CR TDM01 HTML CSS

Noémie PRIN et Maria-Bianca ZUGRAVU

Remarques pratiques

Attention à la difference entre *head* et *header* !

- **head**: est placée juste après *<!DOCTYPE html>* *<html>*; il contient, en géneral, les balises suivantes: *<title>* *<meta>* *<link>*
- **header**: est placée dans le *<body>* . Bonne idée de placer ici les images.
- boutons qui ne sont pas cochables en même temps: utilisation de <**radio**> en mettant le paramètre **name** identique dans la déclaration des boutons.

Points importants CM

html

- lors d'une balise **input** dans un formulaire si on veut afficher une boîte où l'utilisateur doit rentrer ses données en écrivant un petit message on utilise le paramètre : **placeholder**

Example:

```
<label> Name : </label> <input type = "text" name = "monNom" id ="name" placeholder="Sa
```

- associer une étiquette à un bouton : *<label* **for="name"**> Name :
- faire une saute de ligne: <**br**/>
- faire une ligne horizontale: <**hr**/>

Exemple :

```
<label> Statut:  </label> <input type="radio" name="statut" id="enseignant" /><label f
<input type="radio" name="statut" id="etudiant"/><label for="etudiant">Etudiant</label
```

- **pour placer les éléments dans un rectangle** : *<fieldset>* *</fieldset>*
- **titre associé au rectangle** : *<legend>* Légende *</legend>*
- dans un selecteur, ne pas oublier de préciser le nombre d'option sélectionnées à la fois avec **size**. Exemple :

```
<label>Si étudiant, année : </label> <select name="annee" size="1">
<option value="1">1</option>
<option value="2">2</option>
```

  - définition d'un id class="nomClasse" pour le CSS
  - rectangle, class pour le CSS :

- **légende**

  ```
  <legend class="nomClasse">légende</legend>
  ```

- **bouton** (en fait un élément <**input**>):
  - types possibles :
  - text
  - radio
  - image
  - file
  - submit
  - reset
- **liste**:
- liste simple :

  ```
  <ul>
        <li> item </li>
  </ul>
  ```

- liste numérotée: * *
- liste de définitions:

  ```
  <dl>
        <dt> item </dt> <dd> definition </dd>
  </dl>
  ```

- **tableaux** :

Syntaxe:

```
<!DOCTYPE html>
<html>
   <head>
      <title>Tableau</title>
      <meta http-equiv="content-type" content="text/html;charset=utf-8" />
   </head>
   <body>
      <table border="1">
         <caption>
            <h2>Titre du tableau</h2>
         </caption>
         <thead>
            <tr>
               <th>Titre 1</th><th>Titre 2</th><th>Titre 3</th>
            </tr>
         </thead>
         <tfoot>
            <tr>
```

```
                  <th>Titre 1</th><th>Titre 2</th><th>Titre 3</th>
               </tr>
            </tfoot>
            <tbody>
               <tr>
                  <td>cellule 1</td><td>cellule 2</td><td>cellule 3</td>
               </tr>
               <tr>
                  <td>cellule 4</td>
                  <td>cellule 5</td>
                  <td rowspan="2">cellule 6</td>
               </tr>
               <tr>
                  <td colspan="2">cellule 7</td>
               </tr>
            </tbody>
         </table>
      </body>
   </html>
```

- <**thead**> et <**tfoot**> permettent de répéter l'élément dans les tableaux sur plusieurs paragraphes
- <**tr**> pour déclarer une ligne
- <**td**> cellule normale <**th**> cellule titre/gras
- attributs **rowspan** et **colspan** pour fussionner des cellules

Exemple :

```
<label> Nom : </label> <input type = "text" name = "name" id="identifier" placeholder=

<input class="Bouton" type="submit" name="action" value="ValeurEcriteSurLeBouton" />
```

- rectangle pour entrer du texte avec un message

  ```
  <textarea name="texte" rows="10" cols="80"> Message... </textarea>
  ```

- élément bloc **div** : élément précédé et suivi d'un saut de ligne (paragraphe, tableau, liste)
- <*p*> </*p*> : construire des paragraphes; pour spécifier la justification –> attribut **align**

  ```
  <style type="text/css">
  p {text-align: justify}
  </style>
  ```

- élément inline **span**: élément qui s'insère dans le fil du texte; ne peut contenir que du texte ou d'autres éléments inline (éléments italique, gras)
- em

- strong
- cite
- code
- samp
- kbd
- var
- dfn
- abbr

CSS

On définit la CSS de 2 façons: * directement dans le fichier html via <*style*> :

```
<head>
  <style type="text/css">
    ...
  </style>
</head>
```

- par lien vers une CSS externe via <**link**>

```
<head>
  <link href="fichier.css" rel="stylesheet" type="text/css" />
</head>
```

- padding : espace entre les limites d'un élément et son contenant
- **Sélecteurs**:
- *sélecteur de descendant* : p h2 {color: green}
- *sélecteur d'enfant* : p > h2 {color: green}
- *sélecteur d'adjacent* : p + {color: green}
- *sélecteur de pseudo-classe* : **.**

  Liste des pseudo-classes standards:

  - **:active** : permet de cibler un élément lorsque celui-ci est activé par l'utilisateur. Elle permet de fournir un feedback indiquant que l'activation a bien été détectée par le navigateur. Par exemple si on clique sur un lien qui est de base en bleu il se transforme en rouge
  - **:checked** : le statut du contenu
  - **:hover** : position souris
  - **:first-child**: permet de cibler un élément qui est le premier élément fils par rapport à son élément parent.

- :nth-child(a*n+b) **ou** :nth-child(**odd**) **ou :nth-child**(**even**)
- **:required**: permet de cibler un élément <**input**> pour lequel l'attribut required est activé. Cela permet de mettre en forme les éléments obligatoires pour remplir correctement un formulaire. Par exemple, dans un formulaire où certains champs sont obligatoires comme le nom, le numéro de téléphone.
- **:invalid** : cible tout élément <*input*> pour lequel la validation du contenu échoue par rapport au type de donnée attendu. Ceci permet de mettre en forme les champs non valides pour aider l'utilisateur à identifier et à corriger les erreurs.
- **margin**
- **padding**
- **background-color**
- **color**
- **position**
- **width**, **height**
- **right**, **left**, **top**, **bottom**
- **border-width : thick, thin, medium, 10px**

Pour voir des exemples cliquer ici https://developer.mozilla.org/fr/docs/Web/CSS/Pseudo-classes!

- *sélecteur de pseudo élément première lettre* : p:first-letter {text-transform: capitalize}
- *sélecteur de pseudo élément* ***:before*** *et* ***:after*** : Généralement utilisé pour ajouter du contenu esthétique à un élément via la propriété CSS content. Par défaut, l'élément créé est de type en-ligne (inline). Le content peut être laissé comme vide mais on ne peut pas le supprimer.

```
#example:before {</br>
    content: ""; </br>
    display: block; </br>
    width: 100px;</br>
    height: 100px;</br>
  }
```

Pour voir des exemples cliquer ici: https://developer.mozilla.org/fr/docs/Web/CSS/::before

Différence entre ***:before*** *et* ***::before*** *: dans CSS3 pour différencier les pseudo classes des pseudo éléments*

Positionnement: * flottant : élément positionné à droite ou à gauche du bloc, les éléments qui le suivent dans le flux prennent l'espace qui leur reste * absolu : position n'est pas par rapport aux autres éléments du bloc qui les contient tous, mais par rapport aux limites du bloc lui-même * fixe : position immuable même en cas de scroll de la page. * relatif : position dépendant de l'élément qui a été positionné juste avant dans le flux (élément parent)

ATTENTION! Pour comprendre le positionnement il faut tenir compte de la

taille initiale du flux (+ environ 2 px pour l'encadrement) qu'on fixe avec les attributs width et height car si la taille de la balise qu'on veut placer dépasse cette taille, elle se met pas en ligne horizontale avec l'autre d'avant mais en dessous et ainsi de suite pour les autres

Sélecteurs en bref:

Syntaxe : sélecteur[, sélecteur...] {déclarations}

- sélecteur d'élément :element
- sélecteur d'id : #id
- sélecteur de classe : .class
- sélecteur universel : *
- sélecteur de descendant : element element
- sélecteur d'enfant (descendant direct) : element > element
- sélecteur d'adjacent :element1~element2
- sélecteur d'adjacent direct: element+element

Scripts