

TP n° 9

Consignes les exercices ou questions marqués d'un ★ devront être rédigés sur papier (afin de se préparer aux épreuves écrites du partiel et de l'examen). En particulier, il est recommandé d'être dans les mêmes conditions qu'en examen : pas de document ni de calculatrice. Les questions marquées d'un ◇ sont des questions supplémentaires permettant d'aller plus loin (et ne seront pas forcément corrigées en TP). Tous les TPs se font sous Linux.

But du TP

- Le but de cette feuille d'exercice est :
- familiarisation avec les concepts HTML et CSS

Exercice 0

- Ouvrir un terminal :
- créer un répertoire **IntroInfo** et un répertoire **TP9** à l'intérieur. Note : les fichiers des TPs précédents ne sont pas accessibles directement depuis cet environnement.
 - se placer à l'intérieur du répertoire **TP9**
- On suppose pour les autres exercices que le répertoire **TP9** est le répertoire courant.

Exercice 1

Le but de cet exercice est la création d'une page personnelle. Dans un premier temps on ne créera qu'une simple page HTML, sans information de styles. La feuille de style sera ajoutée dans l'exercice suivant. Dans tout cet exercice, si les supports de cours ne vous donnent pas assez d'indications, vous pouvez utiliser le site W3Schools et en particulier sa référence sur HTML qui explique au moyen d'exemples et de démo la spécification du langage HTML :

<http://www.w3schools.com/tags/default.asp>

Entre chaque question, on s'attachera à toujours créer une page *valide*. Pour savoir si une page est valide, on peut utiliser le validateur du W3C :

http://validator.w3.org/#validate_by_upload

Attention : les pages Web ainsi créées ne sont visibles **que** localement (en ouvrant le fichier avec Firefox par exemple). Pour valider le fichier il faut donc :

- Penser à le sauvegarder
- Cliquer sur **Parcourir** et aller chercher le fichier HTML
- Cliquer sur **Check**

Il faut refaire cette opération après chaque modification.

Une autre façon de faire consiste à aller sur l'onglet "Validate by input" ou directement à l'adresse :

http://validator.w3.org/#validate_by_input

et y copier/coller le code HTML écrit dans votre éditeur de texte.

1. Créer une page vide, mais valide. Cette dernière doit contenir la balise **html**, la balises **head** et **body** et une balise **title** dans la section **head**. Où s'affiche le texte que vous avez mis dans la balise **title** ? Utiliser le validateur pour vérifier votre document. Ce dernier est-il valide ?
2. Ajouter des caractères accentués (çâé) dans la partie **body**. Les caractères s'affichent-ils correctement ? Changez maintenant l'attribut **charset** de la balise **meta** pour qu'il vaille "ISO-8859-1". Télécharger le fichier et l'ouvrir localement avec votre navigateur (CTRL-O, puis aller chercher le fichier html téléchargé) Les caractères s'affichent-ils correctement ? pourquoi ? (à la fin de la question, effacez les caractères accentués et revenez au document de la question 1).

3. Rajouter un titre de section en utilisant la balise `<h1>Page personnelle de ...</h1>` (insérez votre nom). La page est-elle valide ?
4. Rajouter du texte (un court paragraphe vous décrivant ou un texte générique à la *lorem ipsum*) sous la balise `h1`. La page est-elle valide ? Ajouter maintenant un élément ` `. La page est-elle valide ? Dans quel situation a t'on le droit d'utiliser un `li` ? Supprimer cette balise.
5. Créez une table avec votre emploi du temps de la semaine, sous votre paragraphe. Une table se crée en utilisant la balise `<table>...</table>`. Dans cette balise, on peut utiliser des balises `<tr> ...</tr>` pour chaque lignes. Chaque balise `<tr>...</tr>` peut contenir une suite de balise `<td>...</td>` pour les colonnes. On souhaite que la table ait 5 lignes (une ligne de titre avec les jours de la semaine, lundi - vendredi et 4 lignes 2 pour les créneaux du matin et 2 pour les créneaux de l'après-midi). La table doit avoir 6 colonnes, une pour chaque jours de la semaine (sans les week-end) et une pour l'intitulé de chaque créneau.
6. Mettre les titre de colonne et de lignes en gras
7. rajouter sous la table une liste non-énumérée avec vos loisirs/activités extra-universitaires, précédé d'un titre de sous-section *Loisirs* (balise `<h2>`)
8. rajouter ensuite une liste énumérée de choses à faire (*todo-list*), précédé d'un titre de sous-section *Choses à faire*
9. Rajouter une nouvelle sous-section intitulée « où me trouver ? ». Ajouter ensuite une balise `<p>` contenant un lien vers l'URL suivante :

`https://goo.gl/maps/cKufC`
10. Rajouter enfin une sous-section « liens utiles ». Le contenu de cette dernière sera un paragraphe de texte contenant 5 ou 6 liens hypertextes (balise `...` pointant vers des sites que vous trouvez utile.
11. Enfin, une fois que votre document est valide, rajoutez l'image se trouvant à l'URL :

`http://www.w3.org/html/logo/downloads/HTML5_Logo_64.png`

 (et revalidez une dernière fois).

Exercice 2

On souhaite maintenant écrire une feuille de style pour la page conçue à l'exercice précédent. Dans tout ce qui suit, et sauf mention contraire, on écrira toutes les propriétés CSS dans un fichier CSS séparé, référencé par la page HTML. On n'utilisera pas l'attribut `style="..."` mais on fera au contraire un usage judicieux des attributs `class`, `id` et des sélecteurs CSS.

1. créer un fichier `exo2.css` et le référencer dans la page HTML
2. Faites en sorte que tous les éléments de la page utilisent une police sans sérif (`font-family: sans`), en utilisant le sélecteur `*`. Faites de même en utilisant uniquement le sélecteur `body` puis uniquement le sélecteur `table`. Que constatez vous ? Que peut-on en déduire sur la transmission de la propriété `font-family` aux sous-éléments ?
3. Centrez le titre horizontalement (élément `h1`, propriété `text-align`), soulignez le et changez sa couleur (selon votre goût)
4. Faites en sorte que la première lettre du paragraphe de description soit en gras et très grande. Vous pouvez mettre uniquement la première lettre dans un élément `C` (si C est la première lettre de votre paragraphe) puis assigner un style particulier à l'élément d'id `p1`. Utilisez une police de taille 200%.
5. Forcer la largeur de tous les paragraphes (éléments `p`) à être de 40em.
6. faites en sorte que le haut de la première lettre du paragraphe (l'élément `span` d'id `p1`) soit alignée avec le texte (et non pas le bas). Rajoutez la propriété `float:left;` à cet élément. Que constatez vous ?
7. faites en sorte que chaque case de la table fasse 10em de largeur, 3em de hauteur, et que le texte s'y trouvant soit centré horizontalement et verticalement.
8. faites en sorte que les lignes (élément `tr`) paires aient un fond gris foncé et les lignes impaires aient un fond clair (vous pouvez vous inspirer de l'exercice 1)
9. faites en sorte que la première case de la table (en haut à gauche) ait un fond blanc (en utilisant un id)

```

1 EMOJI_CODE = [ (169, 169), (174, 174), (8252, 8252), (8265, 8265),
2                 (8419, 8419), (8482, 8482), (8505, 8505), (8596, 8601),
3                 (8617, 8618), (8986, 8987), (9000, 9000), (9167, 9167),
4                 (9193, 9203), (9208, 9210), (9410, 9410), (9642, 9643),
5                 (9654, 9654), (9664, 9664), (9723, 9726), (9728, 10223),
6                 (10548, 10548), (10549, 10549), (11008, 11263),
7                 (12336, 12336), (12349, 12349), (12951, 12951),
8                 (12953, 12953), (126976, 127023), (127136, 127231),
9                 (127232, 128591), (128640, 128767), (129296, 129387),
10                (129408, 129504) ]
11
12 LEN = 15    # on veut des tables de LEN colonnes
13
14
15 #Compléter ci-dessous

```

FIGURE 1 – Le fichier `emoji.py`

10. faites en sorte que lorsqu'on survole l'une des cases, sauf la première, le fond et le texte changent de couleur
11. faites en sorte que les liens hypertexte de la page ne soient pas soulignés et soient encadrés en bleu. Lorsqu'un lien est visité le cadre doit devenir rouge.
12. faites en sorte que le bloc contenant des listes non énumérées aient une largeur de 200pt et un fond gris clair
13. faites en sorte que la couleur des titres de sous-section soit la même que celle du titre de la page
14. faites en sorte que l'image HTML5 se trouvant en fin de fichier apparaisse toujours en bas à droite de la fenêtre du navigateur (même si on défile). Vous devrez modifier le type de position et les propriétés `right` et `bottom`.
15. redimensionnez la fenêtre du navigateur de manière à ce qu'apparaisse des barres de défilement. Changez la propriété `position` de l'image à `absolute`. Que se passe-t-il ?
16. Faites en sorte que votre texte de paragraphe introductif soit très long (au pire, copiez-collez le plusieurs fois). Forcer la hauteur de l'élément `p` correspondant à faire au maximum 100pt. Que se passe-t-il pour le texte ? Mettez le fond du paragraphe en jaune (composante rouge et verte plus forte que le bleu ou simplement `yellow`). Faites en sorte que le texte puisse défiler et ne dépasse pas de la boîte.

Exercice 3◇

Récupérer le fichier Python `emoji.py` sur la page du cours. Le contenu de ce dernier est donné à la figure 1. Ce fichier définit un tableau de couples d'entiers. Un couple d'entier (a, b) de ce tableau indique que les caractères dont le code Unicode est compris entre a et b (inclus) est un « symbole ». La variable globale `LEN` définit un entier (ici 15).

1. Écrire un programme Python qui imprime sur la sortie standard (au moyen de `print`) un fichier HTML complet et valide contenant une table de `LEN` colonnes (et autant de lignes que nécessaire). La table contient tous les caractères représentés par les intervalles de la variable `EMOJI_CODE`. Chaque case de la table aura deux lignes : la première représente le caractère, la seconde son code. On rappelle que la fonction Python `chr(n)` renvoie une chaîne de taille 1, contenant le caractère dont le code est `n`.
2. Ouvrir le fichier avec firefox pour contrôler que la plupart des caractères s'affichent bien ¹.

1. Certains caractères peuvent être affichés comme un rectangle blanc contenant un code. Une telle icône est appelé un « tofu » (par analogie avec les petits morceaux de tofu) et représente un caractère non supporté par la police utilisée pour l'affichage.