

# Étapes que j'ai suivies pour résoudre le problème

## Ajout de l'élément modal

Dans le fichier `LiveDMN.com.html`, j'ai ajouté un élément `<dialog>` nommé `div` avec l'identifiant `"main-modal"`. Cet élément servira plus tard à afficher un tableau de données.

## Création des fichiers nécessaires dans le répertoire frontend

J'ai créé quatre fichiers pour structurer mon code :

- `modal.ts`
- `modal-content.ts`
- `initialData.ts`
- `funcs.ts`

## Description des fichiers et de leur rôle

**modal.ts** : Ce fichier contient une fonction appelée `toggleModal`, qui gère l'ouverture et la fermeture de l'élément `<dialog>` avec l'id `"main-modal"`.

- Si la condition est vraie : le tableau est affiché.
- Sinon, le modal est fermé.

**modal-content.ts** : Ce fichier permet de créer des interfaces (ou composants) réutilisables. Les principales interfaces incluent :

- **tableRender** : Ce composant représente le tableau qui sera affiché dans le modal avec l'id `"main-modal"`.
- **createButtonsWrapper** : Cet élément affiche des boutons pour sélectionner le type de données à afficher dans les graphiques (par exemple : médaille, version, langage de programmation).
- **sideBarKeysDiv** : Comme il n'est pas possible d'afficher du texte directement sur l'axe X des graphiques, ce composant affiche une barre latérale contenant les index et les textes correspondants.

**initialData.ts** : Ce fichier contient les données initiales du dataset.

**funcs.ts** : Ce fichier contient deux fonctions :

- **getCurrentData** : Cette fonction retourne les données du dataset en temps réel.

- `updateSystemData` : Cette fonction modifie les données du dataset après une opération de glisser-déposer (*drag & drop*).

## Transition vers `Dataviz.ts`

Initialement, ce fichier ne contenait qu'une seule fonction appelée : `_Linechart`. J'ai ajouté deux autres fonctions pour enrichir sa fonctionnalité. Voici leur description :

**`select_chart`** : Cette fonction affiche une interface permettant à l'utilisateur de sélectionner le type de graphique à afficher (soit un histogramme (*barchart*), soit un graphique en courbes (*linechart*)).

- Lorsque l'utilisateur clique sur l'un des deux types de graphiques, la fonction correspondante est appelée : soit `_Linechart`, soit `_Histogram`.

**`_Linechart`** : Cette fonction affiche un graphique en courbes basé sur le dataset.

- Avant d'afficher le graphique, trois boutons de sélection sont affichés grâce au composant `createButtonsWrapper` (mentionné précédemment dans `modal-content.ts`).
- Le type d'affichage est un graphique en courbes (*line chart*).

**`_Histogram`** : Cette fonction affiche un histogramme (*barchart*) basé sur le dataset.

- De la même manière que `_Linechart`, elle affiche d'abord trois boutons de sélection via le composant `createButtonsWrapper`.
- Le type d'affichage est un histogramme (*bar chart*).