ANNEXE page WEB

Différents types de propriétés formatées par le langage css :

En version CSS intégré au html :

– Propriétés des conteneurs (boîtes)(version au sein d'un html)

```
<divstyle="border : solid ; border ;  color : blue ; margin ;  left : 20 px ; margin ;  top : 25 px ; height : 100 px
; width :100 px ; position : relative ">
<span style="border : solid ; borde r ;  color : green ; position : relative ; top : 70 px ; left : 2>
xxx
</span>
<span style="border : solid ; border ;  color : yellow">
yyy
</span>
<span style="border : solid ; border ;  color : red ; position : absolute ; top : 30 px ; left : 10 px height : 20
px ; right : 50 px">
z z z
</span>
</div>
```

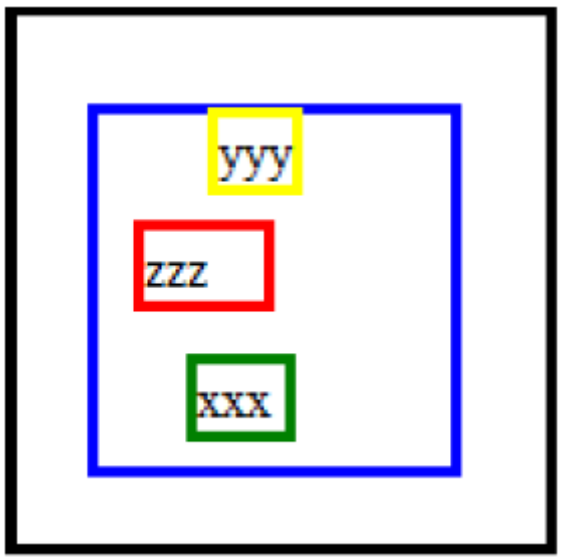
OU en version HMTL + CSS : (je retire les cadres bleu et noir pour raccourcir)

HTML :

```
<!DOCTYPE html>
<html>
        <head>
                <title>  XYetZ </title>
                <link href="style1.css" rel="stylesheet" media="all" type="text/css">
                <meta charset="utf-8">
        </head>


        <body>
                <div style=hautJauneCentre>
                        <p>yyy</p>
                </div>
                <div style=legereDroiteRouge>
                        <p>zzz</p>
                </div>
                <div style=legereBasCentreVert>
                        <p>yyy</p>
                </div>
        </body>
```

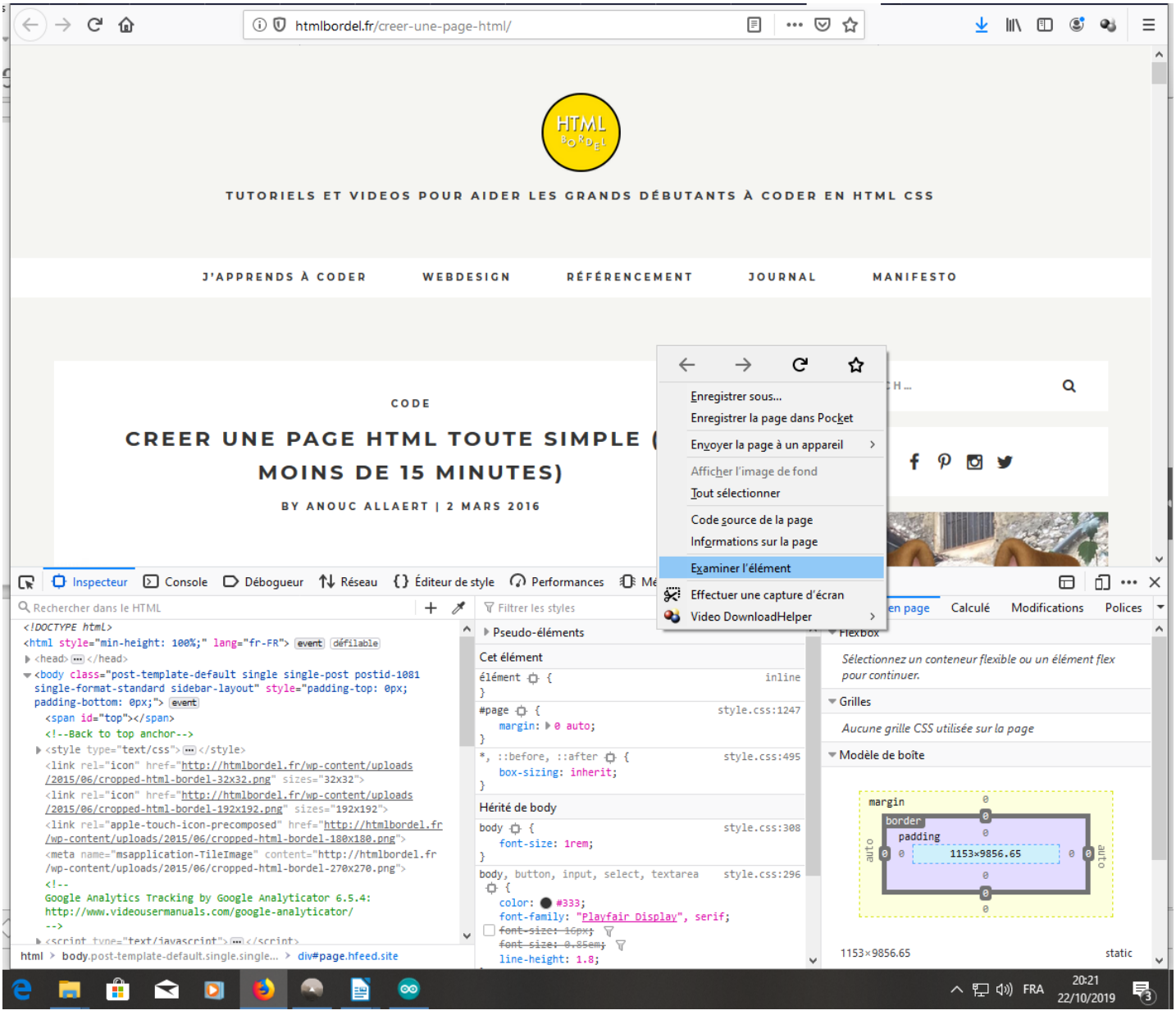
CSS : fichier style1.css

```
hautJauneCentre{
span style"border : solid ; border ;  color : y ellow"
}
legereDroiteRouge
{
span style="border : solid ; border ;  color : red ; position :
absolute ; top : 30 px ; left: 10 pxheight : 20 px ; right : 50 px"
}
legereBasCentreVert{
span style="border : solid ; border ;  color : red ; position :
absolute ; top : 30 px ; left: 10 pxheight : 20 px ; right : 50 px"
}
```



Quelque soit la forme adoptée, le css donne :
– Propriétés textuelles
– Propriétés des listes
– Propriétés de tableaux
– Propriétés de positionnement

Un jeu, examiner ou inspecter une page WEB, retrouver les balises html et css ...puis trouver le liens vers le css et modifier la mise en forme…



balises HTML et CSS

HTML :
Insérer des images ou gif.

Copier un lien vers l'image…
****
alt au cas où l'image ne s'affiche pas
title apparaît au survol par la souris

faire une liste :

(ici trois trucs identiques, le dernier aura l'image sous le texte, les autres à la suite)

 ou <ol> (non ordonnée--→ avec un point au départ/ ordonnée → avec un numéro au départ)

```
    <li> texte
    <img  src= « adresse »></li>
    <li> texte
    <img  src= « adresse »></li>
    <li> texte<br/>
    <img  src= « adresse »></li>
</ul> ou </ol>
```

Insérer une table.

Une table est créée grâce à la balise <table>.
Cette balise contient ensuite une série de balises <tr> (table row) qui définissent les lignes.
Enfin chaque balise <tr> contient une série de balises <td> (table data) qui définissent les cellules.
Exemple 1 :

```
<table>
    <tr>
        <td>(0,0)</td>
        <td>(0,1)</td>
    </tr>
    <tr>
        <td>(1,0)</td>
        <td>(1,1)</td>
    </tr>
</table>
```

Il est possible d'ajouter un titre à une table grâce à la balise <caption>, Ce titre apparaît alors centré au-dessus du tableau. Il est également possible de définir certaines cellules comme étant des lignes d'en-tête en remplaçant la balise <td> par la balise <th>.

Modifications de la couleur, de la forme et de la taille du texte :

dans la balise de début h1 ou p… on insère <h1 style= « color:green »>
** Voici le texte d'un paragraphe.
La balise de retour ligne va imposer un coupure dans le paragraphe, mais sans provoquer de saut de ligne. **
**
 pour aller à la ligne en plein paragraphe(par défaut , on y va en fermant une division)**

Ce texte apparaît en gras.
Ce texte apparaît en italique.
Ce <u>texte</u> sera souligné.

<acronym title= ”Union Européenne ”>UE</acronym> l’explication apparaît au survol de la souris

Ajouter un lien :

objet image texte
par exemple :
<img src="indx.jpeg">

<img src="indx.jpeg"> → l’image indx.jpeg qui est dans le même dossier

Truc en plus : envoi vers des **ancres** sur des pages web :

sur la page MaPage.html appelée on place des ancres avec id :
<h2 id="mon_ancre">Titre</h2>

Dans le lien, on appelle l’ancre :
Aller vers l'ancre si c’est au sein de la page qui appelle.
ou si c’est une autre page que celle qui appelle :
Aller vers l'ancre

la page s’ouvrira à l’endroit de l’ancre… Pratique !

Ajouter un tableau :

code html :

```
<table border 2px solid red width=600px > < !-- ajoute un tableau avec des caractéristiques -->
<caption>Mon premier tableau de ma vie en html</caption> < !-- ajoute un titre au tableau-->
        <tr>  < !-- ajoute une ligne-->
            <td width= ‘40 %’>Nom</td>  < !-- ajoute une case sur la ligne… Dans l’ordre...-->
            <td width= ‘40 %’>Prénom</td>< !-- avec une précision sur les largeurs...-->
            <td width= ‘20 %’>Age</td>
        </tr>
        <tr>
            <td>Giraud</td>
            <td>Pierre</td>
            <td>25 ans</td>
        </tr>
        <tr>
            <td colspan=2 >Joly</td> < !-- la case occupe 2 colonnes-->
            <td>Pauline</td>
            <td>23 ans</td>
        </tr>
    </table>
```

code css :

```
#td
{border: 1px solid black ;} < !-- ajoute bordure autour des cases de 1px en format solid noire.-->
```

Code html :
<td class="PAUSE">PAUSE</td> la case Pause prend les caract css de .PAUSE

coté css :
.PAUSE { background-color: Wheat;
  font-style: italic;}

Peut marcher pour des balises cases mais aussi texte, tableau….

ajout d'un css :

Soit directement dans une balise, dans le code HTML :
<p **style="color:red;"** >Texte.</p>
Soit dans l'en tête(head)

Soit on doit enregistrer un fichier nomdufichier.css dans le même dossier que le fichier html

dans la balise de la partie du code html que l'on veut mettre en forme, insérer une balise lien vers css :
<link rel="stylesheet" href="nomdufichier.css">
enregister tout et on appelle chaque #style du fichier css quand on en a besoin….

body
{**background-color :** purple ;}

ATTENTION, les « : » doivent ÊTRE COLLES à la COMMANDE !!!!!

premiere page.html | premiere page.css

```
1  h1
2  {
3      color:blue;
4  }
5  body
6  {
7      background-color: purple;
8  }
9
10 #nomdebloc
11 {
12  width: 800px;
13 margin:0 auto;
14 background-color: white;
15 text-align: center;
16 align-content: center;
17 }
```

pour créer un bloc de contenu :

dans le html :

balise pour créer un bloc de contenu dans le quel on mettra ce qu'on veut :

<**div id**=nom de bloc>

dans le css :

on appelle le bloc de contenu et on donne ses caractéristiques :

#nom de bloc

{

width: 800px;

margin:0 auto;

background-color: white;

text-align: center;

align-content: center;

}

ici largeur 800px ; placé au milieu de la page ; fond blanc ; aligné centré pour le texte et le contenu.

Dans le html, on place les balises

</div id=nomdebloc>

de part est d'autre de notre bloc qui contient n'importe quoi : images, table, paragraphe, titre....

Grâce au fichier css, on peut l'appeler de plusieurs pages html différentes , la mise en forme sera toujours la même. !!!

On distingue 4 types de positionnements : absolu (**absolute**) : la position du bloc est définie par rapport au coin supérieur gauche de la page ; fixe (**fixed**) : tout comme avec " absolu ", on définit la position initiale du bloc par rapport au coin supérieur gauche de la page, à la différence que le bloc restera fixe à l'écran lorsque les internautes utiliseront l'ascenseur pour parcourir la page ; relatif (**relative**) : ce positionnement permet de " décaler " le bloc par rapport à sa position normale dans la page ; statique (**static**) : c'est le positionnement normal, par défaut, qui respecte l'ordre d'apparition des blocs dans le code source.

#nomDeBloc {position: absolute; top:80px ; } **top**, **bottom**, **left** et **right** pour la référence

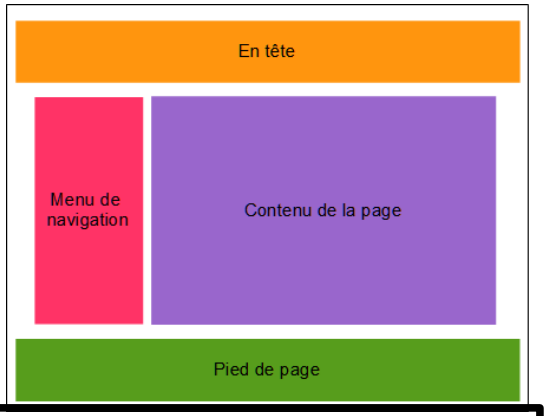
premiere page.html × premiere page.css ×

```
<!DOCTYPE html>
<html>
<head>
    <div id=nomdebloc>
    <meta charset="utf-8>"
    <title>Bon anniversaire Sly ! </title>
    <link rel="stylesheet" href="premiere page.css">
    <basefont: size=72>
    </div id=nomdebloc>
</head>
<body>

    <link rel="stylesheet" href="premiere page.css">
    <h1> Bon anniv Sly!</h1>
    <img src="https://media.giphy.com/media/GFamofJHhruow/giphy.gif">
    <p>Maintenant t'es un vieux con</p>
    <p> blablabla <br> Et puis blablabla </p>
<ol>
    <a href="https://p0.storage.canalblog.com/09/86/137590/114343406.jpg"><div id=nomdebloc>
        <li>ça mitraille sec<br>
        <img src="https://www.yves.brette.biz/public/humour/sapin_vert_parfum_chatte.png"</li></a>
    </div id=nomdebloc>
    <li>ça mitraille sec<br>
        <img src="https://www.yves.brette.biz/public/humour/sapin_vert_parfum_chatte.png"</li>
    <li>ça mitraille sec<br>
        <img src="https://www.yves.brette.biz/public/humour/sapin_vert_parfum_chatte.png"</li>
</ul>
</body>
</html>
```

Positionner un bloc, ajouter des bordures…

Les balises structurantes de flux naturel :

Pour faciliter la suite de la mise en forme, les boites `menu` et `contenu` seront englobées dans un autre cadre `main`. Le but de l'opération est donc de placer les cadres `menu` et `contenu` l'un à côté de l'autre.



dans le html :

```
<div id="entete">
        En tête
</div>

<div id="main">
         <div id="menu">
        Menu de navigation
        </div>
        <div id="contenu">
        Contenu de la page
        </div>
</div>

<div id="footer">
         Pied de Page
</div>
```

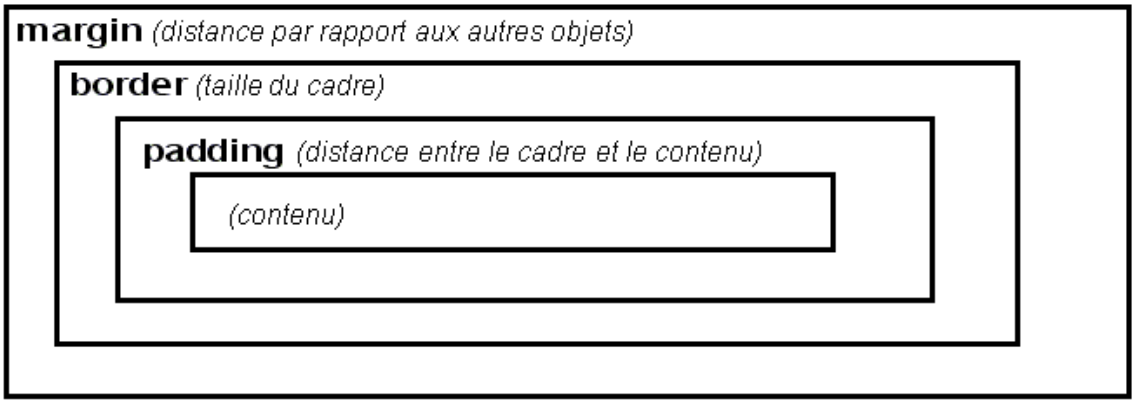
Dans le css :

```
#entete, #menu, #contenu, #footer {padding:1px 0}

#entete {background-color:#FF9900; text-align:center;}

#main {max-with:960px; margin:auto; }

#menu  {float:left; width:240px; background-color:#FF3366;}

#contenu {margin-left:245px; background-color:#9966FF;}
#footer {background-color:#669933; text-align:center; clear:both;}
```

Les instructions principales :

positionnement des cadres :

Une boite (chaque élément "bloc" ***sauf les tables et autres exceptions qui ont des priorités à part*** **) possède l'anatomie suivante:**



Chaque élément correspond à un sélecteur CSS qui permet de définir la largeur générale (des 4 cotés) ou encore la largeur de chaque côté. On peut aussi définir dessin et couleur du cadre.

7.2 Les bords, les cadres et le couleur

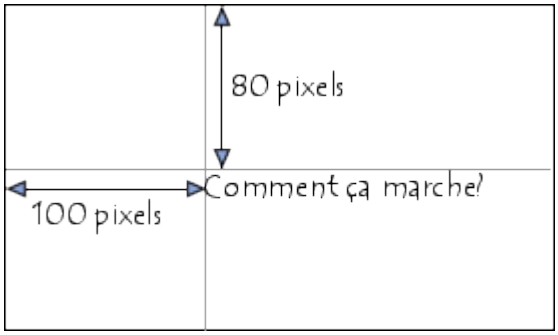
Attributs	Valeurs	se charge de	exemple
margin	pt, px, cm, %	4 marges	*body {margin:1cm;}*
margin-top		marge en haut	*p {margin-top:10px;}*
margin-bottom		marge en bas	*h3 {margin-bottom:3pt;}*
margin-left		marge à gauche	*img {margin-left:50px;}*
margin-right		marge à droite	*p.citation {margin-right:10pt;}*
border	pt,px, cm, %	largeur du cadre	*p {border:5px;}*
border-top			*h1 {border-top:0.2cm;}*
etc ...			
border-style		style de cadre	
	solid	ligne simple	p {border-style:solid;}
	double	ligne double	h1 {border-style:double;}
padding	pt,px,cm,%,etc	marge intérieures	p {padding: 5px;}
color	valeur hexa /nom	couleur d'un élément	#menu {color:#000000;} body {color:blue;}
background	aussi	couleur de l'arrière-plan	h1, h2 {background:silver;}

La position relative se fait par rapport à d'autres éléments, comme une image, c'est-à-dire que les éléments contenus dans la balises *DIV* ou *SPAN* seront positionnés à la suite des éléments HTML après lesquels ils se trouvent. .

Positionnons le texte "Comment ça marche?" à 80 pixels du haut de la fenêtre et à 100 pixels à gauche de la fenêtre :

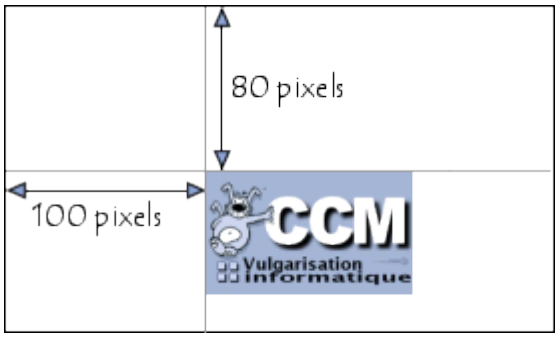
```
<HTML>
        <BODY>
        <SPAN style="position: absolute; top: 80 px; left: 100 px;">
```

```
            Comment ça marche?
        </SPAN>
        </BODY>
</HTML>
```



Positionnons l'image "test.jpg" à 80 pixels du haut de la fenêtre et à 100 pixels à gauche de la fenêtre (l'image fait 103x61) :

```
<HTML>
        <BODY>
                <SPAN style="position: absolute; top: 80 px; left: 100 px;width: 103px; height:
61px">
                        <IMG SRC="test.jpg" >
                </SPAN>
        </BODY>
</HTML>
```



allure du cadre :
background-color : en rvb/hexadécimal ;
texte :
text-align:left, right or central ;
font-size : 15px ;
font-style : bold//// ;
font-color : red, blue, black…. ;
font-family : arial, comic…. ;
marges :
margin-left:50px ;
margin-top:50px ;

On rentre dans le complexe...

Mise en forme précises d'un tableau :

/* transformer les caract de toute une ligne ou une colonne, ici la seconde colonne et la dernière devient en fond Azure*/
td:nth-child(2),td:nth-last-child(1){color: Azure;}
nth : depuis le début
nth-last :depuis la fin

/* afficher les matières quand la souris passe dessus : la police passe alors à noire avec une bordure rouge de 3px de large*/
td:hover {border: 3px solid Red;color: black;}

Ajouter un formulaire :

Un formulaire avec entrées à écrire.
Exemple :

```
<!DOCTYPE html>
<html lang="fr">
  <head>
    <title>Exemple de formulaire</title>
    <meta charset="utf-8">
                <script src="Form_scripts.js" type="text/javascript"></script>
      </head>

  </head>
  <body>
    <!-- Titre du formulaire -->
    <p>Formulaire d'inscription</p>


    <form name="formulaire" method="post"
action="https://webia.lip6.fr/~muratetm/diu/Form_data.php" target="_blank">

<!-- pour envoyer les données du formulaire ajouter à une adresse précise par défaut « méthod get, sinon on ajoute method=post pour que les données soit cachées dans un post HTTP -->
      <!------------------------------------------->
      <!-- Partie 1 : Renseignements administratifs -->
      <!------------------------------------------->
      <p>Renseignements administratifs</p>
      <!-- Nom -->
      <div>
        <label for="Prenom">Prenom</label>
        <input name="Prenom" class="Required" size="100" maxlength="80" type="text">
      </div>

      <div>
        <label for="Nom">Nom</label>
        <input name="Nom" class="Required" size="100" maxlength="80" type="text">
      </div>
```

```
        <div>
          <label for="Adressemail">Adresse mail</label>
          <input name="Adressemail" class="Required" onfocusout="checkEmail(this)"
onfocusin="writeEmail(this)" size="100" maxlength="80" type="text">
        </div>

        <!--------------------------------------->
        <!-- Partie 2 : Options d'inscription -->
        <!--------------------------------------->
                        <p>Quelle voiture de fonction vous ferait plaisir?</p>
        <label for="voiture">Choix de la voiture de fonction</label>
        <select name="voiture">
                                <option value="volvo">Volvo</option>
                                <option value="saab">Saab</option>
                                <option value="mercedes">Mercedes</option>

                        </select>

        <!--------------------------->
        <!-- Partie 3 : Validation -->
        <!--------------------------->
        <br>
        <br>
        <input value="Valider l'inscription" onclick="valider(this)" type="button">
<!-- le type « submit » qui permet de soumettre les données en lignes…. peut être remplacé par un type
« button » pour envoyer vers une fonction javascript -->
      </form>
   </body>
</html>
```

balise **<form>** qui représente la section contenant les contrôles du formulaire permettant à un utilisateur d'envoyer des données à un serveur web
balise **<label>** permet de définir un texte d'information associé à une balise **<input>**, c'est une balise importante pour que le formulaire soit accessible aux personnes en situation de handicap (notamment pour être lu par synthèse vocale). L'association entre une balise <label> et une balise <input> est spécifiée à l'aide des attributs « **for** » et « **name** » de ces dernières balises.

<u>Un formulaire à choix multiple :</u>

Balise <select>

```
                <option value= « truc1 »>Truc1</option>
                ….
                ….
          </select>
ex :                    <select>
                                <option value="volvo">Volvo</option>
                                <option value="saab">Saab</option>
                                <option value="mercedes">Mercedes</option>

                        </select>
```

<u>Ajouter des envois vers JAVASCRIPT :</u>

on crée un fichier ffffff.js qui contient le programme

un exemple :
dans le html :

```
<head>
	<title>Exemple de formulaire</title>
	<meta charset="utf-8">
	<script src="Form_scripts.js" type="text/javascript"></script>
</head>
<!--…
Dans la case concernée par la/les fonction du fichiers Form_scripts.js : -->
<div>
		<label for="Adressemail">Adresse mail</label>
		<input name="Adressemail" onfocusout="checkEmail(this)"
onfocusin="writeEmail(this)" size="100" maxlength="80" type="text">
	</div>
```

le fichier « Form_scripts.js » :

```
// fonction appelée lorsque le focus sort du champ de saisie de l'adresse mail
function checkEmail(email) {
   // Recherche d'un "@"
   var atPos = email.value.indexOf('@');
   // Recherche du dernier "."
   var lastDotPos = email.value.lastIndexOf('.');
   // Vérifier qu'il y a bien un "@" suivi d'un "."en laissant au moins un caractere derriere et encadré
par au moins un caractère
   if (atPos != -1 && atPos < lastDotPos && lastDotPos < email.value.length-1 && lastDotPos-
atPos > 1){
      // Si la vérification réussie, mettre le texte en vert
      email.style.color = "green";
      return true;
   }
   else{
      // Si la vérification échoue, mettre le texte en rouge
      email.style.color = "red";
      return false;
   }
}

function writeEmail(email) {
   {
      // Si la souris est dans la case
      email.style.color = "black";
      return true;
   }
}

// fonction appelée lorsque l'utilisateur valide la saisie du formulaire
function valider() {
   // Vérifier que les champs requis sont bien renseignés
   var champsRequis = document.getElementsByClassName('Required');
```

```
    for (let champ of champsRequis){
        if (champ.value == ""){
            alert(champ.name + " requis");
            return;
        }
    }

    // Soumettre le formulaire au serveur
    document.formulaire.submit();
}
```

Exercice 8 : Affichage conditionnel

Appliquez l'ensemble des éléments que vous avez travaillé dans les exercices précédents pour n'afficher le champ de saisie des restrictions alimentaires uniquement si l'utilisateur choisi de participer au déjeuner.

Aide : Vous aurez certainement besoin d'utiliser la propriété CSS « visibility » qui peut prendre comme valeur « visible » et « hidden ». Mais aussi l'évènement « onchange » sur la balise <select>.

HTLM :

```
<p>Quelle voiture de fonction vous ferait plaisir?</p>
        <label for="voiture">Choix de la voiture de fonction</label>
        <select name="voiture" onchange="Kasher(this)">
                                <option value="volvo">Volvo</option>
                                <option value="saab">Saab</option>
                                <option value="mercedes">Mercedes</option>

                        </select>
        <div>
            <label for="typedegente" style="visibility:hidden" class="visibility" >Type de
gentes</label>
            <input name="typedegente" style="visibility:hidden" class="visibility" size="100"
maxlength="80" type="text">
        </div>
```

JS :

```
function Kasher(X){
        var marque = document.getElementsByClassName('visibility');
        if (X.value=='saab'){
                for (let champ of marque){
                        champ.style.visibility='visible';
                }

        }
        else {
                for (let champ of marque){
                        champ.style.visibility='hidden';
                }
        }
}
```