

TP5 - CSS avancé: sélecteurs et positionnement en profondeur

1 Compléments sur les sélecteurs CSS

En sus de ceux vu au TP précédent, il existe des sélecteurs supplémentaires :

1.1 Sélecteur universel

Le sélecteur `*` désigne tous les éléments **non sélectionnés**. Par exemple, la spécification suivante dans un fichier CSS :

```
* { color: yellow; }
```

fera apparaître en jaune tous les éléments dont la couleur n'est pas spécifiée.

1.2 Sélecteur par identifiant

En HTML, la valeur de l'attribut `id` d'une balise doit être **unique** dans toute la page. On pourra expressément associer des propriétés CSS à **un** élément identifié par exemple par `id="monid"` avec le sélecteur CSS `#monid`, ou `balise#monid`.

Par exemple, le sélecteur `div#header` permettra de spécifier les propriétés du bloc d'en-tête de la page, auquel on donnera l'attribut `id="header"`.

1.3 Pseudo-éléments

Ils sont identifiés par la syntaxe `balise::pseudo-élément`. Les plus courants (s'appliquent à toutes les balises pouvant contenir du texte) :

- `::first-letter` : permet de sélectionner la première lettre
- `::first-line` : permet de sélectionner la première ligne

1.4 Pseudo-classes

Elles sont identifiées par la syntaxe `balise:pseudo-classe`, et permettent de sélectionner des éléments suivant l'interaction qu'ils ont avec l'utilisateur. Les principales :

- `:hover` : activé pour un élément désigné via la souris par exemple, mais non activé.
- `:active` : activé pour un élément activé par l'utilisateur, pendant la durée du clic-souris, par exemple.
- `:link` (uniquement pour la balise `a`) actif si le lien n'a pas été visité.
- `:visited` (uniquement pour la balise `a`) actif si le lien a été visité.

1.5 Sélection descendante

Grâce aux CSS, on peut assigner des propriétés à certaines balises uniquement si celles-ci sont filles d'autres balises, ou insérées dans d'autres balises. Par exemple, le sélecteur `p` désigne toutes les balises `p` par défaut, mais on veut parfois appliquer un formatage spécifique aux paragraphes qui sont insérées dans un bloc donné, ou dans une autre structure (un tableau par exemple).

On parle de sélecteurs **descendants** et de sélecteurs **enfants**.

- Sélecteurs descendants : la syntaxe `table p { ... }` désignera tous les éléments `p` qui sont intégrés à un tableau.
- Sélecteurs enfants : la syntaxe `div>p { ... }` désignera tous les éléments `p` qui sont **directement** insérées dans une balise `div`.

Exemple pratique : la balise `em` permet de donner de l'emphase à une partie de texte. On peut souhaiter la mettre en rouge, on définira alors

```
em {color: red;}
```

Mais si on met également les titres en rouge (avec `h2 {color: red;}`), alors le code HTML suivant sera tout en rouge, perdant l'effet d'emphase.

```
<h2>Ceci est <em>très</em> important</h2>
```

On résoudra ce problème en spécifiant que la balise `em` insérée dans un titre `h2` devra apparaître en bleu : `h2 em {color: blue;}`

On peut aussi cibler des identifiants et classes précises. Par exemple, le sélecteur `table#tintin p.milou` ne concernera que les paragraphes de la classe `milou` à l'intérieur de la table ayant l'identifiant `tintin`.

2 Exercices d'application

2.1 Exercice 1

Reprendre le diaporama du TP4 et ajouter le code nécessaire dans la feuille de style pour que :

- les liens visités apparaissent en jaune,
- les liens non-visités en gras, sans soulignement,
- le passage de la souris sur un lien le fasse apparaître en vert.

Il faudra tout d'abord affecter un style par défaut à la balise `a`, puis spécifier les différences avec les pseudo-classes correspondantes.

2.2 Exercice 2

Pour une page web d'un système d'information, on souhaite afficher une boîte de type "message d'erreur". Saisir dans un fichier **boite.1.html** le balisage html suivant, **sans modification**, et en ajoutant le squelette html nécessaire :

```
<h2>Titre</h2>
<p>Un court texte de description.</p>
<div class="alert">
<h2>Message d'erreur</h2>
<p>La connexion au serveur a été interrompue, vous devez
réinitialiser votre session de travail.</p>
</div>
```

- La boîte aura une largeur de 400 px,
- Le premier titre (h2) doit être en Arial, 20pt, en vert,
- le premier paragraphe doit être en Times New Roman, 12pt, de couleur #555,
- le titre dans la boîte doit être en rouge, avec la police "Comic Sans MS" en taille 14 pt, et le texte en Verdana, 10 pt, noir. Le fond de la boîte doit être en gris clair (20%).

Définir les styles correspondants pour obtenir le rendu ci-dessous :



Vous trouverez l'image associée (**Alert.png**) dans le dossier **P:\M1106\img**. Il faut l'associer comme image de fond à la boîte. Les styles de la boîte doivent être définis de façon descendante. Ici, il faudra définir un style pour le paragraphe inclus dans la boîte avec le sélecteur enfant **div.alert > p { ... }**

2.3 Exercice 3

Dans un fichier **boite.2.html**, saisir le code suivant (entre les balises "body") :

```
<div class="d1"></div>
```

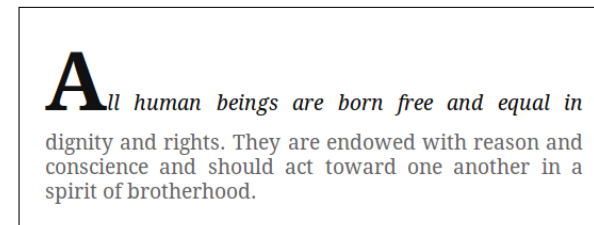
Définir dans la feuille de style la classe **d1** telle que on ait sur la page :

- un carré rouge de 200 pixel de coté,

- qui s'élargit à 400 pixels et devient vert avec une bordure jaune de 5 pixels d'épaisseur quand on passe la souris dessus,
- et qui se transforme en un rectangle bleu de 800 pixels avec une bordure noire de 2 pixels d'épaisseur quand on clique dessus.

2.4 Exercice 4

Reprendre le source du TP3 sur l'article 1 de la déclaration des droits de l'homme et lui assigner un nouveau style qui donnera le rendu suivant :



Ceci ce fera **sans** modifier le balisage HTML et en utilisant les pseudo-éléments **first-letter** et **first-line**. Modifier la taille de la fenêtre du navigateur, et vérifier que l'adaptation à la largeur se passe bien.

3 Positionnement en profondeur

Jusqu'à présent, une page est considérée comme un plan 2-dimensions. Mais on peut jouer sur l'ordre dans lequel les éléments sont affichés et positionner des éléments devant d'autres via la propriété **z-index** qui se verra assigner une valeur numérique : plus elle est élevée, plus l'élément sera au premier plan.

Exercice : Dans un fichier **profondeur.html**, saisir le code suivant (entre les balises "body") :

```
<div id="d1"></div>
<div id="d2"></div>
<div id="d3"></div>
```

Dans la feuille de style associée, définir les styles ci-dessus de façon à ce que les 3 blocs soit représentés sous la forme de 3 carrés de 300 pixels de cotés, de couleur rouge, vert, bleu, et positionnés en absolu et respectivement à 50, 100, et 150 pixels du bord de l'image via les propriétés CSS suivantes :

```
{ position: absolute; top: 50px; right: 50px; }
```

Une fois l'affichage vérifié, associer à la **div d2** une propriété **z-index: 5;**. Observer le résultat. Ajouter ensuite **z-index: 10;** à **d1** et observez le résultat.