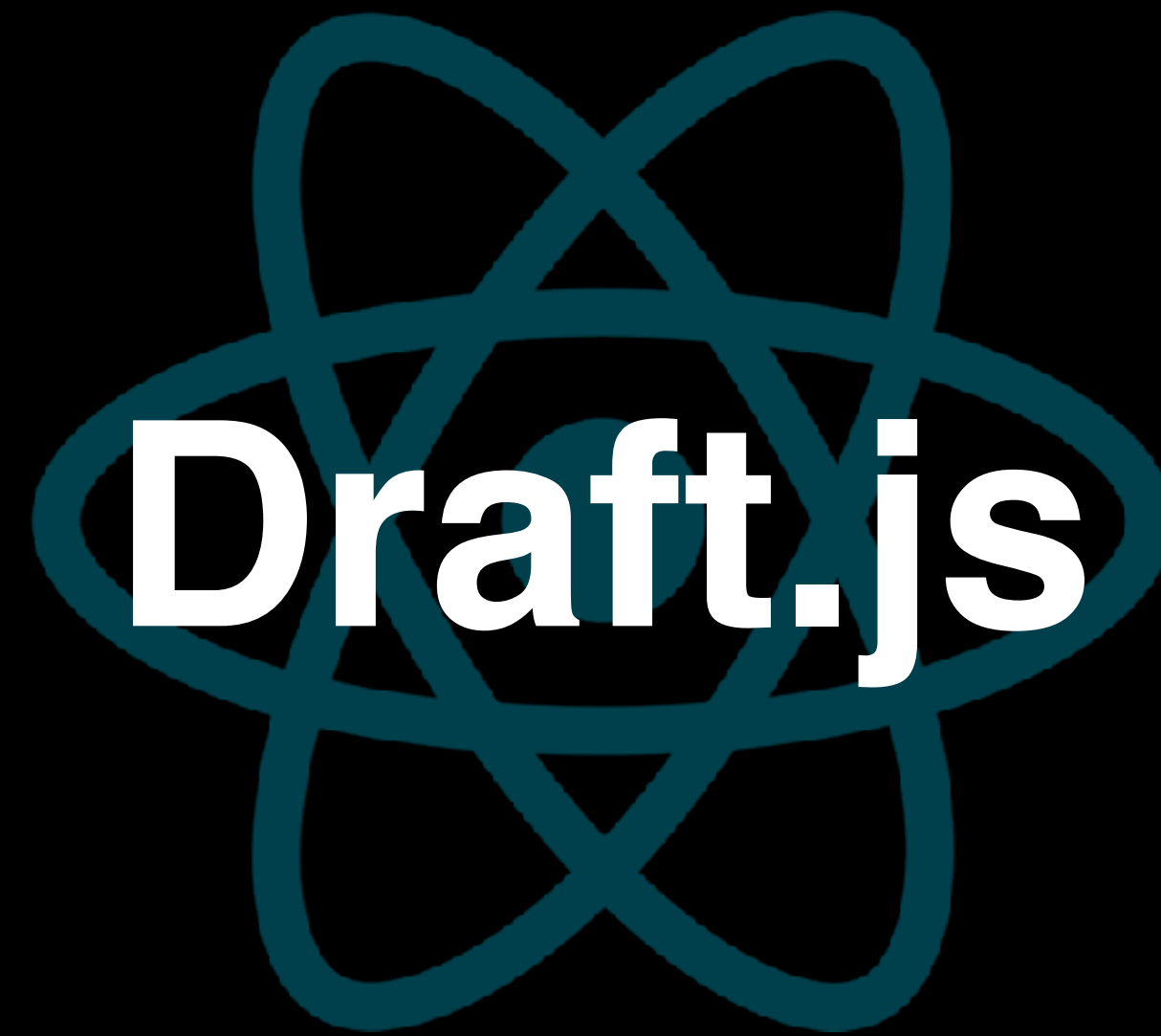
- ・ 時間も時間なのでDraft.jsすごい！という話だけします
- ・ あまり深い仕様の話や、実装ベースの話には触れません。
- ・ 明日の業務が良くなる知識にも触れません

こんな人におすすめ

- ・ 編集画面の要望に対して<textarea>で頑張ることに限界を感じた人
- ・ TinyMCEなどのエディタの状態管理や拡張性に不満がある人
- ・ テキストデータはMarkdown以外使わないという強い意志を持つ人
- ・ SPAにまともなテキストエディタを入れたい人
- ・ 「ちょっとMediumつくってよ」って言われて困ってる人

Agenda

- ・ はじめに
- ・ **Draft.jsとcontentEditable**
- ・ 特長と選定するメリットについて
- ・ つらい話
- ・ まとめ



Draft.js

Draft.js

- ・ いわゆる**WYSIWYG**を簡単に作るためのライブラリ
- ・ おおよそいつもどおり**Facebook**が開発している
- ・ 「**contentEditable**」を隠蔽し、統合管理できるようにしたもの
- ・ エディタの状態が**Reactコンポーネント**として提供される
- ・ プラグインによる拡張を前提としており、非常に**汎用的なI/F**を提供

Draft.js

The logo is a teal-colored atom symbol, consisting of a central circle and three intersecting elliptical orbits.

DOMベースのテキスト編集の抽象化ライブラリ

Draft.js

RICH TEXT EDITOR FRAMEWORK FOR REACT

Draft.js is a framework for building rich text editors in React, powered by an immutable model and abstracting over cross-browser differences.

Draft.js makes it easy to build any type of rich text input, whether you're just looking to support a few inline text styles or building a complex text editor for composing long-form articles.

In Draft.js, everything is customizable – we provide the building blocks so that you have full control over the user interface. Here's a simple example of a rich text editor built in Draft.js:

H1 H2 H3 H4 H5 H6 Blockquote UL OL Code Block
Bold Italic Underline Monospace

Tell a story...

[Learn more about Draft](#)

<https://draftjs.org/>



contentEditable



<h1>タグをh1と認識したまま書き換え可能

<p>もそう認識されたままに書き換えが可能

HTML

```
<div contenteditable="true">
  <h1>この夏流行る!モテカワコーデ</h1>
  <p>
    あのイーハトーヴォのすきとおった風、夏でも底に冷たさをもつ青いそら、うつくしい
    森で飾られたモリーオ市、郊外のぎらぎらひかる草の波。
  </p>
</div>
```

CSS

JS

Console Assets Keyboard

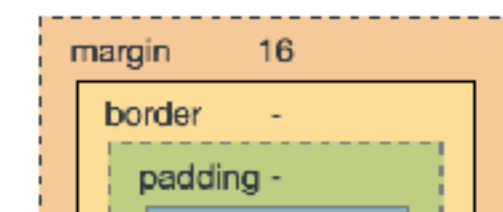
Elements Console Network Application Memory Sources Performance Security Audits

```
<!DOCTYPE html>
<html class>
  <head>...</head>
  <body>
    <div contenteditable="true">
      <h1><h1>タグをh1と認識したまま書き換え可能</h1>
      <p><p>もそう認識されたままに書き換えが可能 </p> == $0
    </div>
  </body>
</html>
</iframe>
<div class="assets-wrap" id="assets-wrap">...</div>
<div id="editor-drag-cover" class="drag-cover" style="display: none;"></div>
</div>
<div id="box-console" class="box box-console">...</div>
```

Styles Computed Event Listeners >>

Filter :hov .cls +

```
element.style {
}
p {
  user agent stylesheet
  display: block;
  -webkit-margin-before: 1em;
  -webkit-margin-after: 1em;
  -webkit-margin-start: 0px;
  -webkit-margin-end: 0px;
}
```



html body div div div div.output-sizer div#result_div.result iframe#iFrameKey-e92fd24b-cb61-99e4-4d31-5fd81331f8e9.result-iframe html body div p

contentEditable

- ・ HTML上のあらゆる要素を編集可能にするHTML Attribute
- ・ **実は太古の昔からひっそりと生きているもの(IE6にもある)**
- ・ 現在は **HTML 5の属性**として正式に仕様入りしている
- ・ HTML上で高度なテキスト編集をする際は必ず出てくる概念

Agenda

- ・ はじめに
- ・ Draft.jsとcontentEditable
- ・ **特長と選定するメリットについて**
- ・ つらい話
- ・ まとめ

Draft.jsでできること

- ・ **だいたいのテキスト編集**
 - ・ 選択範囲のテキストの文字装飾の変更(色/大きさ/書体)
 - ・ D&Dした画像を編集エリアにそのまま表示
 - ・ よくあるポップメニューや、いわゆる「みたままモード」の利用
 - ・ その他多くの機能（プラグイン依存）
- ・ **Reactによる統合的な状態の管理**
 - ・ その他多くの機能（プラグイン依存）

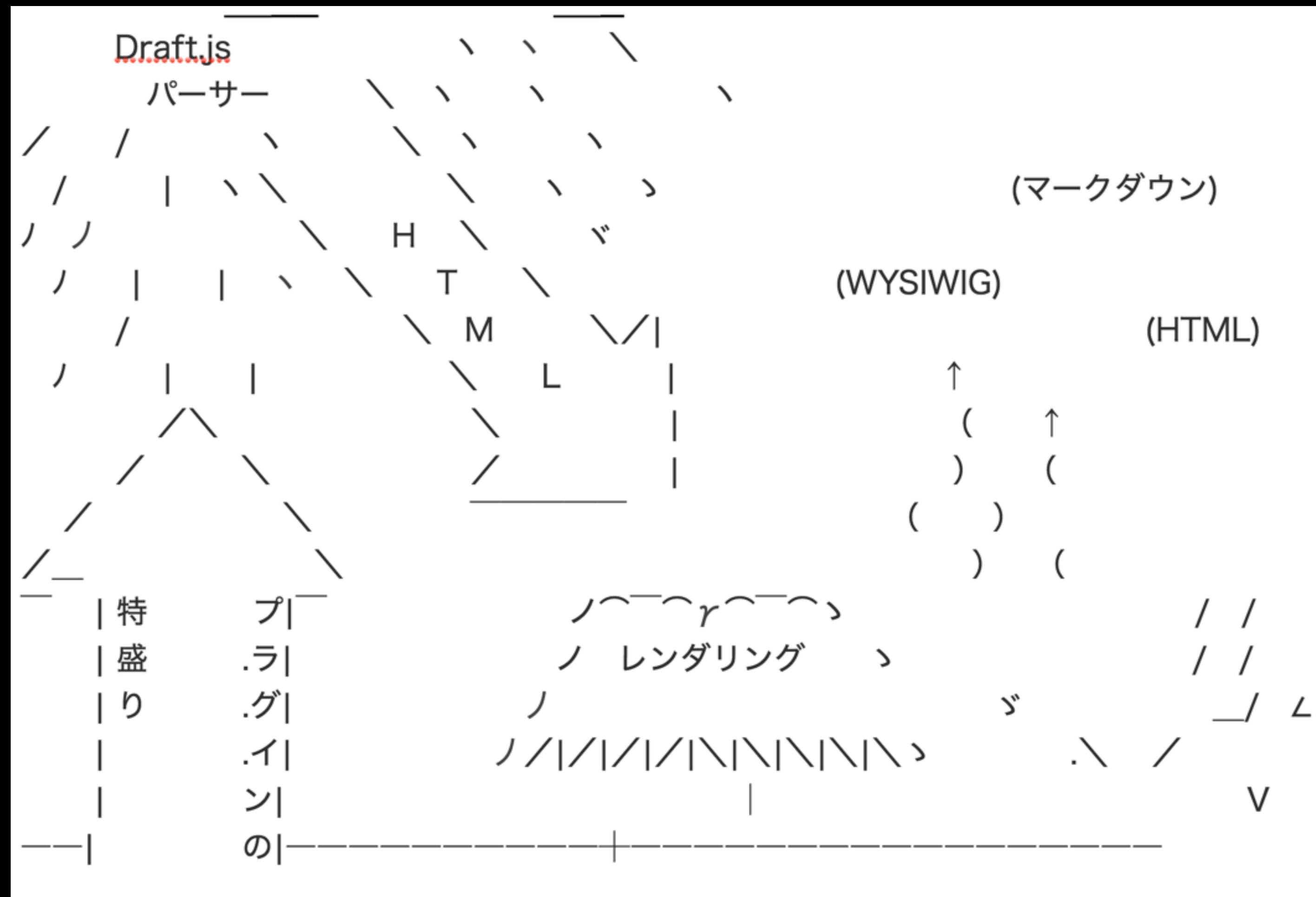
Demo

Draft.jsの利点

Draft.jsの利点

- ・ **全てのフォーマットを一度editorStateに変換する**
 - ・ Draft.jsは、ImageMagickのように、様々なテキストの形式を一度**Reactコンポーネント上のStateとして変換**、保持する仕組みを持っている
 - ・ これにより様々なフォーマットの相互変換が可能
 - ・ 極端な例で言うと、Markdownしか書きたくない人は**MD to HTML**の変換をしても良い(実質pandocみたいなこともできる)
 - ・ 書式以外にも、例えば🤔がUnicodeであるべきか:thinking:であるべきかは文脈に依存する
 - ・ そういった差異を気にせずに利用できるインターフェースとプラグインが充実している
 - ・ 「**どういう実装の都合か**」ではなく、「**どう書きたいか**」目線にたつことができる

Draft.jsの利点



Draft.jsの利点

- ・ **Reactコンポーネントとして管理されることによる**
 - ・ 全てはReactのコンポーネントとして管理されていく
 - ・ エディタ周りのライブラリ特有の、独自イベント&独自フックという仕様からは開放される
 - ・ Reactを経由してDOMへのドローは行われるため、比較的制御しやすい
 - ・ SPAなどに導入する際にReactであればどうやっても喧嘩することがない
 - ・ **現代的な開発をまともに行うための機能が揃っている以上の話ではない**

Agenda

- ・ はじめに
- ・ Draft.jsとcontentEditable
- ・ 特長と選定するメリットについて
- ・ **つらい話**
- ・ まとめ

つらい話

UNMET PEER DEPENDENCY

つらい話 1/2

- ・ 極端に安定しない

- ・ 久々に依存関係を入れ直すと壊れるのは日常茶飯事(5回以上やった気がする)
- ・ マイナーバージョンが一つ上がるだけで大体プラグイン同士で競合する、もしくは互換性が切れるので導入しているものを一つずつ追っていく必要がある
- ・ 追い打ちをかけるように発展途上ゆえのアップデの多さ(=ぶっ壊れやすさ)
- ・ 日曜プロジェクトだとやっているうちにつらくなってくる
- ・ 依存関係にチルダを使ってはいけない

つらい話 2/2

- ・ **良くも悪くもReact依存**
 - ・ **いつも通りFacebookが悪い**
 - ・ 個人的に好きなVue.jsなどと組み合わせるのは厳しいのでそちらを利用する場合はTinyMCEあたりでお茶を濁すしかない
 - ・ フロントエンドの勢力図がばらついているからこそなかなか流行らない
 - ・ 流行らないので海外のプラグインを気合で発掘して検証していく作業
 - ・ 気合でやってると依存関係がぶっ壊れてつらい
 - ・ **依存関係がぶっ壊れてつらい**

Agenda

- ・ はじめに
- ・ Draft.jsとcontentEditable
- ・ 特長と選定するメリットについて
- ・ つらい話
- ・ まとめ

まとめ

まとめ 1/2

- **contentEditable**

- HTML5系APIの一つ（とはいえ存在でいえずっとあった）
- HTML要素に対して直接的な介入・編集が可能なHTML attribute
- JavaScriptによる強制的な書き換えと違い、ある程度はStandardな仕様であるため、健全な編集が可能

- **Draft.js**

- 基本的にはcontentEditableを隠蔽し、ReactのI/Fをもたせた汎用ライブラリ
- 従来型のWYSIWYGライブラリと違い、他のFWなどと競合しづらいという利点がある

まとめ 2/2

- ・ **Draft.jsを採用する意義**

- ・ これまでの競合ライブラリと比較して、Reactコンポーネントであることが優位となっており、現代的なJavaScript開発と競合しない仕様となっている
- ・ ImageMagickよろしくな仕様で全てDraft.jsフォーマットに一度落とし込む都合上、様々な仕様や制約をまたいでのテキスト編集が可能
- ・ 正しくStandardな仕様を保つことにより、長く使われやすいライブラリとなっている

- ・ **つらさ**

- ・ 待てど暮らせど安定しないライブラリなので、ある程度苦しむ覚悟で使うこと
- ・ その覚悟さえあれば快適かつ堅牢な世界でのテキスト編集が可能

Draft.js使いましょう

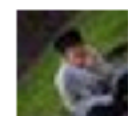
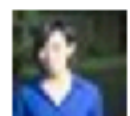
おわりに

[募集をみる](#)[フィードをみる](#)[プロフィール](#)[ツール](#)[Web エンジニア](#)

Mastodonのサーバー「Pawoo」で一緒に革命を起こすエンジニア募集

[ピクシブ株式会社](#)[埋め込む](#)[いいね！ 360](#)[ツイート](#)[Bookmark 62](#)[🕒 2017/04/17](#)

社員とあなたの共通の友達 (277)



+ 272

[👁 15496](#)[📣 応援する](#)[116](#)[ピクシブ株式会社のメンバー](#)[全員みる\(4\)](#)[応募資格がありません \(?\)](#)

現在47人がエントリーしています