

IHM – HTML/CSS/TypeScript – exercice 2

Architecture

Se reporter au fichier « **Application HTML_CSS_TS préparation et exécution.pdf** » - **étape 1** pour les explications détaillées.

Sur votre « devhom », emplacement où se trouve vos applications web sur le serveur « devweb » de l'IUT de Metz, dans le dossier créé ou à créer IHM,

- a) créer le dossier TP2
- b) copier le contenu du dossier « squelette » (fichier compressé « squelette » à télécharger sur Arche), dans le dossier TP2
 - le fichier « tsconfig.json » et les dossiers « vue », « src/contrôleur » et « src/modele » apparaissent
 - le TP2 n'utilise pas de données extraites d'une base de données → supprimer le dossier « modele »

Travail à faire

Ecrire le fichier « **tp2.html** » dans le **dossier « vue »** et le fichier **tp2.css** associé dans le **sous-dossier « css »** du dossier « vue » à partir de la maquette ci-contre.

La maquette présente une interface web simple. En haut, un titre « Mise en boîte de petits textes » est suivi d'une instruction « Taper votre texte puis Entrer ». En dessous se trouve un champ de saisie unique avec le placeholder « entrer le texte ici ». À droite du champ, deux boutons rectangulaires sont empilés : « Supprimer » (en haut) et « Vider » (en bas). En bas à gauche de la zone de saisie, il y a une case à cocher non cochée avec l'étiquette « liste triée ».

Conception HTML/CSS

Se reporter au fichier « **Application HTML_CSS_TS préparation et exécution.pdf** » - **étape 2**

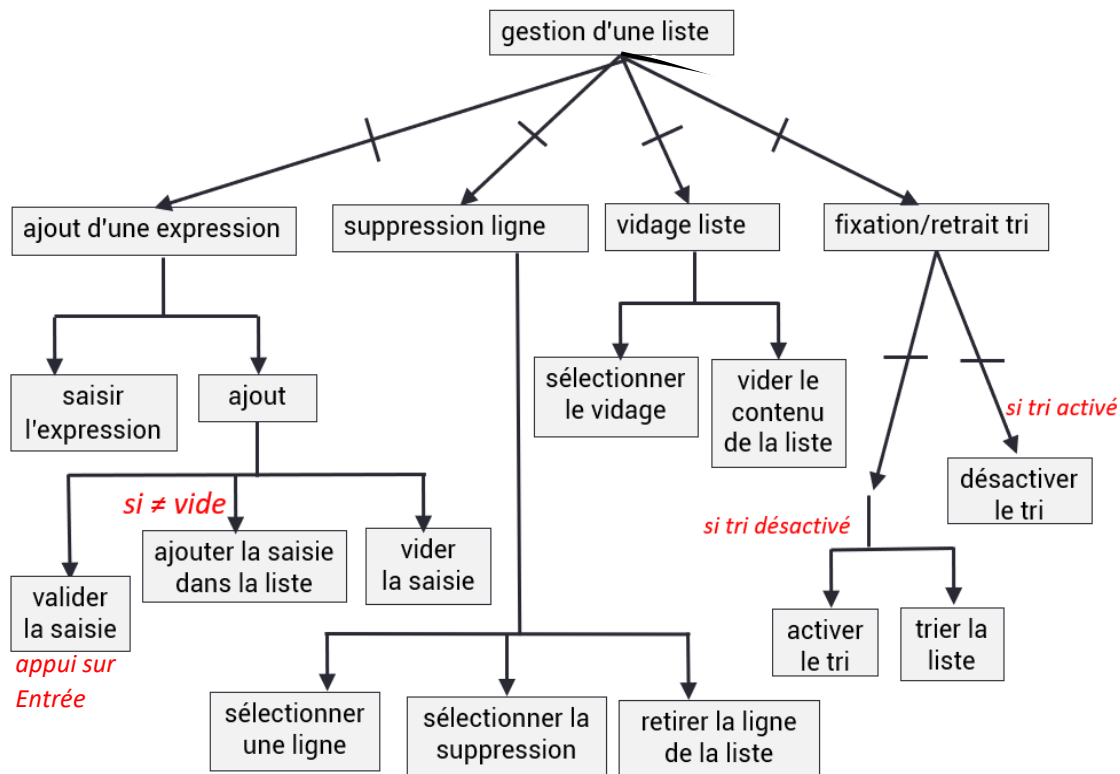
Créer le fichier « vue/tp2.html »

- 1) Définir la structure globale du fichier « tp2.html » avec les parties
 - <head> avec le lien vers le fichier « tp2.css »
 - <body>
 - <script> avec l'appel du fichier « tp2.js »
- 2) Définir une zone identifiée « div_tp2 » dans <body>
- 3) Est-ce utile que la zone occupe entièrement l'espace écran ?
- 4) Définir le style associé à la zone ?
- 5) Déterminer le nombre de zones horizontales y compris le titre et définir les zones sans définir leur contenu.
- 6) Déterminer pour chaque zone,
 - si un découpage en colonnes est nécessaire ?
 - définir le contenu et ajouter les styles nécessaires

IHM – HTML/CSS/TypeScript – exercice 2

Travail à faire

Ecrire les fichiers « **class_tp2.ts** » et « **tp2.ts** » dans le dossier « **src/controleur** » à partir du modèle des tâches



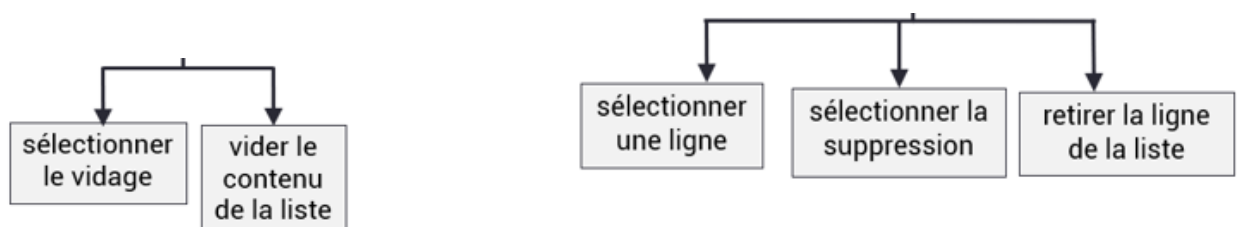
Programmation TypeScript

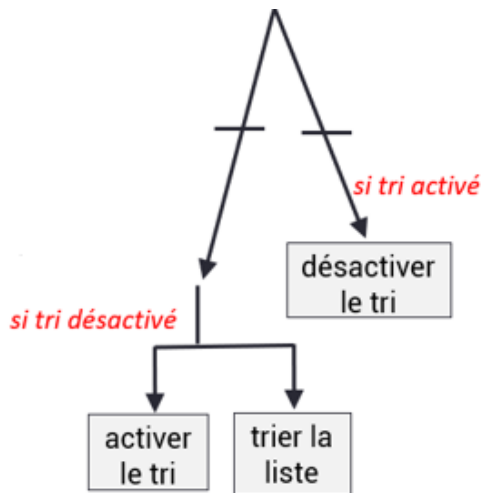
Se reporter au fichier « **Application HTML_CSS_TS préparation et exécution.pdf** » - **étape 3**

Créer le fichier « src/controleur/class_tp2.ts »

- 1) Définir le type « TTp2Form »
- 2) Créer la classe « VueTp2 » et définir l'attribut privé « `_form` », la fonction « `init` », le getter associé à l'attribut privé
- 3) Quels sont les événements/actions déclenchés par l'utilisateur ?

A partir du modèle des tâches





coup de pouce

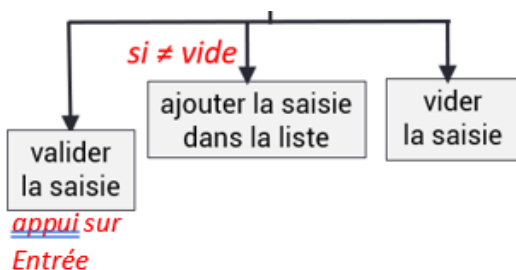
événement « **change** » de la case à cocher « **tri** »
 → appel de la méthode **trierListe()**

coup de pouce

code complet avec tri d'une liste en typescript

```

triListe(liste : HTMLSelectElement):void {
  const options : HTMLOptionsCollection = liste.options;
  let optionsArray : HTMLOptionElement[] = [];
  for (let i = 0; i < options.length; i++) {
    optionsArray.push(options[i]);
  }
  optionsArray = optionsArray.sort(function
    (a:HTMLOptionElement, b:HTMLOptionElement):number {
      if (a.value > b.value) { return 1 } else { return -1 }
    });
  for (let i = 0; i <= options.length; i++) {
    options[i] = optionsArray[i];
  }
}
  
```



coup de pouce

événement « **keydown** » de la zone de saisie
 → appel de la méthode **ajouterSaisie(ev:KeyboardEvent)**

IHM – HTML/CSS/TypeScript – exercice 2

- 4) Définir les méthodes de la classe `viderListe()`, `supprimerLigne()`, `trierListe()`, `ajouterSaisie()`

```
coup de pouce : début méthode « ajouterSaisie »  
if (ev.key === 'Enter') {
```

- 5) Créer l'objet « vueTp2 », instantiation de la classe
- 6) Écrire la commande d'exportation de l'objet
- 7) Écrire le corps des méthodes de la classe
- 8) Ajouter la définition des événements dans la fonction « init »
- « keydown » sur la zone de saisie du texte,
 - « change » sur la case à cocher
 - « click » sur les boutons « vider », « supprimer »

Créer le fichier « src/controleur/tp2.ts »

- 1) Écrire la commande d'importation de l'objet créé précédemment
- 2) Appeler la méthode « init » avec les paramètres

Transpiler vos fichiers « ts » et tester l'application

Se reporter au fichier « **Application HTML_CSS_TS préparation et exécution.pdf** » - **étape 4**