

CORRIGE - exercice 1 - 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,

- a) créer le dossier TP1
- b) copier le contenu du dossier « squelette » (fichier compressé « squelette » à télécharger sur Arche), dans le dossier TP1
 - le fichier « tsconfig.json » et les dossiers « vue », « src/contrôleur » et « src/modele » apparaissent
 - le TP1 n'utilise pas de données extraites d'une base de données → supprimer le dossier « modele »

Travail à faire

Ecrire le fichier « **tp1.html** » dans le dossier « **vue** » et le fichier **tp1.css** associé dans le sous-dossier « **css** » du dossier « vue » à partir de la maquette ci-contre.

Hors Taxes ou Toutes Taxes Comprises		
Taux de TVA		
<input type="radio"/> 5%	<input type="radio"/> 10%	<input type="radio"/> 20%
HT	<input type="text" value="entrer le HT"/>	
TTC	<input type="text" value="entrer le TTC"/>	



Conception HTML/CSS

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

Créer le fichier « vue/tp1.html »

- 1) Définir la structure globale du fichier « tp1.html » avec les parties
 - <head> avec le lien vers le fichier « tp1.css »
 - <body>
 - <script> avec l'appel du fichier « tp1.js »

```
<!DOCTYPE html>
<html>
  <head>
    <meta http-equiv="content-type" content="text/html; charset=UTF-8">
    <title>Hors Taxes ou Toutes Taxes Comprises</title>
    <link type="text/css" rel="stylesheet" href="../css/tp1.css">
  </head>
  <body>
  </body>
  <script type="module" src="../contrôleur/tp1.js">
  </script>
</html>
```

CORRIGE - exercice 1 - HTML_CSS_TypeScript

2) Définir une zone identifiée « div_tp1 » dans <body>

```
<!DOCTYPE html>
<html>
  <head>
    <meta http-equiv="content-type" content="text/html; charset=UTF-8">
    <title>Hors Taxes ou Toutes Taxes Comprises</title>
    <link type="text/css" rel="stylesheet" href= " ./css/tp1.css">
  </head>
  <body>
    <div id='div_tp1'>
    </div>
  </body>
  <script type="module" src="../controleur/tp1.js">
  </script>
</html>
```

3) Est-ce utile que la zone occupe entièrement l'espace écran ? **NON**

4) Définir le style associé à la zone ? *tp1.css*

```
#div_tp1 {
  width      : 400px;
  height     : 400px;
  border     : 1px solid #808080;
  font-size  : 1.5rem;
}
```

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

```
<div id='div_tp1'>
  <header>
  </header>
  <div class='div_tva'>
  </div>
  <div class='div_saisie'>
  </div>
  <div class='div_bouton'>
  </div>
</div>
```

The diagram illustrates the visual layout of the form, with red boxes and arrows mapping the HTML elements from the code to the corresponding UI components:

- Header:** A grey box containing the text "Hors Taxes ou Toutes Taxes Comprises".
- div_tva:** A white box containing the text "Taux de TVA" and three radio buttons labeled "5%", "10%", and "20%".
- div_saisie:** A white box containing two input fields. The first is labeled "HT" and contains the text "entrer le HT". The second is labeled "TTC" and contains the text "entrer le TTC".
- div_bouton:** A white box containing a calculator icon.

CORRIGE - exercice 1 - HTML_CSS_TypeScript

- 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>
Hors Taxes ou Toutes Taxes Comprises
</header>
```

6.2) *div_tva* : **NON pour la première ligne du titre**
, OUI pour les boutons radios (3 colonnes égales)

Taux de TVA		
<input type="radio"/> 5%	<input type="radio"/> 10%	<input type="radio"/> 20%

```
<div class='div_tva'>
  Taux de TVA<br>
  <div class='div_tvadetail'>
    <div><input type='radio' id='radio_taux1'
      name='taux_tva' value='5' >5%</div>
    <div><input type='radio' id='radio_taux2'
      name='taux_tva' value='10'>10%</div>
    <div><input type='radio' id='radio_taux3'
      name='taux_tva' value='20'>20%</div>
  </div>
</div>
```

```
tp1.css
.div_tva {
  border          : 1px solid black;
  padding         : 0.25rem;
}
.div_tvadetail {
  display         : grid;
  grid-template-columns : 1fr 1fr 1fr;
  grid-gap       : 1rem;
}
input[type=radio] {
  cursor         : pointer;
}
```

CORRIGE - exercice 1 - HTML_CSS_TypeScript

6.3) *div_saisie* : **OUI (2 colonnes égales avec une marge à droite)**

HT	entrer le HT
TTC	entrer le TTC

```
<div class='div_saisie'>
  <label for='edt_ht' >HT</label>
  <input type='text' name='ht' id='edt_ht'
    placeholder="entrer le HT" size='10' maxlength='10'>
  <label for='edt_ttc' >TTC</label>
  <input type='text' name='edt_ttc' id='edt_ttc'
    placeholder='entrer le TTC' size='10' maxlength='10'>
</div>
```

tp1.css

```
.div_saisie {
  display          : grid;
  grid-template-columns : 1fr 1fr;
  padding-right     : 6rem;
  grid-gap         : 1rem;
}
label {
  text-align       : right;
}
input {
  border           : 1px solid #5b5b5b;
  color            : #000000;
  font             : inherit;
  text-align       : left;
}
```

CORRIGE - exercice 1 - HTML_CSS_TypeScript

6.4) *div_bouton* : **NON**

```
<div class='div_bouton'>  
  <input id="btn_calculer" type="button" >  
</div>
```



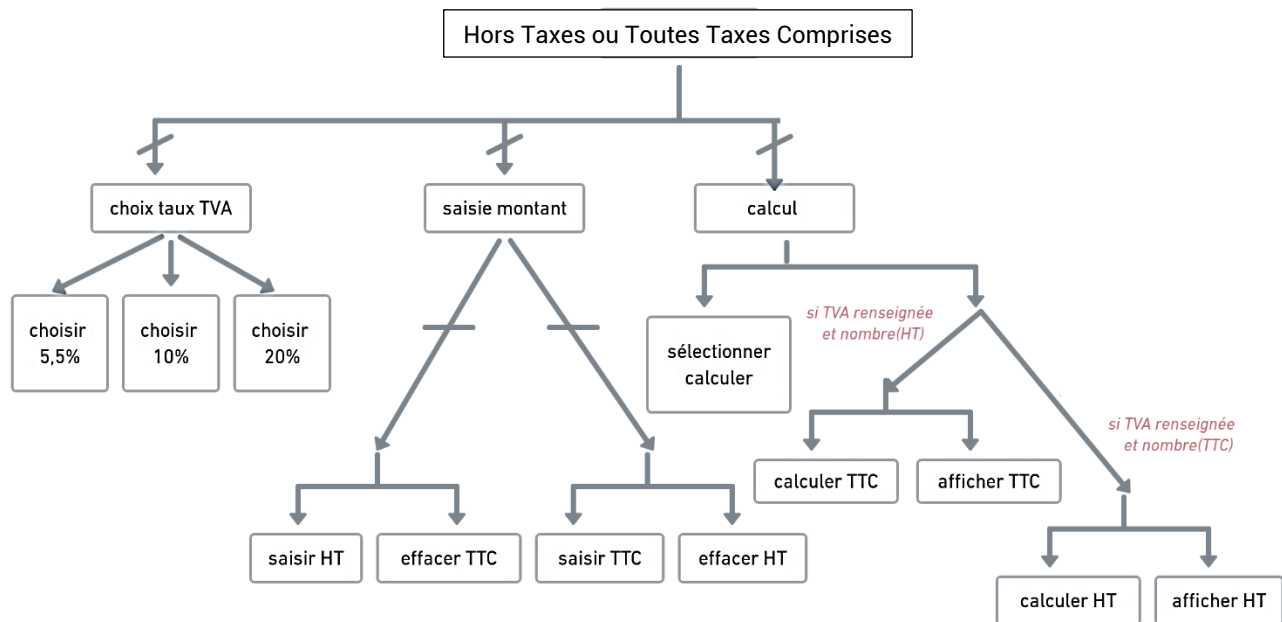
tp1.css

```
.div_bouton {  
  float : right;  
  padding-right : 1rem;  
}  
input[type=button] {  
  cursor : pointer;  
  border : none;  
  vertical-align : middle;  
  width : 3rem;  
  height : 3rem;  
}  
input[type=button]:hover {  
  border : 2px solid #0000ff;  
}  
#btn_calculer {  
  background:url(calculer.jpg)  
  no-repeat center center;  
}
```

CORRIGE - exercice 1 - HTML_CSS_TypeScript

Travail à faire

Ecrire les fichiers « **class_tp1.ts** » et « **tp1.ts** » dans le dossier « **src/controleur** » à partir du modèle des tâches



CORRIGE - exercice 1 - 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_tp1.ts »

1) Définir le type « TTp1Form »

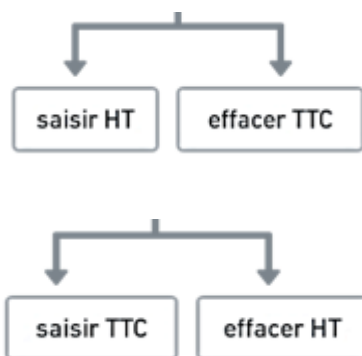
```
type TTp1Form = {  
    edtHt      : HTMLInputElement  
    , edtTtc    : HTMLInputElement  
    , radioTaux1 : HTMLInputElement  
    , radioTaux2 : HTMLInputElement  
    , radioTaux3 : HTMLInputElement  
    , btnCalculer: HTMLInputElement  
}
```

2) Créer la classe « VueTp1 » et définir l'attribut privé « _form », la fonction « init », le getter associé à l'attribut privé

```
class VueTp1 {  
    private _form : TTp1Form  
  
    init(form : TTp1Form) : void {  
        this._form = form  
    }  
  
    get form() : TTp1Form { return this._form }  
}
```

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

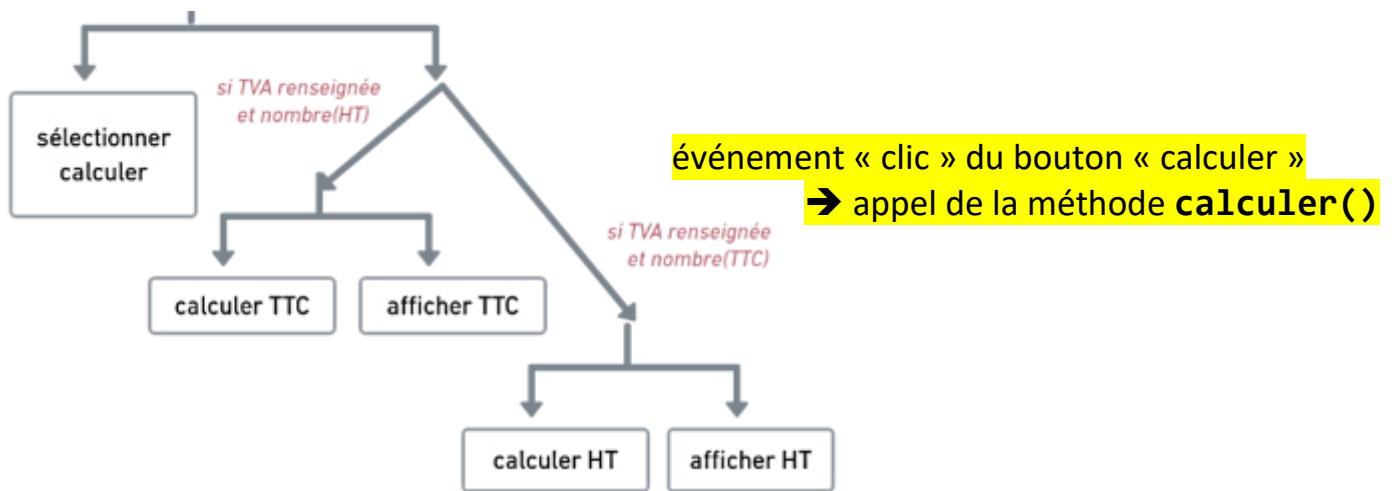
A partir du modèle des tâches



événement « keypress » de la zone de saisie HT
→ appel de la méthode **viderTtc()**

événement « keypress » de la zone de saisie TTC
→ appel de la méthode **viderHt()**

CORRIGE - exercice 1 - HTML_CSS_TypeScript



4) Définir les méthodes de la classe **viderHT()**, **viderTTC()**, **calculer()**

```
class VueTp1 {  
  
    . . .  
  
    viderHT() :void {  
        // à compléter  
    }  
    viderTTC() :void {  
        // à compléter  
    }  
    calculer() :void {  
        // à compléter  
    }  
}
```

```
let vueTp1 = new VueTp1;
```

```
export { vueTp1 }
```

5) Créer l'objet « vueTp1 », instantiation de la classe

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

CORRIGE - exercice 1 - HTML_CSS_TypeScript

7) Écrire le corps des méthodes de la classe

```
viderTtc():void {
    this.form.edtTtc.value = '';
}

viderHt():void {
    this.form.edtHt.value = '';
}

calculer():void {
    const taux1 = this.form.radioTaux1;
    const taux2 = this.form.radioTaux2;
    const taux3 = this.form.radioTaux3;
    const edtHt = this.form.edtHt;
    const edtTtc = this.form.edtTtc;

    let taux = '';
    if (taux1.checked) { taux = taux1.value }
    else if (taux2.checked) { taux = taux2.value }
    else if (taux3.checked) { taux = taux3.value }

    if (taux !== '') {
        const tauxTva = Number(taux)/100;
        const ht = edtHt.value.trim();
        if (ht !== '') {
            const nb = Number(ht);
            if (isNaN(nb)) {
                alert("Erreur - calcul impossible : le montant HT " + ht
                    + " est invalide.")
            }
            else { edtTtc.value = this.calculHtVersTtc(nb, tauxTva).toFixed(2); }
        }
        else {
            const ttc = edtTtc.value.trim();
            if (ttc !== '') {
                const nb = Number(ttc);
                if (isNaN(nb)) {
                    alert("Erreur - calcul impossible : le montant TTC " + ttc
                        + " est invalide.")
                }
                else {
                    edtHt.value = this.calculTtcVersHt(nb, tauxTva).toFixed(2);
                }
            }
        }
    }
    else { alert("Erreur - calcul impossible : le Taux de TVA n'est pas renseigné.") }
}

calculHtVersTtc(ht:number, taux:number) : number {
    return ht * (1+taux)
}

calculTtcVersHt(ttc:number, taux:number) : number {
    return ttc / (1+taux)
}
```

CORRIGE - exercice 1 - HTML_CSS_TypeScript

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

```
init(form : TTp1Form) : void {  
    ...  
    // définition des événements  
    this.form.edtHt.onkeydown = function():void { vueTp1.viderTtc(); }  
    this.form.edtTtc.onkeydown = function():void { vueTp1.viderHt(); }  
    this.form.btnCalculer.onclick = function():void { vueTp1.calculer(); }  
}
```

Créer le fichier « src/controleur/tp1.ts »

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

```
import {vueTp1} from "../controleur/class_tp1"
```

2) Appeler la méthode « init » avec les paramètres

```
vueTp1.init (  
{   edtHt      :document.querySelector('[id=edt_ht]')  
  , edtTtc     :document.querySelector('[id=edt_ttc]')  
  , radioTaux1 :document.querySelector('[id=radio_taux1]')  
  , radioTaux2 :document.querySelector('[id=radio_taux2]')  
  , radioTaux3 :document.querySelector('[id=radio_taux3]')  
  , btnCalculer:document.querySelector('[id=btn_calculer]')  
} );
```

Transpiler vos fichiers « ts » et tester l'application

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