

Fantstatic

CheatSheet

By Matton

Structure du projet :

```
└─ dev
    └─ components
    └─ imgs
    └─ js
        └─ main
        └─ specific
    └─ pages
        └─ contact
        └─ home
    └─ style
        └─ main.scss
        └─ tailwind.css
```

Installation :

```
git clone https://github.com/monsieurwai/fantstatic.git
cd fantstatic
npm install
```

Arborescence :

- └─ Dev
- └─ components - sous-dossier / composants / HTML et SCSS
- └─ imgs - Images (Création de sous-dossier possible ex: "./imgs/icons/")
- └─ js
- └─ main - contient les scripts qui seront compilés dans un même fichier
- └─ specific - chaque scripts est compilés dans son propre fichier
- └─ pages - Un sous-dossier par pages, avec un fichier HTML et SCSS
- └─ styles - main.scss où l'on importe chaque component.scss
- tailwind.css qui met à notre disposition ses class prédefinis

Commandes :

```
"generate:tailwind": "postcss dev/style/tailwind.css -o dist/style/tailwind.css compressed --si
"minify-imgs": "node compress-images.js",

"watch-html": "html-includes --watch --minify --src dev --dest dist/html",
"watch-scss": "node-sass --watch dev/style/main.scss -o dist/style --output-style compressed",
"watch-js": "npm-run-all --parallel watch-main-js watch-specific-js",

"watch-main-js": "babel --watch dev/js/main --out-file dist/js/main/main.js --source-maps",
"watch-specific-js": "babel --watch dev/js/specific --out-dir dist/js/specific --source-maps",

"dev": "npm-run-all --parallel generate:tailwind watch-scss watch-js watch-html",
"build": "postcss dev/style/tailwind.css -o dist/style/tailwind.css && html-includes --minify

"generate:tailwind": génère le fichier tailwind qui sera ensuite compilé avec seulement les class css utilisées
"minify-imgs": compresse les images du dossier "dev/imgs".
```

les scripts "watch-[nom]" regardent si les fichiers sont modifiés, à chaque sauvegarde d'un fichier modifié il lance la compilation adéquate.

"dev" : lance toute les tâches nécessaires pour développer l'application, c'est cette commande que vous utiliserez la plupart du temps pour développer.

"build" : build l'application dans le dossier "/dist"