

# 次世代フロントエンドツールVite 入門会

2021/10/30(Sat) 10:00～ / 主催: @Meijin\_garden

<https://connpass.com/event/227990/>

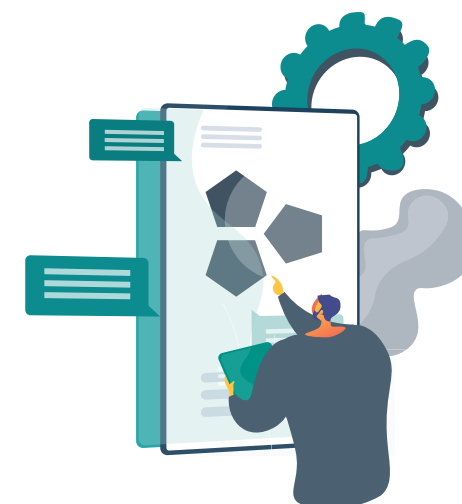


# イベント概要

- **環境構築** - Viteの公式ガイドに則って、環境構築から一通りの機能を使うところまで進めます
- **プラスアルファ** - 終わったらStorybookを入れたり、プラグインを入れたり、Vercelにホスティングしたり各自興味のある活動を行います

※なんとなくイベント立ち上げたらめっちゃ登録数があるとおののいています

※なので、**チャットや会話で盛り上げて**いただけると助かりますw



# 予定

- 自己紹介
  - 主催
  - 話す枠の方々
- 画面共有して環境構築開始
  - HMR, pre-bundle, plugin, CSS Modules, env variables and mode...など
  - あくまで入門会なので、公式ドキュメントを順になぞります
- 感想など雑談
  - この場で皆に質問したいことがあればチャット等で募集とか
- 時間に余裕があればプラスアルファなにかやる
- 解散



# 自己紹介

- 「名人」というニックネームで活動
- Twitter: [@Meijin\\_garden](https://twitter.com/Meijin_garden)
- Webエンジニア6年目
- 好きな分野はWebフロントエンド
- 好きなエディタのテーマはPanda
- 趣味は将棋（初段くらい）
- 株式会社NoSchool CTO
  - オンライン家庭教師マナリンク -> <https://manalink.jp>



# 話す枠の方々の自己紹介

- お名前
- 好きな技術
- Viteとの関わり





**Viteに入門  
しよう！**



# create vite app

- <https://vitejs.dev/guide/#scaffolding-your-first-vite-project>

React & TypeScriptで書きたい場合

```
yarn create vite react-application --template react-ts
```

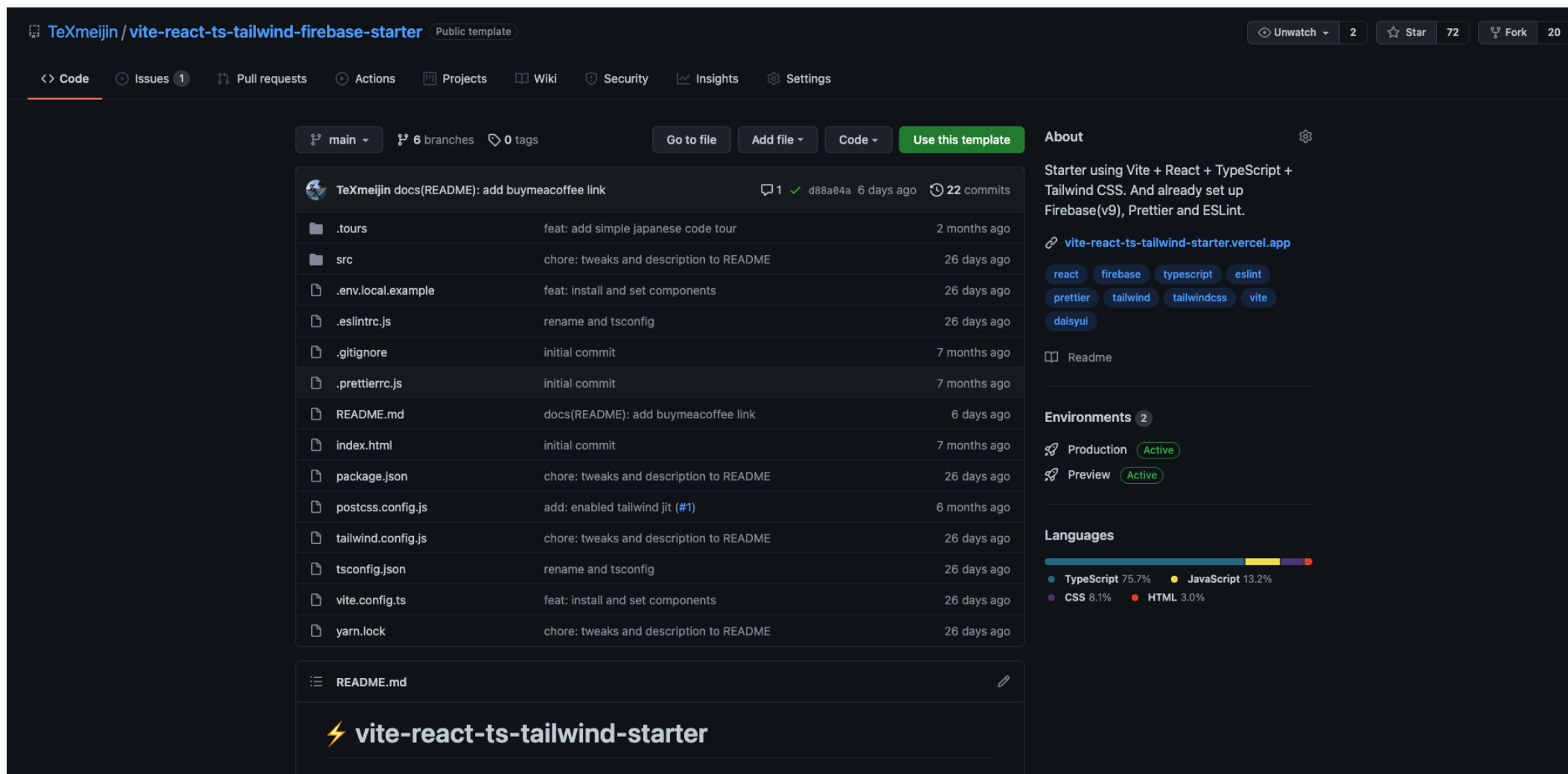


# use templates (optional)

- degitを使えばtemplateから簡単に開始できます。

```
npx degit TeXmeijin/vite-react-ts-tailwind-firebase-starter vite-hoge
```

- 以下は自作のテンプレです



The screenshot shows the GitHub repository page for 'vite-react-ts-tailwind-firebase-starter' by TeXmeijin. The repository is a public template with 2 stars and 20 forks. The main branch is 'main' with 6 branches and 0 tags. The repository contains a README, a src directory, and various configuration files. The README is titled 'vite-react-ts-tailwind-starter' and describes the project as a starter using Vite + React + TypeScript + Tailwind CSS, already set up with Firebase(v9), Prettier, and ESLint. The repository also includes a 'Use this template' button and a 'Readme' link. The 'About' section provides more details about the project, including a link to the Vercel app and a list of technologies used: react, firebase, typescript, eslint, prettier, tailwind, tailwindcss, vite, and daisyui. The 'Environments' section shows 'Production' and 'Preview' environments, both of which are active. The 'Languages' section shows the following distribution: TypeScript 75.7%, JavaScript 13.2%, CSS 8.1%, and HTML 3.0%.

File	Commit Message	Commit Date
.tours	feat: add simple japanese code tour	2 months ago
src	chore: tweaks and description to README	26 days ago
.env.local.example	feat: install and set components	26 days ago
.eslintrc.js	rename and tsconfig	26 days ago
.gitignore	initial commit	7 months ago
.prettierrc.js	initial commit	7 months ago
README.md	docs(README): add buymeacoffee link	6 days ago
index.html	initial commit	7 months ago
package.json	chore: tweaks and description to README	26 days ago
postcss.config.js	add: enabled tailwind jit (#1)	6 months ago
tailwind.config.js	chore: tweaks and description to README	26 days ago
tsconfig.json	rename and tsconfig	26 days ago
vite.config.ts	feat: install and set components	26 days ago
yarn.lock	chore: tweaks and description to README	26 days ago





# yarn dev

```
yarn  
yarn dev
```



Hello Vite + React!

count is: 0

Edit App.tsx and save to test HMR updates.

[Learn React](#) | [Vite Docs](#)



～ファイル更新時の反映の速さを  
体感する時間～



# vite devは型チェックしません

- <https://vitejs.dev/guide/features.html#typescript>
  - Vite only performs transpilation on .ts files and does **NOT** perform type checking.
  - tscはCIやIDEが別途実行してくれるよねっていう考え
- ``npx esbuild src/App.tsx``
  - 爆速でJavaScriptへの変換が終わる
  - むしろ型情報(+JSX)を落とすことだけをesbuildがやっている
    - [https://github.com/evanw/esbuild/blob/master/internal/js\\_lexer/js\\_lexer.go](https://github.com/evanw/esbuild/blob/master/internal/js_lexer/js_lexer.go)



# index.html

- SPAなのでHTMLはindex.htmlのみ
- 基本的にはドキュメントルートにindex.htmlがあります
- `<html>` の`lang="ja"`とかはここで変える



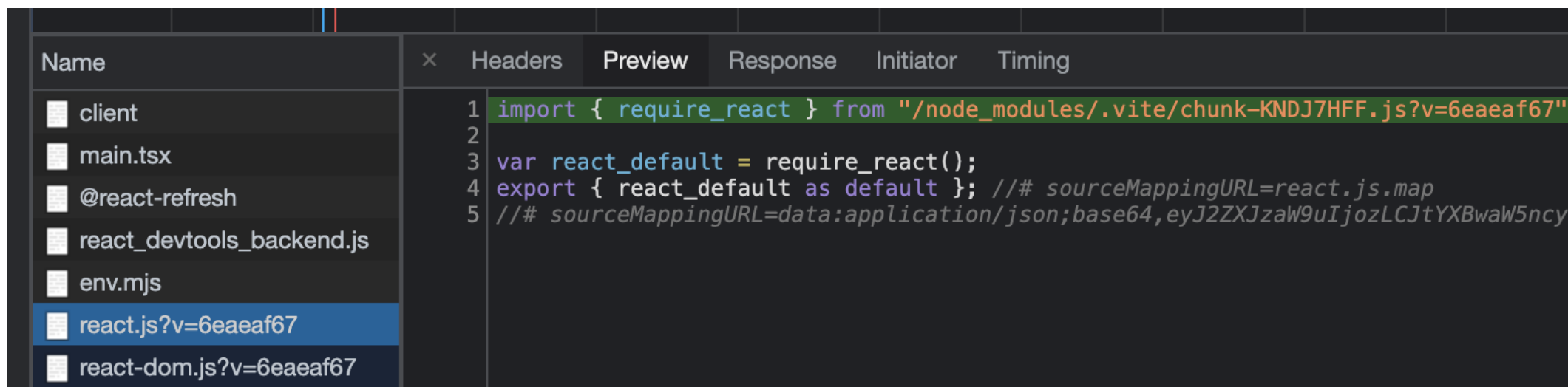
# pre-bundle

- <https://vitejs.dev/guide/features.html#npm-dependency-resolving-and-pre-bundling>
- esbuildが依存をバンドルして`node\_modules/.vite`以下に配置してくれる

Pre-bundling dependencies:

```
react
react-dom
react/jsx-dev-runtime
```

(this will be run only when your dependencies or config have changed)



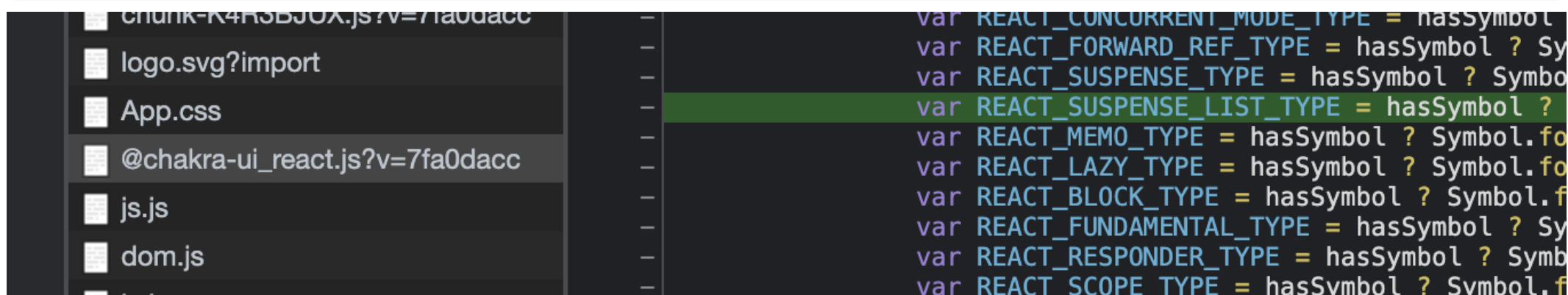
# pre-bundle (2)

- 起動中にライブラリを追加し、新しいimportを追加すると再度Prebundleが走るっぽい

```
yarn add @chakra-ui/react @emotion/react @emotion/styled framer-motion
```

```
import { Button, ButtonGroup } from "@chakra-ui/react"
// 本当はProviderとか要るけど雑に追加
<Button>
  hoge
</Button>
```

```
18:26:01 [vite] ✨ dependencies updated, reloading page...
```



# pre-bundle (3)

- TSXコンポーネントを作ったり、Hookを切ってimportすると、それらも別個のファイルでリクエストされる
- 手元の実運用中のViteアプリケーションでは1画面で300以上のリクエストが飛んでる
  - 手元のNuxtアプリケーションでは170程度なので多いほうと思われる
  - 依存しているファイル数だけリクエストが増えるので当然といえば当然



# plugin

- vite本体はシンプルな機能で、プラグインを足していく

Reactの場合のデフォルトは:

```
import { defineConfig } from 'vite'
import react from '@vitejs/plugin-react'

// https://vitejs.dev/config/
export default defineConfig({
  plugins: [react()]
})
```





# plugin (2)

たとえばSentryの場合

```
import viteSentry from 'vite-plugin-sentry';

export default defineConfig(({ mode }) => {
  return {
    plugins: [react(), viteSentry(getSentryConfig(mode))],
```

- mode(後述)を受け取れる関数として定義するとmodeごとに違うSentry設定を反映できる



# plugin (3)

- 本番ビルドのVisualize方法気になりますよね？
- 設定は以下の記事がおすすめ（手前味噌）

<https://zenn.dev/meijin/articles/vite-bundle-analyzer>



## Vite製のアプリケーションにBundle Analyzerを設定する

📅 2021.10.21に公開 ⌚ 4 min read

  
12  
  


### 概要

ViteでセットアップしたアプリケーションにBundle Analyzerを設定する方法です。

Webpackには[Webpack Bundle Analyzer](#)があり、Nuxt.jsで開発しているならば `nuxt build -a` で実行できるようにデフォルトで組み込まれていることから利用したことのある人は多いと思います。

### Topics

 JavaScript

 React

 パフォーマンス

 Vite

 Tech



# CSS Modules

- デフォルトでCSS Modulesに対応してます(すごい)
- <https://vitejs.dev/guide/features.html#css-modules>
- `.scss` ももちろん対応できます

```
yarn add -D sass
```

たとえば↓

```
import { ReactNode } from 'react';
import styles from './ErrorText.module.scss';

export const ErrorText = ({ children }: { children: ReactNode }) => <span
```



# env variables

- みんな気になる環境変数
- <https://vitejs.dev/guide/env-and-mode.html>

```
VITE_FIREBASE_APIKEY=hogehogehogehogehogehoge
```

- `VITE\_` から始まる変数だけクライアント側に放出される
- 環境変数へのアクセスはデフォでは型安全でないが、  
`env.d.ts` といったファイルを定義してstringにしておく  
とよい



# modes

- ``vite build``でProduction Buildができるけど、検証環境と本番環境で環境変数などを切り分けたいケースはある
- ``vite build --mode staging``といった``mode``引数が指定できる。便利。CIもこれで組める
- ``.env.[mode]``のファイル名の環境変数を読み込んでくれる
- ``vite.config.ts``で指定するconfigを関数にできるのでその引数にmodeが含まれる

```
// https://vitejs.dev/config/  
export default defineConfig(({ mode }) => {  
  return {
```



# まとめ

- だいたいこれで実装するときに必要な知識は網羅できた...？
- 個人的に良いなと思っているところ
  - とにかくHMRの反映が速い
  - シンプルかつ拡張可能な仕組み
  - 本番ビルドはrollupなので安定してる
    - <https://vitejs.dev/guide/why.html#why-bundle-for-production>



自由時間に入る前に...

# 告知

# マナリンクではエンジニアを採用中です！

オンライン家庭教師マナリンク(<https://manalink.jp>)は以下の課題を解決しています。

## 先生を直接指名できない

多くの家庭教師サービスは、先生ではなく「運営会社に資料請求」するので体験指導と本指導の先生が違うこともあるなど、先生にこだわりたいご家庭にとって不満が残ります。マナリンクでは**すべての先生のプロフィールを作成し、YouTube上に自己紹介動画も公開**しています。もちろん直接指名できます。



## 保護者から指導が見えない

教育サービスの大きな課題は**価値を受ける人（生徒）と対価を支払う人（親）が別人**なことです。マナリンクでは専用のReact Native製アプリで先生-生徒間の指導を実施していただき、**保護者さん**に**閲覧権限を与えることで指導内容を可視化**、解約を防ぎます。



## 職業自体の認知度が低い

オンライン家庭教師自体はここ数年でZoom等の普及によって生まれた新しい職業です。そのため、オンライン家庭教師という仕事を普及し**より多くの方のキャリアの選択肢に入れてもらう**ため、マナリンクTeachersというメディアを公開・運営しています。Next.js × microCMS製です。





# カジュアル面談を随時受付中

詳しく訊きたいと思っていただいた方はMeetyでぜひご連絡を！

<https://meety.net/matches?q=%23マナリンク>

# テックブログも公開中

採用技術はTypeScript(React, React Native, Vue, AWS CDK, Vite, Firebase), PHP(Laravel), AWSなどです。

<https://zenn.dev/mandalink>



# 個人的な告知

- Twitterフォローしてね
  - [@Meijin\\_garden](https://twitter.com/Meijin_garden)
- ZennでViteの記事なども書いているので、本日の内容が勉強になったという方がいらっしゃったらサポートください💰
  - <https://zenn.dev/meijin>



# 自由時間

- ・ 感想など雑談
- ・ プラスアルファ

デプロイ、Storybook、プラグイン探しなどなど

