

# CORRIGE - exercice 3 - HTML\_CSS\_TypeScript

## 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 crée ou à créer IHM,

- créer le dossier TP3
- copier le contenu du dossier « squelette » (fichier compressé « squelette » à télécharger sur Arche), dans le dossier TP3
  - le fichier « tsconfig.json » et les dossiers « vue », « src/contrôleur » et « src/modele » apparaissent
  - le TP3 n'utilise pas de données extraites d'une base de données → supprimer le dossier « modele »

## Travail à faire

- écrire le fichier « **tp3.html** » dans le **dossier « vue »** et le fichier **tp3.css** associé dans le **sous-dossier « css »** du dossier « vue » à partir de la maquette ci-contre.

## Conception HTML/CSS

Se reporter au fichier « **Application HTML\_CSS\_TS préparation et exécution.pdf** » - **étape 2**

### Créer le fichier « vue/tp3.html »

- Définir la structure globale du fichier « tp3.html » avec les parties
  - <head> avec le lien vers le fichier « tp3.css »
  - <body>
  - <script> avec l'appel du fichier « tp3.js »

```
<!DOCTYPE html>
<html>
  <head>
    <meta http-equiv="content-type" content="text/html; charset=UTF-8">
    <title> Du repos à l'astreinte </title>
    <link type="text/css" rel="stylesheet"
          href= "../css/tp3.css">
  </head>
  <body>
  </body>
  <script type="module" src= "../contrôleur/tp3.js">
  </script>
</html>
```

## CORRIGE - exercice 3 - HTML\_CSS\_TypeScript

2) Définir une zone identifiée « div\_tp3 » dans <body>

```
<!DOCTYPE html>
<html>
  <head>
    <meta http-equiv="content-type" content="text/html; charset=UTF-8">
    <title> Du repos à l'astreinte </title>
    <link type="text/css" rel="stylesheet"
          href= "../css/tp3.css">
  </head>
  <body>
    <div id='div_tp3'>
    </div>
  </body>
  <script type="module" src = "../controleur/tp3.js">
  </script>
</html>
```

3) Est-ce utile que la zone occupe entièrement l'espace écran ? **pas nécessairement, on choisit tout de même d'occuper l'intégralité de l'espace écran.**

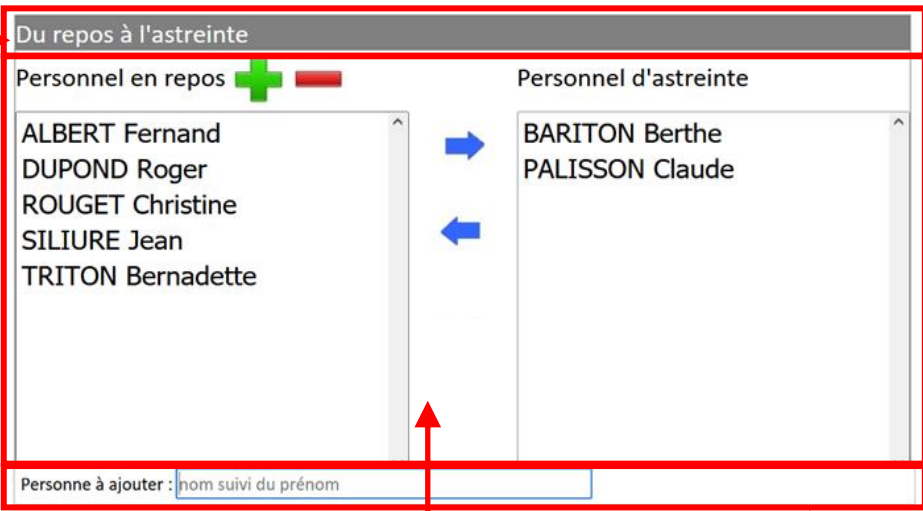
4) Définir le style associé à la zone ?

*tp3.css*

```
#div_tp3 {
  width      : 100%;
  height     : 100%;
  border     : 1px solid #808080;
  font-size  : 1.5rem;
}
```

## CORRIGE - exercice 3 - HTML\_CSS\_TypeScript

- 5) Déterminer le nombre de zones horizontales y compris le titre et définir les zones sans définir leur contenu.



```
<div id='div_tp3'>
  <header>
  </header>
  <div id="div_listes" class="div_listes">
  </div>
  <div id="div_saisie" class="div_saisie">
  </div>
</div>
```

- 6) Déterminer pour chaque zone,
- si un découpage en colonnes est nécessaire ?
  - définir le contenu et ajouter les styles nécessaires

6.1)header: **NON**

```
<header>
Du repos à l'astreinte
</header>
```

tp3.css

```
header {
  background-color : #808080;
  color           : #ffffff;
}
```

## CORRIGE - exercice 3 - HTML\_CSS\_TypeScript

6.2)div\_listes: **OUI (3 colonnes)**

```
<div class='div_listes'>
  <div>
    <div class='div_titre'>
      Personnel en repos
      <input id="btn_ajouter" type="button">
      <input id="btn_supprimer" type="button">
    </div>
    <select id="select_repos" class="liste" size="10">
    </select>
  </div>
  <div class='div_boutons'>
    <input id="btn_vers_astreinte" type="button">
    <br><br>
    <input id="btn_vers_repos" type="button">
  </div>
  <div>
    <div class='div_titre'>
      Personnel d'astreinte
    </div>
    <select id="select_astreinte"
      class="liste" size="10">
    </select>
  </div>
</div>
```



tp3.css

```
.liste {
  width : 100%;
}
.div_titre {
  line-height      : 3rem;
}
.div_listes {
  display          : grid;
  grid-template-columns : 4fr 6rem 4fr;
}
.div_boutons {
  text-align       : center;
  margin-top       : 4rem;
}
select {
  font-size        : 1.5rem;
}
input[type=button] {
  cursor           : pointer;
  border           : none;
  vertical-align    : middle;
  width            : 3rem;
  height           : 3rem;
}
```

## CORRIGE - exercice 3 - HTML\_CSS\_TypeScript

```
#btn_vers_astreinte {  
background:url(vers_astreinte.jpg) no-repeat center center; }  
#btn_vers_repos {  
background:url(vers_repos.jpg) no-repeat center center; }  
#btn_ajouter {  
    background:url(ajouter.jpg) no-repeat center center; }  
#btn_supprimer {  
    background:url(supprimer.jpg) no-repeat center center; }
```

## CORRIGE - exercice 3 - HTML\_CSS\_TypeScript

6.3)div\_saisie: **NON**

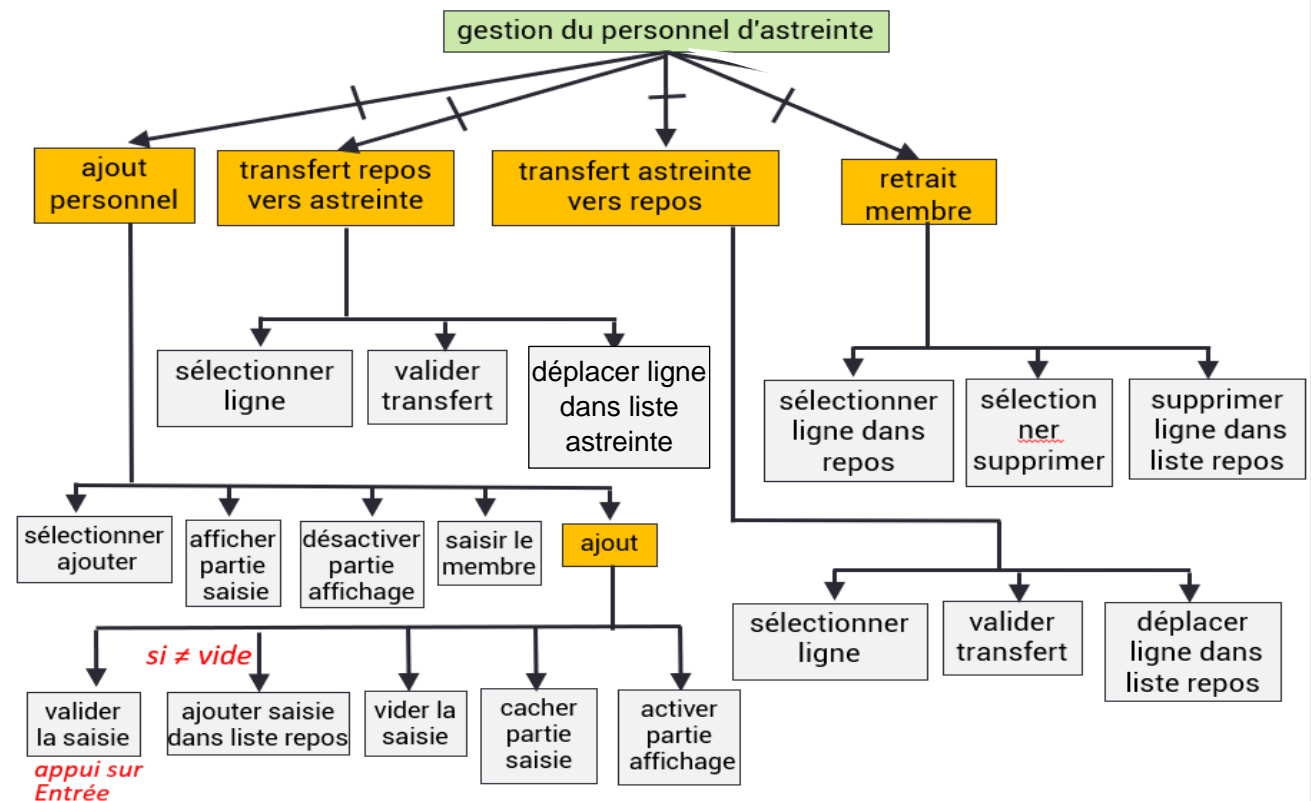
Personne à ajouter :

```
<div id='div_saisie'>
  <label for="edt_saisie">Personne à ajouter :</label>
  <input type='text' id="edt_saisie" name="edt_saisie"
    placeholder="nom suivi du prénom" size='40'>
</div>
```

tp3.css

```
input {
  border      : 1px solid #d0d0d0;
  color       : #000000;
  font        : inherit;
  font-size   : 1.5rem;
  text-align  : left;
  height      : 2rem;
}
input:hover {
  border      : 2px solid #0000ff;
}
```

# CORRIGE - exercice 3 - HTML\_CSS\_TypeScript



# CORRIGE - exercice 3 - HTML\_CSS\_TypeScript

## Programmation TypeScript

Se reporter au fichier « **Application HTML\_CSS\_TS préparation et exécution.pdf** » - **étape 3**

**Créer le fichier « src/controleur/class\_tp3.ts »**

1) Définir le type « TTp3Form »

```
type TTp3Form = {  
  divListes : HTMLElement  
  , listeRepos: HTMLSelectElement  
  , listeAstreinte: HTMLSelectElement  
  , btnAjouter: HTMLInputElement  
  , btnSupprimer: HTMLInputElement  
  , btnVersAstreinte: HTMLInputElement  
  , btnVersRepos: HTMLInputElement  
  , divSaisie : HTMLElement  
  , edtSaisie : HTMLInputElement  
}
```

2) Créer la classe « VueTp3 » et définir l'attribut privé « \_form », la fonction « init », le getter associé à l'attribut privé

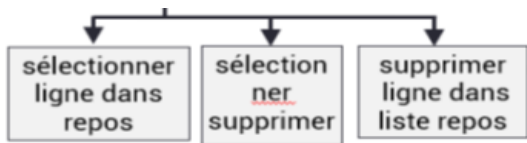
```
class VueTp3 {  
  private _form : TTp3Form  
  
  init(form : TTp3Form) : void {  
    this._form = form;  
    this.form.divSaisie.hidden = true;  
  }  
  
  get form() : TTp3Form { return this._form }  
}
```



## CORRIGE - exercice 3 - HTML\_CSS\_TypeScript

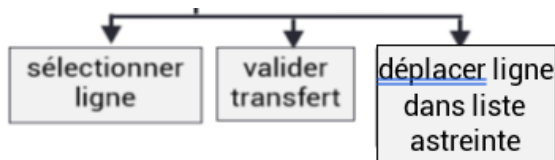
3) Quels sont les événements/actions déclenchés par l'utilisateur ?

A partir du modèle des tâches



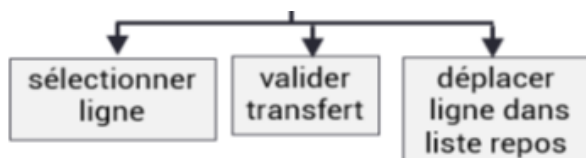
événement « clic » du bouton « **supprimer** »

→ appel de la méthode **supprimerLigne()**



événement « clic » du bouton « **vers astreinte** »

→ appel de la méthode **versAstreinte()**



événement « clic » du bouton « **vers repos** »

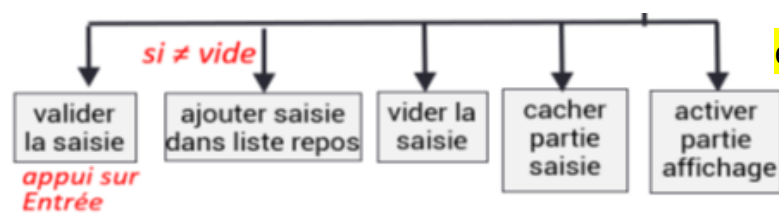
→ appel de la méthode **versRepos()**

Le transfert « repos » vers « astreinte » et « astreinte » vers « repos » appelle le même traitement : le déplacement de la ligne sélectionnée d'une liste origine vers une liste destination.



événement « clic » du bouton « **ajouter** »

→ appel de la méthode **afficherSaisie()**



événement « **keypress** » de la zone de saisie

→ appel de la méthode **ajouterSaisie(ev:KeyboardEvent)**

## CORRIGE - exercice 3 - HTML\_CSS\_TypeScript

4) Définir les méthodes de la classe `afficherSaisie()`, `ajouterSaisie()`, `supprimerLigne()`, `versAstreinte()`, `versrepos()`

`liste1VersListe2(liste1 : HTMLSelectElement, liste2 : HTMLSelectElement)` appelée par les méthodes « `versAtreinte` » et « `versRepos` »

```
class VueTp3 {  
  
    • • •  
  
    afficherSaisie() :void {  
        // à compléter  
    }  
  
    supprimerLigne() :void {  
        // à compléter  
    }  
  
    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];  
        }  
    }  
  
    ajouterSaisie(ev:KeyboardEvent) :void {  
        // à compléter  
    }  
  
    liste1VersListe2(liste1 :HTMLSelectElement  
        , liste2 : HTMLSelectElement) :void {  
        // à compléter  
    }  
  
    versAstreinte() :void {  
        // à compléter  
    }  
  
    versRepos() :void {  
        // à compléter  
    }  
}  
  
let vueTp3 = new VueTp3;  
  
export { vueTp3 }
```

5) Créer l'objet « `vueTp3` », instantiation de la classe

6) Écrire la commande d'exportation de l'objet

## CORRIGE - exercice 3 - HTML\_CSS\_TypeScript

7) Écrire le corps des méthodes de la classe

```
afficherSaisie() : void {
    this.form.divListes.style.pointerEvents = 'none';
    this.form.divSaisie.hidden = false;
    this.form.edtSaisie.focus();
}

supprimerLigne():void {
    const noLigne : number = this.form.listeRepos.selectedIndex;
    if (noLigne > -1) {
        this.form.listeRepos.remove(noLigne);
    }
}

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 {
        return Number(a.value > b.value);
    });
    // tri dans l'ordre croissant : retourne 1 si vrai et 0 si faux
    for (let i = 0; i <= options.length; i++) {
        options[i] = optionsArray[i];
    }
}

ajouterSaisie(ev:KeyboardEvent):void {
    if (ev.key === 'Enter') {
        const elt = this.form.edtSaisie;
        const liste = this.form.listeRepos;
        const chaine : string = elt.value.trim();
        if (chaine !== "") {
            const opt = new Option(chaine, chaine);
            liste.options.add(opt);
            this.triListe(liste);
        }
        elt.value = "";
        this.form.divListes.style.pointerEvents = 'auto';
        this.form.divSaisie.hidden = true;
    }
}
```

## CORRIGE - exercice 3 - HTML\_CSS\_TypeScript

```
liste1VersListe2 (liste1 : HTMLSelectElement, liste2 :
HTMLSelectElement) :void {
    const noLigne : number = liste1.selectedIndex;
    if (noLigne > -1) {
        liste2.options.add(liste1.options[noLigne]);
        this.triListe(liste2);
    }
}

versAstreinte():void {
    this.liste1VersListe2(this.form.listeRepos,
    this.form.listeAstreinte);
}

versRepos():void {
    this.liste1VersListe2(this.form.listeAstreinte,
    this.form.listeRepos);
}
```

8) Ajouter la définition des événements dans la fonction « init »

```
init(form : TTp3Form) : void {
    ...

    this.form.btnAjouter.onclick
        = function ():void      { vueTp3.afficherSaisie();    }
    this.form.btnSupprimer.onclick
        = function ():void      { vueTp3.supprimerLigne();    }
    this.form.btnVersAstreinte.onclick
        = function ():void      { vueTp3.versAstreinte();    }
    this.form.btnVersRepos.onclick
        = function ():void      { vueTp3.versRepos();        }
    this.form.edtSaisie.onkeydown
        = function (event):void { vueTp3.ajouterSaisie(event); }
}
```

# CORRIGE - exercice 3 - HTML\_CSS\_TypeScript

## Créer le fichier « src/controleur/tp3.ts »

1) Écrire la commande d'importation de l'objet créé précédemment

```
import {vueTp3} from "../controleur/class_tp3"
```

2) Appeler la méthode « init » avec les paramètres

```
vueTp3.init (  
{  
  divListes      : document.querySelector('[id=div_listes]')  
  , listeRepos   : document.querySelector('[id=select_repos]')  
  , listeAstreinte: document.querySelector('[id=select_astreinte]')  
  , btnAjouter   : document.querySelector('[id=btn_ajouter]')  
  , btnSupprimer : document.querySelector('[id=btn_supprimer]')  
  , btnVersAstreinte : document.querySelector('[id=btn_vers_astreinte]')  
  , btnVersRepos : document.querySelector('[id=btn_vers_repos]')  
  , divSaisie    : document.querySelector('[id=div_saisie]')  
  , edtSaisie    : document.querySelector('[id=edt_saisie]')  
} );
```

## Transpiler vos fichiers « ts » et tester l'application

Se reporter au fichier « Application HTML\_CSS\_TS préparation et exécution.pdf » - **étape 4**