WEB TECHNOLOGIES 2022- 2023

Bases de données Web et Web Sémantique (I-ILIA-113) - TP

TP 2.1: Encodage de factures avec HTML, CSS et Javascript

Durant ce TP, nous allons utiliser le langage Javascript afin de vérifier la cohérence des champs de notre formulaire d'encodage de facture avant de l'envoyer.

- 1. Télécharger depuis Moodle le code démarrage « TP2_Exercice.zip » incluant les fichiers suivants : « facture.html », « style.css » et « code.js ». La figure 1 illustre le visuel attendu.
- 2. Analyser le code, tester un envoi avec « envoyer » sans saisir le nom. Que constatez-vous ?

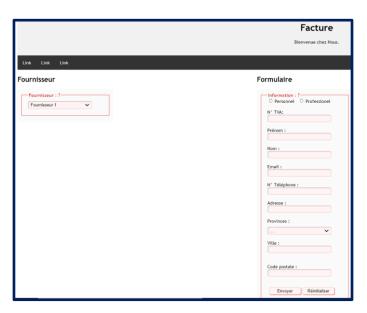


Figure 1: formulaire d'encodage de facture (HTML + CSS)

3. Vérifier la présence du lien entre le fichier « **facture.html** » et le fichier javascript « **code.js** » afin d'appeler les fonctions javascript.

- Question 1 : encoder l'appel de la fonction « mfValidation » depuis le formulaire afin de récupérer les champs obligatoires et vérifier leur présence avant l'envoi du formulaire. Par exemple, si l'utilisateur oublie d'encoder le nom, la fonction enverra un message d'erreur ;
 - Pour appeler cette fonction depuis le fichier HTML, il faut déclencher l'évènement **onsubmit** lors de la définition du formulaire
 - Dans le fichier JavaScript « code.js », il faut utiliser la fonction mfValidation

```
<form action="#" method="post" onsubmit="return mfValidation(this)">
```

Note: regex1, regex1, regex3: expressions régulières pour vérifier le format du numéro de tél, code postal, le CVV.

TP2.1: Javascript Sidi Ahmed Mahmoudi

WEB TECHNOLOGIES 2022- 2023

Question 2: adapter la fonction pour imposer que tous les champs soient obligatoires (*) pour l'envoi du formulaire (sauf le N° TVA qui n'est pas obligatoire).

- Question 3 : créer une fonction qui permet de supprimer ou ajouter le numéro de TVA en fonction du type d'utilisateur (Personnel ou professionnel).
 - 3.1. Définir l'évènement onclick pour le type radio (personnel) en indiquant la fonction « personnel ()» qu'il faudra définir dans le fichier code.js :

```
<input type="radio" name="type" value="perso" onclick="personnel();"> Personnel&nbsp;&nbsp;
```

```
function personnel()
{
          document.getElementById("tva").innerHTML = "";
          document.getElementById("gender").innerHTML = "<input type='radio'
name='gender' value='male' checked> Male<br><input type='radio'
name='gender' value='female'> Female<br><input type='radio' name='gender'
value='other'> Other<br><br><"; }</pre>
```

3.2. Ajouter une fonction « **professionnel()** » liée aux utilisateurs professionnels disposant d'un numéro de TVA (le sexe « **gender** » n'est pas nécessaire dans ce cas)

<u>Note</u>: la ligne (document.getElementById("tva").innerHTML="") permet de supprimer le champs TVA qui sera remplacé par une ligne vide.

- Question 4 : implémenter la fonction permettant de choisir la ville en fonction de province en créant une fonction « ville » qui récupère le nom de province et propose les villes situées dans cette province:
 - 4.1. Analyser la liste déroulante (HTML) de choix la région (Hainaut, Brabant wallon, Liège, Namur)
 - **4.2.** Intégrer la fonction Javascript « **ville** » dans le fichier « **code.js** » et créer le lien entre la balise intitulée « **region** » avec cette fonction de choix de ville dynamique.

```
function ville()
{
    var e = document.getElementById("region"); var
    region = e.options[e.selectedIndex].value;
    if(region=="f2")
    {
        document.getElementById("ville").innerHTML = "<select name='ville2'
    id='ville2'> <option>Mons</option><option>Charleroi</option></select>";
        return true;
    }
    if(region=="f3")
    {
        document.getElementById("ville").innerHTML = "<select name='ville2'
    id='ville2'> <option>Archennes</option><option>Bornival</option></select>";
        return true;
    }
}
```

- Question 5 : de la même manière, ajouter d'autres provinces et d'autres villes.
- Question 6 : est-ce que nous pouvons faire la même chose pour le code Postal ?

TP2.1: Javascript Sidi Ahmed Mahmoudi