

脱・無限カスタムフィールド

～ 動的CMSをやめて、GitHubをCMSとして使う話

# 自己紹介



榊原昌彦



一般社団法人リレーションデザイン  
研究所 代表理事



一般社団法人エリア・  
イノベーション・アライアンス CTO



Ionic Japan User Group  
代表



Ionic Developer Expert



Stripe Community Expert



女友だちをつくるための  
女性専用ソーシャルサービス



# 食品表示印刷



スマホとプリンタだけで食品表示ラベルをつくれる日本初の専用アプリ



多種類多数量のワインを複数拠点で管理できる日本唯一のアプリ。



CMS構築は罨がいっぱい

# ソフトウェアのメンテナンス

## WordPress

SERVER-APP

Last updated on 22 May 2023 🇯🇵



WordPress is a free and open-source content management system written in PHP.

Release	Released	Active Support	Security Support	Latest
6.2	2 months and 2 weeks ago (29 Mar 2023)	Yes	Yes	6.2.2 (20 May 2023)
6.1	7 months ago (02 Nov 2022)	Ended 2 months and 2 weeks ago (29 Mar 2023)	Yes	6.1.3 (20 May 2023)

<https://endoflife.date/>

## PHP

LANG

Last updated on 09 June 2023 🇯🇵



PHP: Hypertext Preprocessor (or simply PHP) is a general-purpose programming language originally designed for web development.

Release	Released	Active Support	Security Support	Latest
8.2	6 months ago (08 Dec 2022)	Ends in 1 year and 5 months (08 Dec 2024)	Ends in 2 years and 5 months (08 Dec 2025)	8.2.7 (08 Jun 2023)
8.1	1 year and 6 months ago (25 Nov 2021)	Ends in 5 months (25 Nov 2023)	Ends in 1 year and 5 months (25 Nov 2024)	8.1.20 (08 Jun 2023)
8.0	2 years and 6 months ago (26 Nov 2020)	Ended 6 months and 2 weeks ago (26 Nov 2022)	Ends in 6 months and 2 weeks (26 Nov 2023)	8.0.29 (08 Jun 2023)
7.4	3 years and 6 months ago (28 Nov 2019)	Ended 1 year and 6 months ago (28 Nov 2021)	Ended 6 months ago (28 Nov 2022)	7.4.33 (08 Nov 2022)

Show more unmaintained releases

## MySQL

DB

Last updated on 19 April 2023 🇯🇵

MySQL is an open source database developed by Oracle. With its proven performance, reliability and ease-of-use, MySQL has become the leading database choice for web-based applications, used by high profile web properties including Facebook, Twitter, YouTube, Yahoo! and many more.

Release	Released	Premier Support	Extended Support	Latest
8.0	5 years ago (08 Apr 2018)	Ends in 1 year and 10 months (30 Apr 2025)	Ends in 2 years and 10 months (30 Apr 2026)	8.0.33 (16 Mar 2023)
6.7	7 years ago (09 Oct 2015)	Ended 2 years and 7 months ago (31 Oct 2020)	Ends in 4 months and 2 weeks (31 Oct 2023)	5.7.42 (16 Mar 2023)
6.6	10 years ago (01 Feb 2013)	Ended 5 years ago (28 Feb 2018)	Ended 2 years and 3 months ago (28 Feb 2021)	5.6.51 (06 Jan 2021)
5.5	12 years ago (03 Dec 2010)	Ended 7 years ago (31 Dec 2016)	Ended 4 years and 8 months ago (31 Dec 2018)	5.5.89 (21 Dec 2018)



The screenshot shows a website header with a hamburger menu icon on the left, the LoliPop! logo (a red flame icon followed by 'LOLIPOP!' and 'by GMO AGENCY' below it) in the center, and a search icon on the right. Below the header is a breadcrumb trail: 'ロリポップ!レンタルサーバー > お知らせ > 新着情報 >'. The main content area features a blue link: '【注意喚起】All in One SEOプラグインで重大な脆弱性が確認されております'. Below this is a large heading 'お知らせ' and a sub-heading '『ロリポップ!』からのお知らせです。'. The main announcement is titled '2021/12/24 【注意喚起】All in One SEOプラグインで重大な脆弱性が確認されております'. The body text reads: 'いつもロリポップをご利用いただき、誠にありがとうございます。この度、WordPress のプラグイン「All In One SEO」について脆弱性が確認され修正版が公表されました。ロリポップでもWordPressを多くのお客様にご利用いただいているためご案内いたします。脆弱性の概要としては、サイト攻撃者によるサイトの乗っ取り、データベースに記録された情報の盗み出しなどの恐れがあるという内容です。'



# 構造が難しいものを更新可能に？

九州キャンパス	
募集	<p>【1次募集】（2月28日をもって1次募集を締め切りました）            受付期間：2022年12月9日（金）～2023年2月28日（火）            選考期間：2023年3月1日（水）～2023年3月15日（水）</p> <p>【2次募集】            受付期間：2023年3月1日（水）～2023年4月30日（日）            選考期間：2023年5月1日（月）～2023年5月9日（火）</p>
eラーニング	2023年6月初週に配信開始予定。以降、2023年12月まで毎週配信予定。録画配信のため、各自の都合の良い日時に受講可能。詳細は受講時に情報共有予定。
実践研修	各自の状況に応じて、受講開始前を含めできるだけ早稲実業の構築に向けた準備開始。
オンラインゼミ	初回の集中研修以降、実践研修のフォローアップを中心としたオンラインゼミを定期的に開催予定。
集中研修	6回の集中研修開催を予定。原則として、1回の集中研修は2023年度の開催日程、未定。（決定次第、本報を2024年4月以降分は、開講後に受講生と相談の上）

## eラーニングについて / カリキュラム（予定）

0	本課程ガイダンス	本課程の意図・ゴール、カリキュラム内容、学びの姿勢、学び方などについて解説
1	公民連携事業【概論】	公民連携事業とは地域に何をもたらす方法論であるか。公民連携人材に求められる志について。
2	都市経営【概論】（1）	都市経営という視点を持つことが、都市・地域が抱える課題をどのように捉えられるか。
3	都市経営【概論】（2）	都市経営課題の明確化、都市経営資源の整理、都市経営ビジョンの策定に必要なポイント
4	公民連携事業ケーススタディ（1）	先進的取組みである岩手県紫波町オガールプロジェクトのポイントとプロセス
5	公民連携事業ケーススタディ（2）	プロスクールOBOGが取り組む、公民連携による公営住宅・公園一体整備事業である大阪府大東市北条プロジェクトのポイントとプロセス
6	公民連携事業ケーススタディ（3）	公民連携事業のプロセスと未来を検討します

# バックアップのメンテナンス

DB吹き飛びました。  
最終バックアップって昨日ですよね？！

先月からバックアップとれてませんでした。  
ディスク容量がいっぱいだったようです。

どうしたらいいんですか？！？！

どうにもなりません・・・。

## アクセス集中した時の対応

Webサイトが見れないんだけどどうして？  
せっかくテレビに取り上げられたのに。

アクセスが集中してサーバが落ちたからですね。  
落ち着くまで待つしかできないです。

どうしたらいいんですか?!?!

がっつりサーバにお金いれて再発防止を！

# ユーザがレイアウトを壊す

どこを更新したらいい？

AとBのフィールドを更新してください。

Cのフィールドを更新したらこわれた！

Cのフィールドを更新したからです。

# ホスティング料金

	月額	オートスケーール	PHP/MySQL
Netlify	0円～	あり	不可
Firebase Hosting	0円～	あり	不可
S社	425円～	なし	可
A社	20999.70円～	なし	可

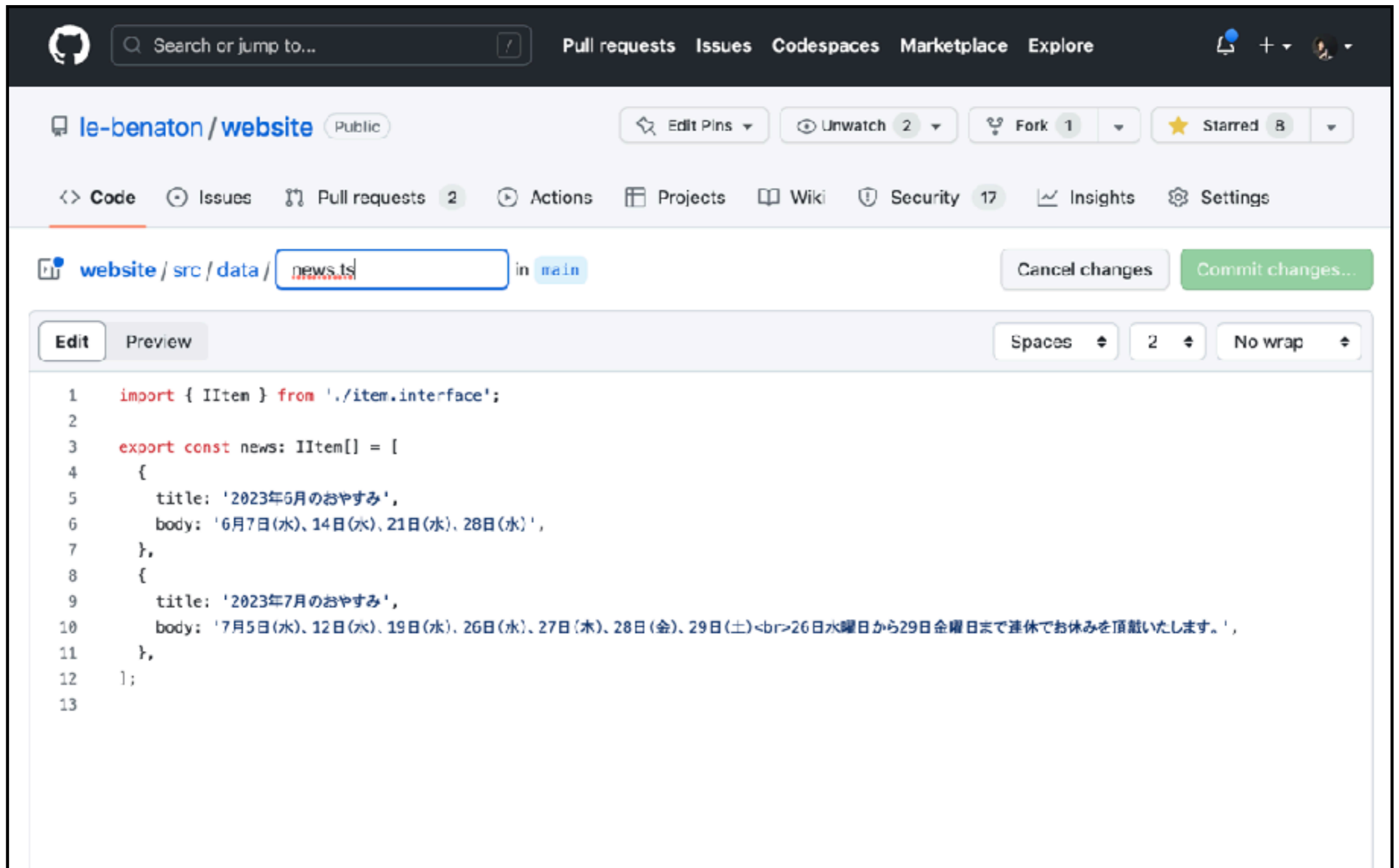
ユーザが更新する自社サイトでは、CMS構築。

**それ、必要ですか？**

なぜCMS構築が必要？

1. HTMLを書けないユーザが更新できるようにするため
2. 同一テンプレートで量産した記事を管理するため  
(例：ニュースサイト)
3. ~~ECサイト構築など複雑な手順が必要なものを手軽に構築するため~~

# これGitHubで解決できる！



The screenshot shows the GitHub web interface for editing a file. The repository is 'le-benaton / website' (Public). The file path is 'website / src / data / news.ts' in the 'main' branch. The editor is in 'Edit' mode. The code is as follows:

```
1  import { IItem } from './item.interface';
2
3  export const news: IItem[] = [
4    {
5      title: '2023年5月のおやすみ',
6      body: '6月7日(水)、14日(水)、21日(水)、28日(水)',
7    },
8    {
9      title: '2023年7月のおやすみ',
10     body: '7月5日(水)、12日(水)、19日(水)、26日(水)、27日(木)、28日(金)、29日(土)<br>26日水曜日から29日金曜日まで連休でお休みを頂戴いたします。',
11   },
12 ];
13
```

Buttons for 'Cancel changes' and 'Commit changes...' are visible. The editor also shows 'Spaces: 2' and 'No wrap' settings.



# 1. HTMLを書けないユーザが更新できるようにするため

簡単なペライチだったら、このレベルでいける。

website / src / data / news.ts in main

Edit

Preview

```
1  import { IItem } from './item.interface';
2
3  export const news: IItem[] = [
4    {
5      title: '2023年6月のおやすみ',
6      body: '6月7日(水)、14日(水)、21日(水)、28日(水)',
7    },
8    {
9      title: '2023年7月のおやすみ',
10     body: '7月5日(水)、12日(水)、19日(水)、26日(水)、27日(木)、28日(金)、29日(土)<br>26日水曜日から29日金曜日まで連休でお休みを頂戴いたします。',
11   },
12 ];
13
```

「これ見たら更新できたよ」

☰ README.md



## 西宮・夙川のフレンチレストラン「ル・ベナトン」

西宮・夙川のフレンチレストラン「ル・ベナトン」Webサイトのレポジトリです。以下URLで公開しています。

[benaton.net](https://benaton.net)

### 更新用リンク

項目	ファイル
お知らせ（画像あり）	<a href="https://github.com/le-benaton/website/edit/main/src/data/announcement.ts">https://github.com/le-benaton/website/edit/main/src/data/announcement.ts</a>
お知らせ	<a href="https://github.com/le-benaton/website/edit/main/src/data/news.ts">https://github.com/le-benaton/website/edit/main/src/data/news.ts</a>

### お知らせ（画像あり）に利用する画像をアップロードする方法

1. 画像を用意します。事前に、縦横300px程度にリサイズしてください。
2. 画像ファイル名に日本語が使われている場合、英数字のファイル名に変更します

# まちづくり会社のスタッフ

「キー自体を間違っって削除したら、失敗したってメールきた」

The screenshot displays a GitHub Actions CI run interface. At the top, the run is identified as 'Update course-sauna.ts #360' with a red 'x' icon indicating failure. The left sidebar shows navigation options: Summary (selected), Jobs, Run details, Usage, and Workflow file. The 'Jobs' section shows a failed 'build' job. The main area shows the job summary table and details for the 'ci.yml' workflow on the 'master' branch, triggered by a push 6 months ago. The job 'build' failed with a duration of 1m 34s and a billable time of 2m. The 'Annotations' section lists 1 error and 2 warnings. The error message states: 'Process completed with exit code 1.' The two warnings are: 'Node.js 12 actions are deprecated. For more information see: https://github.blog/changelog/2022-09-22-github-actions-all-actions-will-begin-running-on-no...' and 'The `save-state` command is deprecated and will be disabled soon. Please upgrade to using Environment Files. For more information see: https://github.blog/...'

Triggered via	Status	Total duration	Billable time	Artifacts
MasatakaMurase pushed → 3a91168 master	Failure	1m 34s	2m	—

**Annotations**  
1 error and 2 warnings

- build**  
Process completed with exit code 1.
- build**  
Node.js 12 actions are deprecated. For more information see: <https://github.blog/changelog/2022-09-22-github-actions-all-actions-will-begin-running-on-no-...>  
[Show more](#)
- build**  
The `save-state` command is deprecated and will be disabled soon. Please upgrade to using Environment Files. For more information see: <https://github.blog/...>  
[Show more](#)

# まちづくり会社のスタッフ

書き換えぐらいできるよ。



The screenshot shows a GitHub code viewer interface for the file `course-hokuriku.ts` in the repository `ppp-ps.net / src / writing / school`. The file is 363 lines long (352 loc) and 30.4 KB. The code is a JSON array of course objects. The visible code is as follows:

```
{
  label: 'e-ラーニング',
  line: [
    {
      period: '6月',
      content: `2023年6月初週に配信開始予定、以降、2023年12月頃まで毎週配信予定。録画配信のため、各自の都合の良い日時に受講可能。詳細は受講時に情報共有予定。`,
    },
  ],
},
{
  label: '実践研修',
  line: [
    {
      period: '11月',
      content: `各自の状況に応じて、受講開始前を含めできるだけ早い段階からの実践研修の開始を推奨。初年度の秋以降は次年度の公民連携事業の構築に向けた準備開始。`,
    },
  ],
},
{
  label: 'オンラインゼミ',
  line: [
    {
      period: '11月',
      content: `初回の集中研修以降、実践研修のフォローアップを中心的なテーマとしてグループごとにオンライン会議ツールを活用したオンラインゼミを定期的を開催予定。`,
    },
  ],
},
],
}
```

# 実装例 1 : 自作テンプレート

HTML

```
<ul class="list-news"></ul>
```

+

TypeScript

```
export interface IItem {
  selector: string;
  template: string;
  data: Array<{ title?: string; ja?: string; price?: string; body?: string;
  image?: string }>;
}

export const news: IItem = {
  selector: '<ul class="list-news"></ul>',
  template: '<li><label class="form-label">{{ title }}</label><p>{{ body }}</p></li>',
  data: [
    {
      title: 'コロナ感染症予防対策',
      body: 'コロナ感染症予防対策として、空気の入れ替え、間隔に余裕のある席へのご案内、アクリル板等、接客スタッフのマスク着用などの対応をしてお客様をお迎えいたします。<br>当店は、新型コロナウイルス対策認証店です。<br>2時間程度でのお食事をお願いいたします。',
    },
  ],
};
```

# 実装例 1 : こんなNodeを実行してBuild

TypeScript

```
const targets = ['index.html'];
const items: IItem[] = [news, redWine, whiteWine, champagneWine, announcement];

const build = () => {
  execSync(`rm -r -f ${process.cwd() + '/www'}`);
  mkdirSync(process.cwd() + '/www');
  execSync(`cp -r ${process.cwd() + '/src/template'}/* ${process.cwd() + '/www'}`);

  for (const templateFile of targets) {
    let template = readFileSync(process.cwd() + '/www/' + templateFile).toString('utf-8');
    for (const item of items) {
      const insertData: string[] = [];
      for (const line of item.data) {
        let template = item.template;
        for (const [key, value] of Object.entries(line)) {
          const re_url = /((ftp|http|https):\/\/(\w+:{0,1}\w*@)?(\S+)(:[0-9]+)?(\/|\/([\w#!:.?+=&%@\-\\/]))?)/
          gi;
          const replaceText = (value as string).replace(re_url, '<a href="$1" target="_blank"
          rel="noreferrer">$1</a>');
          template = template.replace(new RegExp(`{{ ${key} }}`, 'g'), replaceText as string);
        }
        insertData.push(template);
      }
      const changeSelector = item.selector.replace(/></g, `>${insertData.join('')}<`);
      template = template.replace(item.selector, changeSelector);
    }
    writeFileSync(process.cwd() + '/www/' + templateFile, template);
  }
};

build();
createWineSearchIndex();
apiBuild();
```

## 実装例 2 : JavaScriptフレームワーク

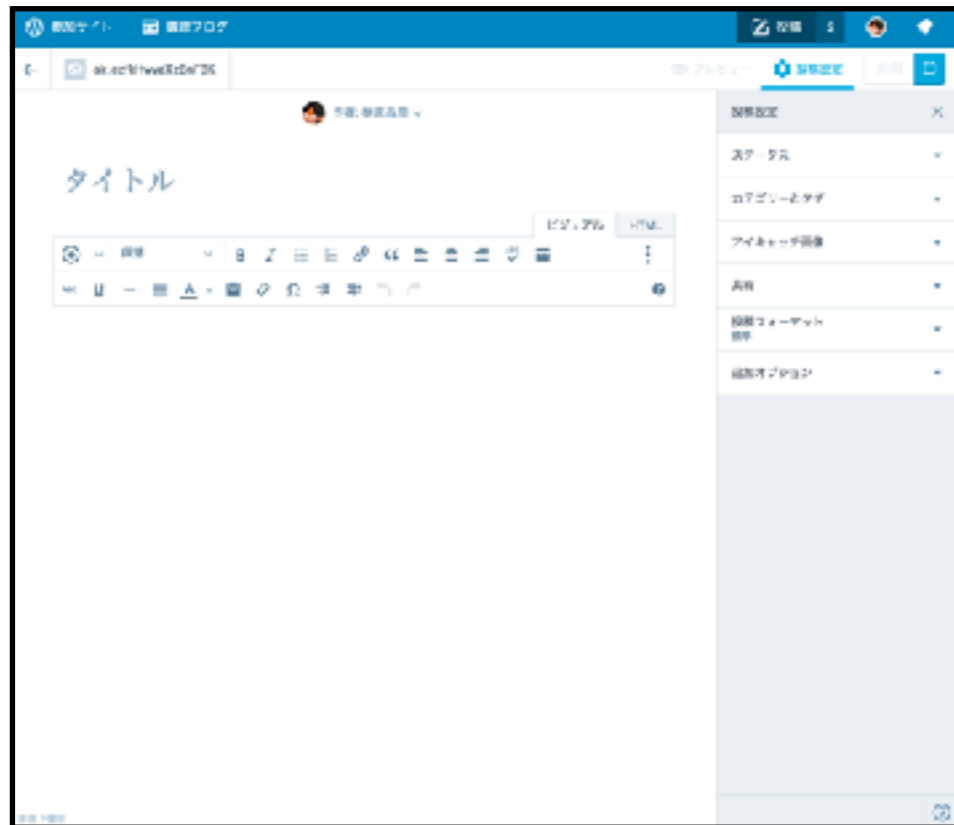
JSX

```
<h4>
  
  お知らせ
</h4>
<ul class="list-news">
  {news.map(d => <li><label class="form-label" innerHTML={d.title}></label><p
    innerHTML={d.body}></p>
  </li>)}
</ul>
```

これだけで書けちゃう。簡単！

## 2. 同一テンプレートで量産した記事を管理するため

ヘッドレスCMSを使って動的にデータ取得しよう



WordPress.com



microCMS

kintone





## 2. 同一テンプレートで量産した記事を管理するため



```
{
  path: ':id',
  component: ArticlePage,
},
```

```
this.route.paramMap.subscribe(async (params: ParamMap) => {
  this.articleId =
    parseInt(params.get('id'), 10);
  const {article} = await
    this.articleService.getCategory(
      this.articleId
    );
  this.article = article;
});
```

```
<article>
  <h3 [innerHTML]="item.title"></h3>
  <p class="attribute">
    <ion-note>{{ item.date | date:
      "yyyy年MM月dd日" }}</ion-note>
    <ion-note
      *ngIf="item.count" > /
    {{ item.count }}Point</ion-note>
  </p>
  <p
    class="excerpt" [innerHTML]="item.
    excerpt"></p>
</article>
```

## 2. SSG/SSR/prerender.ioを使ってレンダリングもできる

	内容
CSR	ユーザ環境でHTMLをレンダリング
SSG	サーバホスティング前にHTMLをレンダリング
SSR	リクエスト時にHTMLをレンダリング
prerender.io	特定IPからのリクエスト時だけ、サーバがレンダリング済みHTMLを返却

## 技術の課題は技術で解決しよう

動的言語が走るサーバの責任バックアップやサーバ負荷のコントロールユーザが壊したらどうしようetc



もちろんリスクマネジメントしながらCMS構築もありますが、静的サイト化することでこれらの問題を乗り越えられる可能性があるので一度考えてみませんか。