Aide exercice 2
Théorie et technique

1. Principes de compositions CSS

Le langage CSS, a pour objectif la construction de l'interface graphique, il s'agit du design. Son emploi permet de séparer la composition de l'interface graphique et du contenu (texte / photos).

Il existe 3 modes de compositions pour ce langage :

- Embarqué
- Local
- Externe ou (feuille de style)

Dans le cadre du développement d'un site web, aucune méthode ne prime sur une autre, il s'agit juste d'opter pour une solution logique, qui peut potentiellement inclure un ou plusieurs modes de compositions selon le cas de figure.

Dans le cadre du développement d'une application mobile (web-app), le CSS doit uniquement être composé en CSS externe. Attention, nous ne parlons pas d'un site web mobile mais bien d'une application mobile.

2. le CSS externe ou feuille de style

Dans un premier temps il convient de créer le document qui permettra d'accueillir les propriétés CSS. C'est ce document que l'on nomme « feuille de style externe». Il s'agit d'un fichier vide, sans aucune écriture, qui a pour extension «.css » .

En fonction de l'environnement de développement que vous utilisez (IDE), qui n'est autre que le logiciel que vous employez pour produire du code, la procédure est généralement la même :

- fichier -> nouveau -> feuille_de style
  / ou
- Créer un nouveau fichier et l'enregistrer avec l'extension « .css »

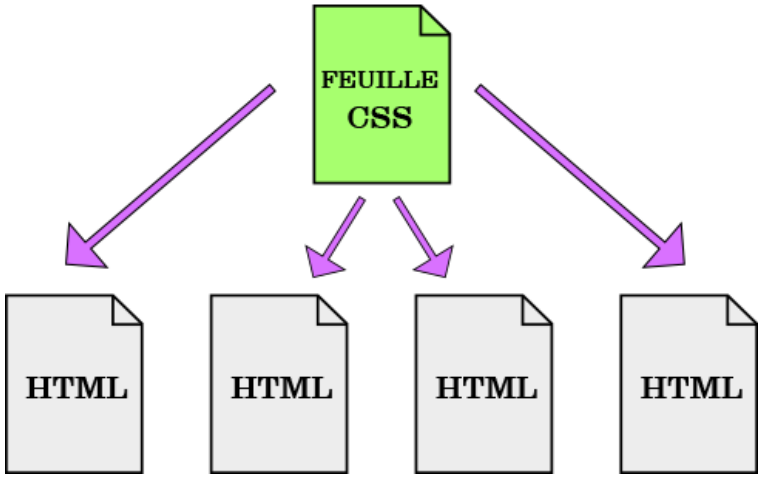
Veillez à ce que votre feuille de style se trouve dans le même dossier que votre page HTML, ou vous pouvez également la placer dans un dossier « css » lui-même situé à lintérieur du dossier de votre site web.

3. Raccorder une feuille de style à un ou plusieurs documents HTML

Le grand avantage d'une feuille de style est qu'elle va nous permettre de définir l'ensemble des styles graphiques de notre document HTML. Par exemple les niveaux de titres, les paragraphes, les fonds de pages, les proportions de la mise en page le positionnement des éléments…

Une fois ces règles définies nous pouvons alors raccorder ces définitions graphiques à un ou plusieurs documents HTML.

Tenant compte du fait que le design graphique d'un site ou d'une application est unifié, nous associerons généralement cette feuille de style à l'ensemble des documents HTML de notre site. Ainsi, si nous modifions une déclaration CSS, cette modification se répercutera sur l'ensemble des pages, ce qui nous épargne un travail considérable si nous devions procéder à cette modification en intervenant sur toutes les pages.



Effectuer le raccord, le lien, entre notre document HTML et la feuille de style, permettra au navigateur d'indiquer que notre document HTML fonctionne avec les déclarations CSS que nous avons définies dans notre feuille de style.

Pour se faire, nous devons utiliser cette ligne de code que nous placerons **dans notre document HTML entre les balises <head>**.

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

- ➔ « url » correspond à l'emplacement de la feuille de style. Le chemin à parcourir depuis notre document HTML pour accéder au document
- ➔ « stylesheet » indique au navigateur la nature du fichier qu'il charge : une feuille de style (stylesheet)
- ➔ « text/css » indique au navigateur la nature des écritures qu'il trouvera à l'intérieur du fichier. A noter que pour les navigateurs récents il n'est plus nécessaires de définir cette attribut.

4. Déclarations CSS et principes syntaxiques

Le principe d'écriture des déclarations CSS externe, est légèrement différent de la méthode embarquée (celle que nous avons employée jusqu'alors en plaçant les propriétés CSS directement à l'intérieur de la balise).
Il fait intervenir de nouveaux signes de ponctuations : « { » et « } », ainsi qu'un « sélecteur ».
Le principe d'écriture sera toujours le même, il n'y a pas d'autres alternatives.
La définition d'une règle graphique en CSS s'intitule « une déclaration ».

Il conviendra de placer ces déclaration à l'intérieur de notre feuille de style « .css » de la manière suivante :

```
/* Déclaration CSS */
Sélecteur{
    propriétéCSS : valeur ;
}
```

NB : Le signe « /* */ » permet de placer un commentaire qui ne sera pas interpréter par le navigateur

Ainsi il est possible de produire plusieurs déclarations les unes à la suite des autres avec éventuellement plusieurs prorpiétés CSS et leurs valeurs de la manière suivante :

```
/* Déclaration CSS */
Sélecteur{
    propriétéCSS : valeur;
    propriétéCSS : valeur;
}
Sélecteur{
    propriétéCSS:valeur;
}
```

NB : Peu importe les retours à la ligne ou les tabulations. L'organisation de l'écriture n'a pas d'incidence sur l'interprétation. Ce qui importe c'est le respect de la ponctuation (« { » , « } », « : », « ; »).

1. Sélecteur de balises avec nom de balises.

Dans la majorité des interfaces, il convient dans un premier temps de définir l'ensemble des propriétés CSS récurrentes. C'est-à-dire des styles CSS associées aux balises les plus fréquemment utilisées.
Ex : <body>, <h1>, <p>…, sont autant de balises couramment employées.

Pour que les propriétés CSS s'appliquent à toutes les balises d'un certain type, il suffit de mettre le nom de la balise dans le sélecteur.

Exemple :

```
p{
     color:green;
}
h1{
     color:blue;
}
/* tous les paragraphes / balises <p> / couleur « green » */
/* tous les titres niveau 1 / balises <h1> / couleur « bleu » */
```

Ainsi, une déclaration avec le sélecteur « body » permet généralement de définir l'ensemble des règles typographies du document. Famille de typo / taille / couleur / fond de page…

Exemple :

```
body{
      font-size:12px;
}
/* Par défaut, la taille du texte sera de 12px pour tout le document
HTML */
```

Autre exemple, une déclaration CSS sur « h1 » va permettre d'associer des propriétés CSS à l'ensemble des balises <h1>, tout comme une déclaration CSS sur « p » définira toutes le règles CSS associées aux paragraphes.

2. Sélecteur de balise avec une « class »

L'emploi d'un sélecteur de balise avec nom de balise (vu précédemment) peut être problématique dans la mesure où la déclaration CSS, la propriété graphique, va systématiquement s'appliquer à l'ensemble des balises qui portent ce nom. Parfois nous aurons besoin d'associer des déclarations CSS à des balises bien précises sans que cela ait une incidence sur les autres balises du même nom.

Un sélecteur de balise avec une « class » permet d'attribuer un profil CSS à une ou plusieurs balises de notre choix.

Dans un premier temps il est nécessaire de définir la ou les balises que nous aurons choisies. Pour se faire nous allons rajouter un attribut dans la balise HTML concernée (coté document HTML). Cet attribut se nomme **« class »**. C'est à vous de définir sa valeur qui est un nom de référence que vous allez utiliser plus tard dans la déclaration.

NB : Le nom de référence du sélecteur que vous allez choisir ne doit comporter, ni caractères accentués, ni espaces, ni sigles, ni caractère numérique seule. Pour compenser vous pouvez utiliser « _ » et «-»

Exemple :

```
<!-- Dans la partie <body> -->
<!-- Ajout d'une class « exemple » à une boite -->
<div class="exemple">contenu</div>

<!-- Ajout d'une class « test » à un niveau de titre 1 -->
<h1 class="test">contenu</h1>
```

NB : Il possible de rajouter une class à n'importe quel balise HTML.

Nous allons maintenant associer une déclaration CSS à la balise pour laquelle nous venons de définir une « class ».

Dans notre déclaration CSS, il va falloir maintenant définir les propriétés CSS associés à cette « class ».

Lorsque nous souhaitons faire référence à une « class », notre sélecteur commence par un « **.** » suivi directement du nom de la class que nous avons défini dans votre balise.

Exemple :

```
.exemple{
        color:blue;
}
.test{
        background:black;
}
```

Ainsi, la balise qui a pour class « exemple » aura un texte de couleur bleu. La balise qui aura pour class « test » aura un fond noir.

Il est possible d'associer une classe à plusieurs balises, qui hériteront de fait, des mêmes déclarations CSS.

Exemple :

```
<p class="NomReference">contenu</p>
<h1 class="NomReference">contenu</h1>
<!-- Le paragraphe <p> et le niveau de titre <h1> ont la même
class -->
```

3. Sélecteur de balise avec un « id »

« class » et « id » fonctionnent exactement de la même façon.
La seule chose qui les différencie est le principe d'association :
Une class peut être associée à plusieurs balises, alors qu'un identifiant ne peut être associé qu'à une seule balise du document HTML.

NB : Cette distinction a en apparence peut d'utilité, car il serait possible de définir le cas échéant, des class à usage unique. Mais, toutefois, il est d'ores et déjà important d'opérer cette différenciation car nous emploierons ce système d'identifiant lorsque travaillerons avec le langage de programmation JavaScript, qui permet de gérer les interactions utilisateurs.

Définir un identifiant nous utiliserons l'attribut « id », placé dans une balise du document HTML. Tout comme les « class » il nous appartient de définir sa valeur qui est un nom de référence que vous allez utiliser plus tard dans la déclaration.

Exemple :

```
<div id="mon_identifiant">contenu</div>
<!-- La boite <div> a un identifiant qui se nomme
« mon_identifiant » -->
```

Dans la mesure où il n'est pas possible d'associer un « id » à plusieurs balises, il serait donc erroné d'avoir le cas de figure ci-dessous :

Exemple :

```
<div id="boite1">contenu</div>
<p id="boite1">contenu</p>
<!-- La boite <div> et le paragraphe <p> ne peuvent pas avoir le même identifiant -->
```

Dans notre feuille de style, il va falloir maintenant raccorder nos propriétés CSS à la balise que nous venons de définir avec un « id ».
Lorsque nous souhaitons faire référence à un « id », notre sélecteur commence par un « # » suivi directement du nom de l' « id » que nous avons défini dans votre balise.

Exemple :

```
#mon_identifiant{
        color:blue;
}
```

4. Sélecteur combinés

Lorsque nous employons du CSS, il est préférable d'utiliser le moins possible d'identifiants et de class, par soucis d'économie d'écriture, mais également parce que les styles graphiques d'un document sont formatés. Pour se faire il est préférable de combiner les sélecteurs.
Ainsi nous pouvons par exemple agir sur un paragraphe, lui-même contenu dans une balise ayant une class ou un identifiant sans avoir à utiliser une nouvelle class ou identifiant dans la balise en question.

Exemple :

```
<div id="mon_identifiant">
    <p>paragraphe</p>
</div>
```

```
#mon_identifiant p{
        color:blue;
}
/* Cette declaration va agir sur le paragraphe contenu dans une balise qui a pour identifiant "mon_identifiant"*/
```

D'une autre manière, il est possible d'associer une déclaration CSS à plusieurs class, identifiants, balises en amployant une « , »

```
#mon_identifiant, .boite1, p{
        color:blue;
}
```

```
 /* la balise ayant pour identifiant "mon_identifiant", la
balise ayant pour class "boite1", ainsi que toutes les balises
ayant pour nom "p", auront une couleur de texte bleu */
```

5. Principes d'hérédités

CSS est l'acronyme de « Cascading Style Sheets », qui signifie « feuilles de styles en cascade ». En définitive, cette terminaison implique un principe d'hérédité et d'écrasement des propriétés CSS. Cela signifie qu'au fil de nos déclarations CSS, nous allons annuler l'effet de certaines propriétés et en rajouter d'autres.

Exemple :

```
 body{
          font-family:Verdana;
 }
 /* l'ensemble de la fonte "Verdana" sera employée sur tout le
document puisque cette déclaration CSS agit sur l'ensemble des
balises contenues dans <body> (fenêtre du navigateur)*/
 p{
          font-family:Arial;
 }
 /* Nous écrasons la fonte initiale avec la font "Verdana" pour
les paragraphes */
```