

Initiation aux feuilles de style

Copiez le code HTML suivant dans un fichier `exercice1.html` et créez une feuille de style externe dans un fichier `monstyle.css` qui lui sera liée :

```
<HTML>
<HEAD>
<TITLE>Exercice 1</TITLE>
<link rel="stylesheet" type="text/css" href="monstyle.css">
</HEAD>
<BODY style="font-family:monospace">
<ul>
<li>Getting started with HTML
<li>Undrestanding Javascript
<li>Introducing CSS.
</ul>
<h1>Getting started with HTML</h1>
<p>
HTML is an acronym for Hypertext Markup Language, which is the language we have used to
create webpages since the first webpages arrived on the web. HTML roots are from an older
markup language that was used in document publishing, called SGML (Standard Generalized
Markup Language). One interesting aspect of HTML and its relationship with browsers was that browsers
were designed to be backward compatible and forward compatible. Creating a browser that is
backward compatible is relatively easy because the problem domain is known, but how is
forward compatibility accomplished? Browsers were created to ignore tags that they didn't
recognize.
</p>
<p>
Although HTML served its purpose for quite some time, people wanted more. Another
evolving technology, called XML (eXtensible Markup Language), was becoming popular. XML
looks a lot like HTML because both languages use tags. However, there are several big differences.
First, XML always requires matching end tags for every tag, although you can use
a shortcut notation to denote a starting tag and ending tag together. Another difference is
that HTML has a very specific set of tag names, and each of these tags denotes a formatting
feature that is to be applied to the rendered webpage. XML doesn't have a defined set of tag
names, so you create your own tag names, and the tags can represent anything.
</p>
</BODY>
</HTML>
```

Partie 1

Mettre en forme le texte :

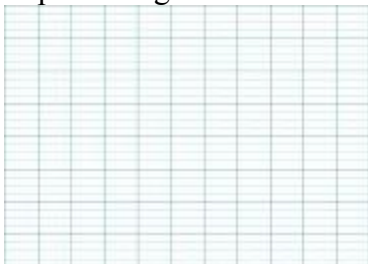
- 1) A l'aide de l'attribut `style` au niveau de la balise `body`, redéfinir la police (propriété `font-family`).
- 2) Régler la taille par défaut du texte à l'aide de la propriété `font-size`. (lui attribuer la valeur 12)
- 3) Régler la taille du premier paragraphe à 80% en utilisant l'attribut `style`.
- 4) Modifier les balises `H1` : caractères gras (`font-weight`), souligné (`text-decoration`) et majuscule (`text-transform`).
- 5) Définir une police différente de celle appliquée à `body`.
- 6) Centrer le texte de la balise `H1` (propriété `text-align`).
- 7) Appliquer au premier paragraphe un interligne de 3em (`line-height`) et un écart entre les lettres de 0,4em (`letter-spacing`).

- 8) Régler l'espace entre les mots à 0.6em (word-spacing) pour le premier paragraphe.
- 9) Limiter la largeur du premier paragraphe à 400 pixels, et appliquer un alignement justifié au texte.
- 10) Ranger tous les styles appliqués au premier paragraphe dans le sélecteur #paragraphe1 qui lui sera ensuite appliqué.
- 11) Centrer tout le texte à partir de la balise body (text-align). Qu'est ce que vous remarquez ?
- 12) Affecter paragraphe2 à l'ID du deuxième paragraphe et appliquer-lui le style suivant :
#paragraphe2{font : italic bold 100% arial, Helvetica, sans-serif;}
- 13) Mettre le style #paragraphe2 en commentaire et donner son équivalent au format détaillé.

Partie 2

Couleurs et images d'arrière plan :

- 1) Appliquer un fond vert clair au premier paragraphe, un fond rose au deuxième paragraphe, un fond bleu clair au titre (h1) et un fond jaune à la liste.
- 2) Appliquer un fond rouge à body.
- 3) Copier l'image suivante sur Paint



Enregistrer la sous le fichier cahier.jpg en le plaçant sous le même répertoire de votre fichier html.

- 4) Appliquer l'image précédente à l'arrière plan du deuxième paragraphe.
- 5) Noter l'effet de background-repeat : no-repeat.
- 6) Centrer l'image d'arrière plan du deuxième paragraphe (background-position).

Partie 3

Jeu de bordures :

- 1) Ajouter une bordure au titre d'épaisseur 3px, de style 3D sortant (ridge) de couleur bleue plus foncée que le fond.
- 2) Tester les autres valeurs pour le style de bordure :double, groove, hidden, solid, dashed, dotted, outset, inset.
- 3) Afficher un trait plein vert foncé à droite du premier paragraphe.
- 4) Copier l'image suivante vers Paint et enregistrer la sous ofppt.jpg (dans votre répertoire de travail):



- 5) Insérer cette image en bas de votre page, associez à cette image un lien vers le site <http://www.ofppt.ma>, centrer l'image.
- 6) Créer un sélecteur .imageStyle qui permet d'appliquer une bordure sur les images avec 2px comme épaisseur, solid comme style et une couleur noire.
- 7) Joindre l'image à ce dernier sélecteur pour appliquer effectivement le style.

Partie 4

Mettre en forme les listes :

- 1) Donner à la liste un identificateur "menu" afin de modifier le style de cette liste.
- 2) En utilisant la propriété list-style-type, remplacer les puces par des cercles puis par des carrés.
- 3) Supprimer les puces de la liste.
- 4) On peut remplacer la puce par une image quelconque avec la propriété list-style-image, trouver ou réaliser une image pour remplacer les puces.
- 5) Retirer toutes les puces, ajouter une propriété margin = 0 et padding = 0.

Partie 5

Liens hypertextes :

- 1) Pour chaque élément de liste créer un lien fictif. (href=#)
- 2) Ajouter un sélecteur #menu a dans votre feuille de style, permettant de mettre le texte en vert et non souligné.
- 3) Appliquer une couleur rouge aux liens visités descendant de la liste #menu.
- 4) Appliquer une couleur rose aux liens au moment de l'activation.
- 5) Pour donner l'aspect d'un bouton aux liens, appliquez le style suivant : color: #FFFF33; text-decoration: none; background: #fc0; line-height: 25px; border : 1px outset #fc0; text-align:center;
- 6) Changer la couleur du texte du lien lorsqu'il est survolé vers le rouge.
- 7) Au moment du clic sur le lien, celui-ci doit donner l'impression qu'on a cliqué sur un bouton, pour cela appliquer une bordure solid pour le lien au moment de l'activation.

Partie 6

Positionnement en CSS

- 1) Copier le code HTML suivant dans un fichier partie6.html et associer lui une feuille de style monstyle2.css :

```
<HTML>
<HEAD>
<TITLE>Partie 6</TITLE>
<link rel="stylesheet" type="text/css" href="monstyle2.css">
</HEAD>
<BODY>
<div>
<p>
HTML is an acronym for Hypertext Markup Language, which is the language we have used to
create webpages since the first webpages arrived on the web. HTML roots are from an older
markup language that was used in document publishing, called SGML (Standard Generalized
Markup Language).
One interesting aspect of HTML and its relationship with browsers was that browsers were
designed to be backward compatible and forward compatible. Creating a browser that is
backward compatible is relatively easy because the problem domain is known, but how is
forward compatibility accomplished? Browsers were created to ignore tags that they didn't
recognize.
</p>
<p>
Although HTML served its purpose for quite some time, people wanted more. Another
evolving technology, called XML (eXtensible Markup Language), was becoming popular. XML
looks a lot like HTML because both languages use tags. However, there are several big differences.
First, XML always requires matching end tags for every tag, although you can use
a shortcut notation to denote a starting tag and ending tag together. Another difference is
that HTML has a very specific set of tag names, and each of these tags denotes a formatting
feature that is to be applied to the rendered webpage. XML doesn't have a defined set of tag
names, so you create your own tag names, and the tags can represent anything.
</p>
```

```
<div>
</BODY>
</HTML>
```

- 2) Donner aux deux paragraphes les identifiants `paragraphe1` et `paragraphe2`.
 - 3) Appliquer un fond jaune à `div`, bleu à `paragraphe1` et vert à `paragraphe2`.
- NB :** Observer le positionnement naturel des blocs dit positionnement dans le flux. Les paragraphes se placent dans l'ordre d'apparition, l'un en dessous de l'autre, à l'intérieur du bloc conteneur (`div`), lui même dans un autre bloc conteneur (`body`).
- 4) Appliquer au bloc `div` une marge interne de 6px.
 - 5) Réduire la largeur des blocs avec l'attribut `width`, 200 px pour le `paragraphe1` et 250 pour le `paragraphe2`.
 - 6) Pour centrer un bloc la propriété `text-align` ne fonctionne pas, car elle ne s'applique pas aux blocs. Il suffit de fixer la largeur du bloc et de fixer les marges droites et gauches à `auto`. Centrer ainsi le `paragraphe2`.
 - 7) Pour sortir du flux courant, on peut positionner le bloc relativement à sa position dans le flux avec la propriété `position:relative`. Décalez le `paragraphe1` de 100px vers la droite et 20 px vers le bas.
 - 8) Décalez le deuxième paragraphe de 120px vers la droite et de 130px vers le haut.
 - 9) On peut aussi positionner le bloc par rapport au coin supérieur gauche de son premier ancêtre positionné, c'est le positionnement absolu. Modifiez le positionnement précédent des deux paragraphes en mettant `position:absolute`. En déduire quel est le premier ancêtre positionné des balises `<p>`.
 - 10) Positionnez maintenant le bloc `div` (`absolute` puis `relative`) et jouer sur sa largeur et sa hauteur.
 - 11) Placez les deux paragraphes pour un affichage sur deux colonnes dans le bloc `div`, en jouant sur les positions et les largeurs des blocs.
 - 12) Voyons maintenant le positionnement flottant qui laisse les éléments dans le flux mais permet de les pousser à gauche ou à droite. Conserver les dimensions du bloc `div` et du `paragraphe1`, supprimez celles du `paragraphe 2`. Supprimer tous les positionnements. Utilisez la propriété `float:right`, sur le style du `paragraphe1`.
 - 13) Réduisez le texte du `paragraphe1` pour ne conserver que les deux premières phrases, comment les blocs s'imbriquent-ils ? testez `float:left`.

Partie 7

Style des paragraphes

- 1) Remettre le document `Partie6.html` à son contenu initial avant la modification dans la partie 6 en gardant les identifiants `paragraphe1` et `paragraphe2`.
- 2) Supprimer tous les sélecteurs de la feuille de style `monstyle2.css`
- 3) Appliquer un fond bleu et une couleur d'écriture rouge sur la première lettre des deux paragraphes.
- 4) Appliquer un fond jaune sur la première ligne de `div`.
- 5) Supprimer le sélecteur de la question précédente et ajouter un sélecteur qui permet d'afficher « Start » sur un fond rouge est une couleur jaune avant la balise `div` si celle-ci est survolée.
- 6) Ajouter un sélecteur qui permet d'afficher « End » sur un fond rouge est une couleur jaune après la balise `div` si celle-ci est survolée.