

デザイン開発のためのドキュメント

はじめに

プログラミングを用いたデザイン開発ではMUIとtailwind cssを使って開発してもらいます。何もわからなくても今からここにこの世のすべてを書くので安心してください。

環境構築

まずはいくつかのソフトのインストールと設定、環境構築です。主にインストールするものはVisual Studio Code, Node.js, Gitです。よっぽどPCの容量がないとかではない限り大丈夫なはずです。

あと焦らずひとつずつ軽く全体に目を通した後にやってください。僕も早とちりしておしまいミスすることがよくあるのですが、早とちりするとまあめんどくさいです。直すの

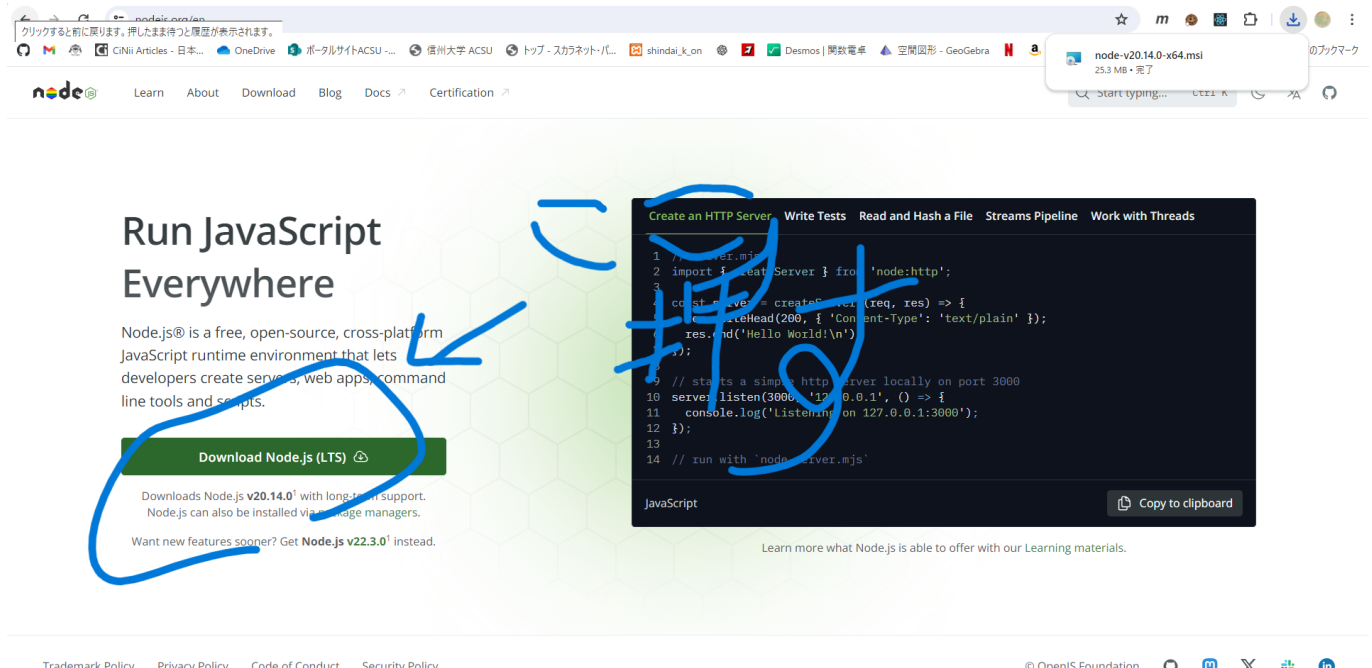
Node.js編

はじめから聞いたことも見たこともない意味不明の単語が出てきてびっくりしていると思いますが、Node.jsのインストールを行います。これは本来Web上でしか動かないJavaScriptというWeb開発のためのプログラミング言語をあなたのPC上で動かすためのソフトです。

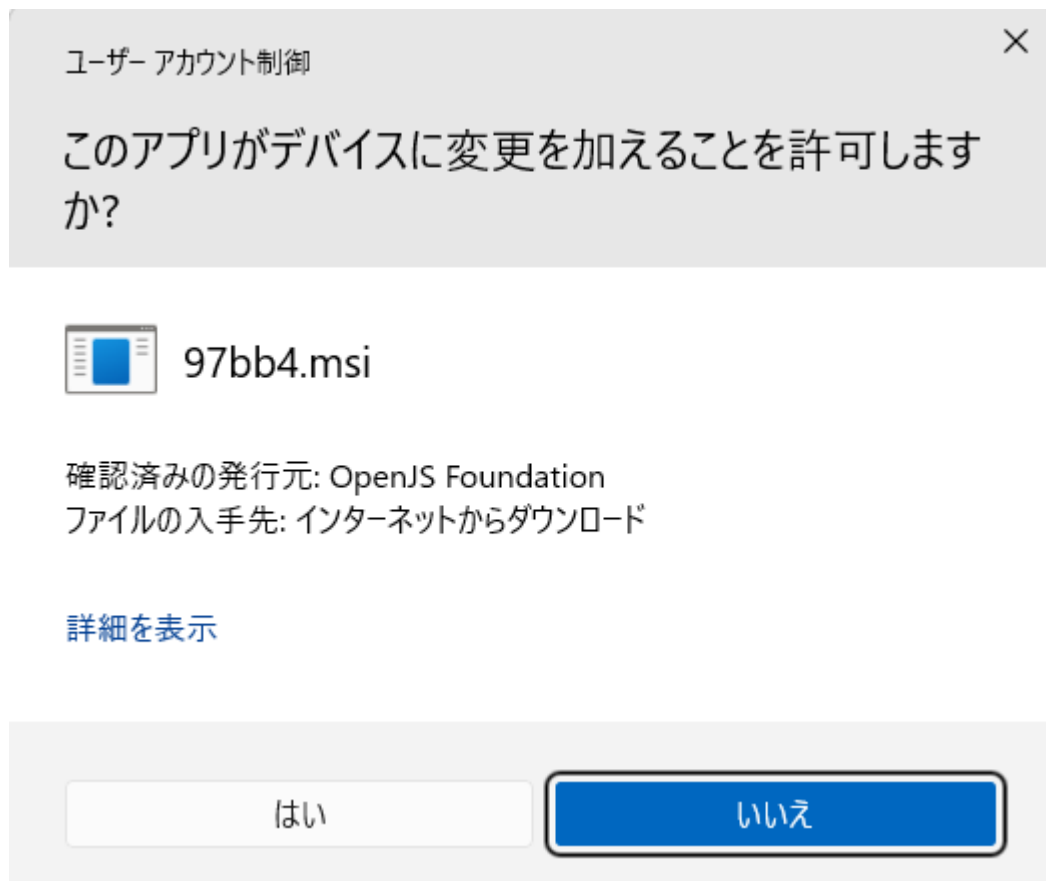
基本的なインストール方法は以下のページを見てやってください

Node.jsのインストール方法(Qiita)

Node.jsのホームページの見た目が上記のページにある画像と違いますが下の画像の部分をクリックしてくれば大丈夫です。



さっきのQiitaのページの手順通り進めていってください。インストール途中で下の画像みたいなのが出てきたら「はい」をクリックしてください。

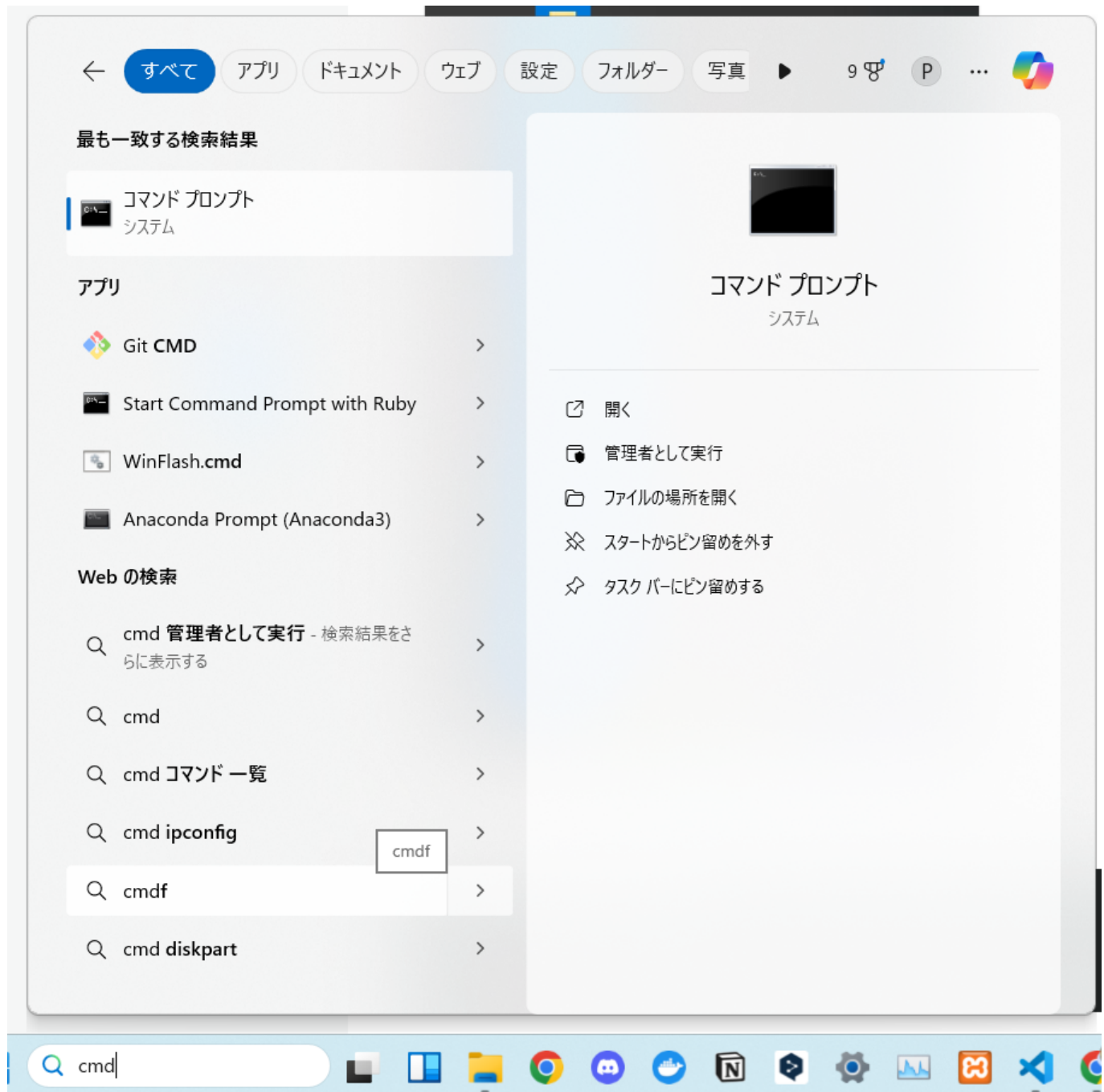


「はい」を押しましたね？ウイルスでした！！！！！！！！！！！！！！！！ざまあみ
ろ！！！！！！！！！！！！！！！！人を疑うことを覚えるんだな！！！！！！！！！！！！ぎ
ゃははwwwwwwwwwwwwww

あ、いや、嘘っすすんません...ほんますんません...

さっきのQiitaのページの手順通りやると「ちゃんと動くのか確認する」段階に入るのですがここは無視しても構いません。

やりたい場合は下の画像のようにwindowsの下の検索バーから「cmd」と検索するとコマンドプロンプトという標準搭載されているソフトが出てくるのでそれを開いてください。

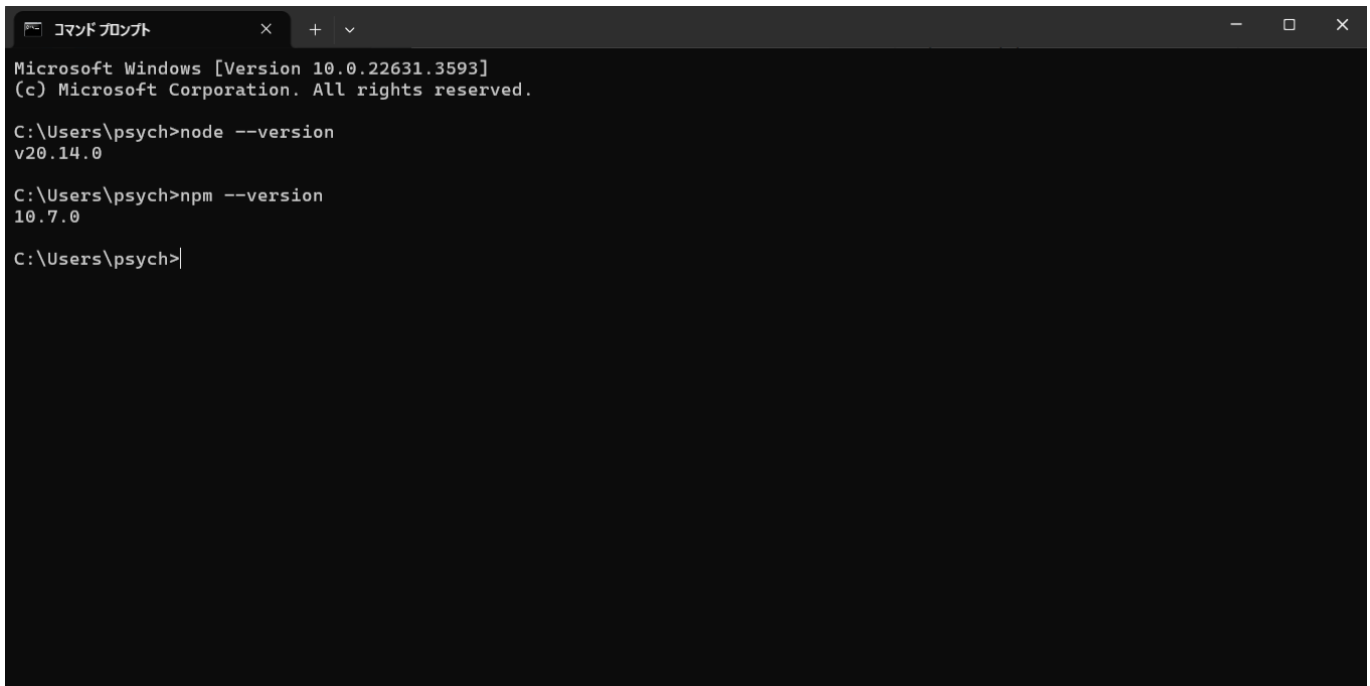


これをクリックするとドラマのハッカーが使ってるみたいな画面が出てくと思うので、下の画像のように

```
node --version
```

```
npm --version
```

と打ってエンターキーを押してみてください。打つのめんどくさかったらこれコピペしてください。



```
Microsoft Windows [Version 10.0.22631.3593]
(c) Microsoft Corporation. All rights reserved.

C:\Users\psych>node --version
v20.14.0

C:\Users\psych>npm --version
10.7.0

C:\Users\psych>
```

画像のようになんか数字が出てきたら成功です。それ以外で赤文字とか出てきたらわたべに相談してください。

これでNode.js編終了です。一旦タバコ休憩でも挟みましょう。

Git, GitHub編

次にGitのインストールとGitHubのアカウント作成、リポジトリのクローンです。また意味不明の単語がつらつらと並んでますね。頭痛がしてくるころだと思うのでロキソニンなどを飲みましょう。

Git、GitHubは書いたプログラミングのコードの編集履歴を保存するためのシステムです。これを使うとみんなと同じコードを編集することが容易になります。うれしいですね。

まあとりあえず[このページ](#)見ながらこっちも見つつやってください。

上記のページの2が終わったらこっちにいったん帰ってきてgitの初期設定を行います。さっきのコマンドプロンプトというソフトをもう一度立ち上げてください。お察しかもしれませんがこのコマンドプロンプトめちゃくちゃ使うのであんまビビらないでください。

そのコマンドプロンプトでまずは

```
git --version
```

を実行しましょう。node.jsでやったことと同じで、ちゃんとインストールされてると

```
git version 2.35.3.windows.1
```

とかって出力されます。なんか数字とかが違ってても気にしないでください。赤い文字とか出てきたらこれもわたべに相談してください。

続いて

```
git config --global user.name "(好きな名前)"  
git config --global user.email "(メールアドレス)"
```

を実行します。焦らず一行ずつでいいので実行してください。ちなみに名前とメールアドレスは本当に何でもいいです。僕は

```
git config --global user.name "watabegg"  
git config --global user.email "roy_wtnb@outlook.jp"
```

で登録してます。やべ、メアドばれてもた。

これでGitの設定は終わりです。まあいったんタバコ吸うか

続いてGitHubのアカウントを作ります。まあ難しいことはないので適当にやります。

GitHubのページ

このページにアクセスしてsign upからページの指示に従ってアカウントを作ってください。作れたらそのユーザーネームをわたべに教えてください。

アカウントが出来たらまたコマンドプロンプトを開いてください。コマンドプロンプトの画面は現状左の端に

```
C:/Users/(Windowsのユーザー名)>
```

って書いてあると思います。ここにこんな感じのことが書いてない人は

```
cd C:/Users/(自分のWindowsのユーザー名)
```

を実行してください。ほとんどの人はこれを実行する必要はないはずです。

これが大丈夫であれば

```
git clone https://github.com/watabegg/k_on_line.git
```

を実行してください。そうすると

```
C:\Users\psych>git clone https://github.com/watabegg/k_on_line.git
Cloning into 'k_on_line'...
remote: Enumerating objects: 676, done.
remote: Counting objects: 100% (189/189), done.
remote: Compressing objects: 100% (120/120), done.
remote: Total 676 (delta 75), reused 121 (delta 49), pack-reused 487
Receiving objects: 88% (595/676)
Receiving objects: 100% (676/676), 842.98 KiB | 4.28 MiB/s, done.
Resolving deltas: 100% (295/295), done.

C:\Users\psych>|
```

こう出てきたら完璧です！ナイス！最高！！出来なかったそこのあなた！地獄行きです！わたべに連絡してください。

ではこれでGit, GitHub編は終了です。二郎とか食べてからタバコを吸いましょう。

GitHub Tips

ちなみにGitHubのAIを使ってプログラミングを爆速にする方法があります。

[GitHub Studentに登録してGitHub Copilotを使う方法](#)

これやっくとくと便利です。学生証の英訳とかいう今後二度とやるわけない作業がありますが...

Visual Studio Code編








最後にプログラミングのコードを書くためのツール、Visual Studio Code(以下VS Code)をインストールしてもらいます。丸投げですがこれ見てやってください

VS Codeのインストール方法

VS Codeをインストールして日本語化まで出来たら次です。

VS Codeにいくつかの拡張機能を入れていただきます。まずはGit Graphです。

Git Graphはみんながgitにした変更をツリー上に見ることができる拡張機能で、これがかなり便利なので入れましょう。

Branches: Show All		<input checked="" type="checkbox"/> Show Remote Branches	🔍 📄 ⚙️ 🔄		
Graph	Description	Date	Author	Commit	
Uncommitted Changes (1)		18 Jun 2024 10:...	*	*	
 develop-main	origin わかんないけど変更	18 Jun 2024 09:...	watabegg	724e4f7c	
 main	origin ドキュメントを追加	18 Jun 2024 09:...	watanegg	fca6c4eb	
バックエンドミスってたので改修		18 Jun 2024 01:...	watabegg	f0088764	
Merge pull request #17 from watabegg/develop-main		17 Jun 2024 09:...	watabegg	2f105b54	
Merge branch 'main' into develop-main		17 Jun 2024 09:...	watabegg	7de94553	
シェルの追加とちょっと編集		15 Jun 2024 23:...	watabegg	886836e9	
ログを登録順にソート		15 Jun 2024 21:...	watabegg	b3b1188e	
本番環境でこんなコミットしていいんすか		15 Jun 2024 21:...	watabegg	28666901	
ちょっと修正		15 Jun 2024 20:...	watabegg	c96a86c8	
vercelがmainだけデブロイするように変更		15 Jun 2024 19:...	watabegg	f7fd2d14	
Merge pull request #13 from watabegg/develop		15 Jun 2024 19:...	watabegg	f3f40628	
 origin/develop	Merge pull request #12 from watabegg/fix/design	15 Jun 2024 19:...	watabegg	2a10ebe7	
 fix/design	origin いくつかのデザインを修正	15 Jun 2024 19:...	watabegg	040ee687	
Merge pull request #10 from watabegg/develop		13 Jun 2024 18:...	watabegg	ef8cde57	
 develop	木曜日を応急処置的に予約禁止に	13 Jun 2024 18:...	watabegg	c2708893	
prisma変更		13 Jun 2024 16:...	watabegg	e36abcdf	
Merge pull request #9 from watabegg/fix/main-design		13 Jun 2024 16:...	watabegg	3dce1dd6	
 fix/main-design	origin いっぱい変更！	13 Jun 2024 16:...	watabegg	b9d83a20	
予約ログ表示		13 Jun 2024 14:...	watabegg	0b143d9b	
メインページデザインの変更		13 Jun 2024 12:...	watabegg	ecf3298e	
Merge pull request #8 from watabegg/feature/header		13 Jun 2024 11:...	watabegg	74f0001c	
 feature/header	origin ヘッダーを追加	13 Jun 2024 11:...	watabegg	e92eb54d	
prismaを開発用に変更		13 Jun 2024 04:...	watabegg	35dc5350	
prisma変更		13 Jun 2024 04:...	watabegg	124b5936	

さっき日本語化する際に触った四角が四つあるところの検索フォームから「git graph」と検索して一番上に出てきたものをインストールしてください。

出来たら次にTailwind CSS IntelliSenseを入れます。これは最初にデザインのために使うといったtailwindについて、補間や予測してくれる拡張機能です。ちょー便利なので入れましょう。

やることはさっきと同じです。拡張機能のページから「Tailwind CSS IntelliSense」(多分tailwindって入れるだけで出てくる)を検索してインストールです。出来ましたか？素晴らしいです。

ここで一旦環境構築はおしまいになるのでパソコンをシャットダウンでもしてタバコとお酒でも楽しんでほろ酔いで10時間くらい寝ちゃいましょう。

実行編

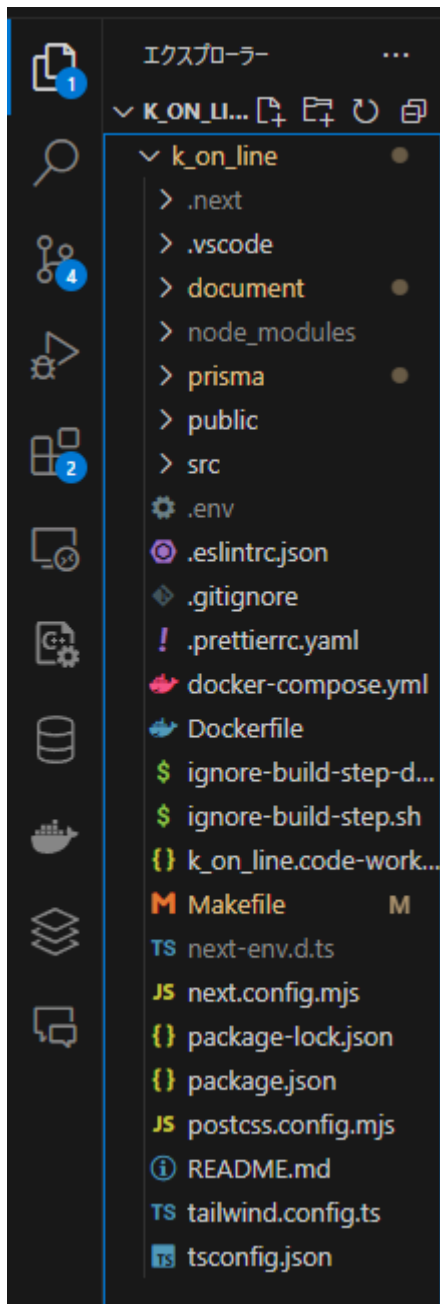
ではここからは実際にコードをローカル環境(あなたのPC上)で実行する方法を教えます。

VS Codeでリポジトリを開く

まずはVS Codeを立ち上げてください。左上にある「ファイル(F)」から「ファイルでワークスペースを開く」をクリックしてください。そうするとエクスプローラーが立ち上がるので「ローカルディスク(C:)」

→「Users」→「(自分のPCのユーザ名)」→「k_on_line」とクリックしていくと「k_on_line.code-workspace」と書かれたファイルが見つかると思います。見つからなかったらわたべに教えて下さい。

見つけたらそこをクリックすることでやっとあしたばコマ表のコード内容が見れるはずです。見れましたか？VS Codeのエクスプローラータブには何やらよくわからない英語の羅列のフォルダや今まで見たことも聞いたこともない拡張子のファイルが見つかるはずです。下の画像と完全一致している必要はありません。



環境変数の設定

ではここから本格的な作業を始めます。まずは上の画像のエクスプローラータブの「tsconfig.json」という名前のファイルの下に何も無い空間を右クリックして「新しいファイル...」をクリックしましょう。こうすることで新しいファイルを生成できます。

まずは名前を入力する必要があるので「.env」と入力してください。なんか歯車アイコンが出てきたうえほかのファイルよりファイル名の色が薄くなれば正解です。意味不明ですね。

と思いましたが、実はここにも中身は書けません！荒らされてしまうので。

送られてきたらそれをコピーして.envファイルの中身にはっつけてください。

A screenshot of the Visual Studio Code editor interface. The top toolbar shows several tabs: "ing.tsx", "TS prisma.ts", ".env" (active), ".env.local", "docker-compose.yml", "schema.prisma M", and "プレビュー". Below the tabs, the ".env" file is open, showing a single line of code: `POSTGRES_PRISMA_URL = postgres://db_XXX`. The cursor is at the end of this line. Line numbers 1, 2, and 3 are visible on the left margin.

これが出来たらCtrl+Sでファイルを保存しましょう！！

いざ実行！

もうあとは一仕事だけです。コマンドプロンプトを開いてください！

今左に書いてある文字が

C:/Users/(Windowsのユーザー名)>

になっていることを確認したら、

```
cd ./k on line
```

を実行してください。そうすると

C:/Users/(Windowsのユーザー名)/k on line>

との表記に変わったと思います。そうしたら

```
npm install
npx prisma generate
npm run dev
```

を一行ずつでいいので実行しましょう。

この際どこかでエラーが出たらわたべにコマンドプロンプトのスクショ付きで相談してください。

おそらく

```
C:\Users\user\k_on_line>npm install

up to date, audited 715 packages in 4s

173 packages are looking for funding
  run `npm fund` for details

7 vulnerabilities (1 moderate, 4 high, 2 critical)

To address issues that do not require attention, run:
  npm audit fix

Some issues need review, and may require choosing
a different dependency.

Run `npm audit` for details.
```

```
C:\Users\user\k_on_line>npx prisma generate
Environment variables loaded from .env
Prisma schema loaded from prisma\schema.prisma

✔Generated Prisma Client (v5.15.0) to .\node_modules\@prisma\client in 49ms

Start using Prisma Client in Node.js (See: https://pris.ly/d/client)
...
import { PrismaClient } from '@prisma/client'
const prisma = new PrismaClient()
...
or start using Prisma Client at the edge (See: https://pris.ly/d/accelerate)
...
import { PrismaClient } from '@prisma/client/edge'
const prisma = new PrismaClient()
...

See other ways of importing Prisma Client: http://pris.ly/d/importing-client
```

Deploying your app to serverless or edge functions?
Try Prisma Accelerate for connection pooling and caching.
<https://pris.ly/cli/--accelerate>

Update available 5.15.0 -> 5.15.1
Run the following to update
npm i --save-dev prisma@latest
npm i @prisma/client@latest

```
C:\Users\user\k_on_line>npm run dev

> k_on_line@0.1.0 dev
> next dev

▲ Next.js 14.2.3
- Local:      http://localhost:3000
- Environments: .env

✓ Starting...
✓ Ready in 3s
|
```

それぞれこんな感じの出力がされるはずですが！実行には少し時間がかかるので気長に待ちましょう。

またコマンドプロンプトは指示があるまで閉じないでください！

これらが完了したら

<http://localhost:3000>

にアクセスしましょう。あなたがいつも使っているブラウザで構いません。すると...

すると...

す、すると...？

ん？

読み込みなが...

あ！すると...

localhost:3000

あしたぼコマ表

カレンダーを更新

使い方の表示



	6月18日(火)	6月19日(水)	6月20日(木)	6月21日(金)	6月22日(土)	6月23日(日)	6月24日(月)	6月25日(火)	6月26日(水)	6月27日(木)	6月28日(金)	6月29日(土)	6月30日(日)	7月1日(月)	7月2日(火)
9:00~10:30	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
10:30~12:00	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
12:00~13:30	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
13:30~15:00	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
15:00~16:30	ザ・モテたいズ わたくし	○	○	○	○	○	○	○	○	○	○	○	○	○	○
16:30~18:00	○	○	×	○	○	○	○	○	○	×	○	○	○	○	○
18:00~19:30	○	○	×	○	○	○	○	○	○	×	○	○	○	○	○
19:30~21:00	○	○	×	○	○	○	○	○	○	×	○	○	○	○	○

きたー！来ました。これであなたのPC上でWebサイトが動くようになりました。やっとですね。お疲れ様です。コーヒーや紅茶などを飲んでこのページを眺めながら優雅な休憩を取りましょう。

これを実行している最中はパソコンがどうしても重たくなってしまいます。そんな時はブラウザの余計なタブを閉じたり、余計なソフトを閉じたりしましょう。くれぐれもコマンドプロンプトだけは閉じないようにしてください。

この状態でさっきのVS Codeに戻り、コードを編集することでこの変更がこちらに反映されるようになります。ここからはあなたがその類まれなるコーディング技術でデザインを作っていくことになります。頑張りましょう。

終了、再起動の方法

では次にこのlocalhostで見える状態を終わらせる方法、もう一度動かす方法をお伝えします。やり方は簡単で、終わらせる際はさっきまで開きっぱなしにしていたコマンドプロンプトを開いて**Ctrl+C**を押すだけです。コマンドプロンプトではCtrl+Cが今動いているタスクのCancelとなります。覚えとくと便利です。

これをすると「バッチジョブを終了しますか？(Y/N)」と出るのでYを入力することで終わることが出来ます。試しに <http://localhost:3000> にもう一度アクセスして見ましょう。何も出なくなりましたよね？これで重すぎ状態になったPCともおさらばです。

この状態になればもう、コマンドプロンプトを閉じて大丈夫です。意味不明のハッカーみたいな画面を見る必要がなくなりましたね！最高！

一旦終わらせちゃったけどもう一回起動させたい！時はさっきも実行した

```
npm run dev
```

を実行しましょう。これだけでまた <http://localhost:3000> にコードの実行結果が出力されます。「え、さっき実行したinstallがどうかprismaがどうかのやつは？」と思うかもしれませんがあれは初期設定です。一度実行してしまえばそれで大丈夫になります。

ただし！たまにわたべが「npm install実行して」とか「npx prisma generate実行頼んだ」とか言うと思うのでそのときはnpm run devを実行する前にやっといってください。

この辺を詳しく知りたい方はこのコマンド名そのまま調べるといっぱい出てくると思います。僕もあんまりよくわかってないので教えてください。

ということで実行も行うことができましたね。ではこのドキュメントはここまでです。こんなにクッソ面倒くさいのによく頑張りました。ここまで実行できた方はもうすでに素晴らしいのでなんかのタイミングでお昼ご飯を奢ります。天才！ありがとう！

では次は「開発のためのドキュメント」でお会いしましょう(絶望)