

2018年10月10日

株式会社インプレスR&D

<https://nextpublishing.jp/>

複雑化した環境構築がさらっとわかる入門書！

『ネコミミでもわかるフロントエンド開発環境構築』発行

技術書典シリーズ・10月の新刊

インプレスグループで電子出版事業を手がける株式会社インプレス R&D は、『ネコミミでもわかるフロントエンド開発環境構築』(著者: 汐瀬なぎ)を発行いたします。

『ネコミミでもわかるフロントエンド開発環境構築』

<https://nextpublishing.jp/isbn/9784844398592>



著者: 汐瀬なぎ

小売希望価格: 電子書籍版 1400 円(税別) / 印刷書籍版 1600 円(税別)

電子書籍版フォーマット: EPUB3 / Kindle Format8

印刷書籍版仕様: B5 判 / カラー / 本文 78 ページ

ISBN: 978-4-8443-9859-2

発行: インプレス R&D

<< 発行主旨・内容紹介 >>

【複雑化したフロントエンド開発環境の構築がこの1冊でさらっとわかる！】

本書は、React や Vue.js などに代表される新しいフレームワークの到来により最適化された Web アプリの開発ができるようになったにもかかわらず、それぞれの開発環境の構築が複雑化したことで技術選定に苦勞しているエンジニアに向けた「サクッとわかる」開発環境構築ガイドです。

ベストプラクティスを提示するのではなく、環境構築に慣れていないエンジニアがとりあえず自分なりの最適なツールを選ぶための入り口に立つことができる入門書です。

《本書の対象読者》

- モダンな JavaScript を書きたいが、書き方がよくわからない
- Babel や Webpack の設定方法がいまいちわかっていない
- ESLint、Flow、Jest などを導入したいが、方法がわからない

(本書は、次世代出版メソッド「NextPublishing」を使用し、出版されています。)

開発環境の構築を準備段階から丁寧に解説

第1章 まずは準備から

本章では、まず環境構築を始める前に、プロジェクト全体のディレクトリ構成や package.json の初期設定について解説します。

1.1 package.jsonファイルの作成

さっそくプロジェクトを作っていきます。好きなディレクトリに移動したら(今回は nekomimi-frontend としました)、npm または Yarn コマンドを使って、package.json ファイルを作成します。対話形式で設定項目について聞かれますが、特に入力する必要はありません。

```
$ mkdir nekomimi-frontend && cd nekomimi-frontend
$ yarn init // または npm init
```

作成された package.json ファイルは次のリスト 1.1 のようになります。

リスト 1.1: package.json の中身

```
1: {
2:   "name": "nekomimi-frontend",
3:   "version": "1.0.0",
4:   "main": "index.js",
5:   "license": "MIT"
6: }
```

Yarn ってなに?

JavaScript のプロジェクトを作る際には、一般的に npm と呼ばれるパッケージマネージャーが利用されますが、最近では npm 互換である Yarn が人気を博しています。本書では、基本的に Yarn の利用をオススメしています。

npm とどこが違うの?

Yarn ではパッケージをインストールした際に、yarn.lock というファイルが自動的に生成されます。

```
1 https://www.npmjs.com/
2 https://yarnpkg.com/en/
3 本書では主に Yarn を紹介していますが、npm を利用する場合は、適切なコマンドを修正してください。
4 yarn をインストールしてください。
```

れます¹。yarn.lock ファイルは簡単に説明すると、インストールしたパッケージのバージョンを記録したファイルです。yarn.lock があることで、どの状態でインストールしてもまったく同じバージョン構成でインストールが行われます。これは、たとえばビルドを利用してテストやビルドを行う際に役立ちます。

```
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
```

1.2 ディレクトリ構成について

基本的なディレクトリ構成を次のリスト 1.2 に示します。

リスト 1.2: nekomimi-frontend のディレクトリ構成

```
./
├── dist/
├── node_modules/
├── src/
│   ├── css/
│   ├── html/
│   └── js/
├── package.json
└── yarn.lock
```

• dist/ ディレクトリ以下は、変換された .js ファイルや .css ファイルが出力されるディレクトリです。第 2 章で開発サーバーを立ち上げる際に、参照されるディレクトリがここになります。

• src/ ディレクトリ以下で開発を行います。分かりやすいように、css・html・js はそれぞれディレクトリを分けるとよいでしょう。

また、ここではまだ記述していませんが、以降以降で多くのコンフィグファイルを用意していくことになります。

1.2.1 .gitignore

Git を使用して開発を行う場合、.gitignore ファイルを用意しておきましょう。gitignore ファイルはリスト 1.3 のように、node_modules/ ディレクトリのほか、dist/ ディレクトリを除外するようにしておきます。

リスト 1.3: .gitignore の設定

```
# node modules
```

1 https://git-scm.com/docs/gitignore#_excluding_patterns

6 | 第1章 まずは準備から | 7

JavaScript をより効率的にコーディングするためのツールを紹介

第3章 JavaScriptのためのパワフルなツール

第 2 章では、JavaScript をトランスパイルするためのツールとして Babel と、開発サーバーを立ち上げるビルドを行うためのツールとして webpack について解説しました。本章では JavaScript をよりきれいに、かつ速く、そして直観的に書くためのツールを紹介し、そのための環境を構築していきます。

3.1 ESLint

ESLint は JavaScript のための、静的構文チェックツールです。ESLint を導入することで、文法的な間違いや、(カンマ) のつけ忘れといったミスを防ぐことができます。これらの間違いは JavaScript のエラーを引き起こすこととなるので、事前に構文チェックを済ませることでエラーの発生を大幅に減らすことができます。

3.1.1 ESLint を導入する

それでは実際に ESLint を導入していきます。基礎となる eslint モジュールと、あらかじめルール(=正しい構文)が定義されているコンフィグモジュール(今回は eslint-config-airbnb)、そしてこれらの依存モジュールを導入します。

eslint-config-airbnb は Airbnb が提供している ESLint のルールセットです。詳しいルールの内容は <https://github.com/airbnb/javascript> から見る事ができます。eslint-config-airbnb を導入する場合、依存モジュールを eslint-config-airbnb が指定するバージョンに合わせる必要があります。依存モジュールのバージョンを確認するには、次のコマンドを実行します。

```
$ yarn info eslint-config-airbnb peerDependencies
```

```
{
  eslint:
    '^4.19.1 || ^5.3.0',
  'eslint-plugin-import':
    '^2.14.0',
  'eslint-plugin-jsx-a11y':
    '^6.1.1',
}
```

1 https://eslint.org/
2 https://www.npmjs.com/package/eslint-config-airbnb
3 詳しくは ESLint のドキュメントを参照してください。
4 https://github.com/airbnb/javascript/tree/master/packages/eslint-config-airbnb を参照。

```
'eslint-plugin-react':
  '^7.11.0' }
```

指定バージョンが確認できたら、そのバージョンで依存モジュールをインストールしましょう。

```
$ yarn add --dev eslint-config-airbnb \
> eslint@5.3.0 \
> eslint-plugin-import@2.14.0 \
> eslint-plugin-jsx-a11y@6.1.1 \
> eslint-plugin-react@7.11.0 \
> babel-eslint
```

babel-eslint は Babel 用の ESLint パーサーです。Babel でトランスパイルを行っている場合、ESLint の標準パーサーでは対応できないことがあるため、導入します。

3.1.2 eslintrc の作成

ESLint を動かすには、コンフィグファイルとしてルートディレクトリ直下にある .eslintrc ファイルを用意します (リスト 3.1)。

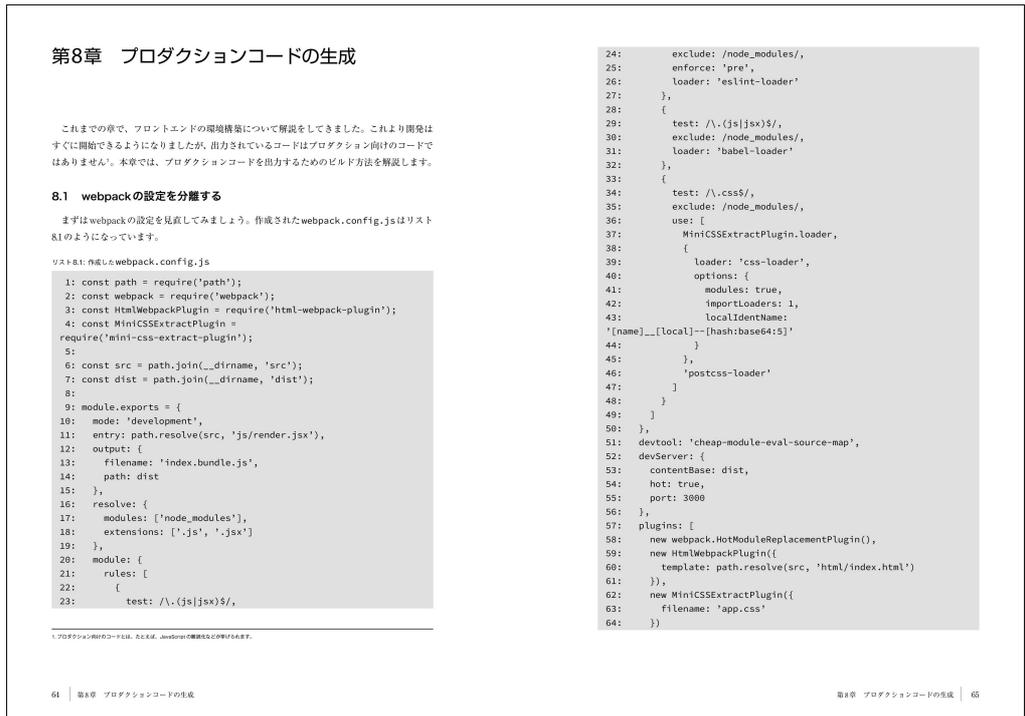
リスト 3.1: .eslintrc を作成する

```
1: {
2:   "extends": ["airbnb"], # airbnb のルールを継承します
3:   "env": {
4:     "browser": true, # ブラウザでのグローバル変数を有効化します
5:     "es6": true # es6 (es2015) の構文を有効化します
6:   },
7:   "parser": "babel-eslint", # babel-eslint をパーサーとして使用
8:   "parserOptions": {
9:     "ecmaFeatures": {
10:      "jsx": true # jsx を有効化します
11:    }
12:   }
13: }
```

.eslintrc が用意できたら、package.json に lint:js コマンドを追加します (リスト 3.2)。

1 https://github.com/babel/babel-eslint

18 | 第3章 JavaScriptのためのパワフルなツール | 19



<<目次>>

第1章 まずは準備から

1.1 package.json ファイルの作成

1.2 ディレクトリー構成について

1.3 EditorConfigを導入する

第2章 JavaScript を動かす

2.1 Babel

2.2 webpack

第3章 JavaScript のためのパワフルなツール

3.1 ESLint

3.2 Prettier

3.3 Flow

第4章 React をはじめる

4.1 React をブラウザで表示させるための準備

4.2 最初で最後の React コンポーネント

第5章 CSS を適用する

5.1 webpack の設定

5.2 CSS ファイルの作成

5.3 StyleLint

5.4 PostCSS

第6章 React と CSS の連携

6.1 CSS をコンポーネントに適用する

6.2 CSSModules

第7章 爆速でテストを書く

7.1 Jest で始めるユニットテスト

7.2 コンポーネントはスナップショットテストで

7.3 コードカバレッジを見る

第8章 プロダクションコードの生成

8.1 webpack の設定を分離する

8.2 CSS を minify する

<< 著者紹介 >>

汐瀬なぎ

フロントエンド UX エンジニア。React をメインとした Web アプリケーション開発の他、アプリケーションのデザイン制作、UI/UX の改善も行う。最近は ReactNative によるアプリ開発や、GatsbyJS を用いた Web サイト構築について勉強中。趣味は絵を描くこと。

<< 販売ストア >>

電子書籍:

Amazon Kindle ストア、楽天 kobo イーブックストア、Apple Books、紀伊國屋書店 Kinoppy、Google Play Store、honto 電子書籍ストア、Sony Reader Store、BookLive!、BOOK☆WALKER

印刷書籍:

Amazon.co.jp、三省堂書店オンデマンド、honto ネットストア、楽天ブックス

※ 各ストアでの販売は準備が整いしだい開始されます。

※ 全国の一般書店からもご注文いただけます。

【株式会社インプレス R&D】 <https://nextpublishing.jp/>

株式会社インプレス R&D (本社：東京都千代田区、代表取締役社長：井芹昌信) は、デジタルファーストの次世代型電子出版プラットフォーム「NextPublishing」を運営する企業です。また自らも、NextPublishing を使った「インターネット白書」の出版など IT 関連メディア事業を展開しています。

※NextPublishing は、インプレス R&D が開発した電子出版プラットフォーム(またはメソッド)の名称です。電子書籍と印刷書籍の同時制作、プリント・オンデマンド(POD)による品切れ解消などの伝統的出版の課題を解決しています。これにより、伝統的出版では経済的に困難な多品種少部数の出版を可能にし、優秀な個人や組織が持つ多様な知の流通を目指しています。

【インプレスグループ】 <https://www.impressholdings.com/>

株式会社インプレスホールディングス(本社:東京都千代田区、代表取締役:唐島夏生、証券コード:東証1部9479)を持株会社とするメディアグループ。「IT」「音楽」「デザイン」「山岳・自然」「旅・鉄道」「学術・理工学」を主要テーマに専門性の高いメディア&サービスおよびソリューション事業を展開しています。さらに、コンテンツビジネスのプラットフォーム開発・運営も手がけています。

【お問い合わせ先】

株式会社インプレス R&D NextPublishing センター

TEL 03-6837-4820

電子メール: np-info@impress.co.jp