

Les feuilles de styles - Partie 1

Introduction

- Une archive contenant les fichiers mentionnés dans cette fiche est à récupérer sur la liste de diffusion.
- Il est recommandé de travailler avec FireFox (notamment, Chrome ne supporte plus MathML en HTML5).
- Vous trouverez une référence pour CSS à cette adresse : <http://www.w3schools.com/css/>
- Pour le reste, voir vos notes...

Exercice 1 – Principes de CSS

Question 1.1 : Ouvrez le fichier `ex1.html` avec un éditeur de texte et dessinez l'arbre de l'application HTML. (Il est recommandé d'annoter les nœuds de l'arbre avec la valeur de leur attribut identifiant)

Question 1.2 : Ouvrez le fichier `ex1.css` et décrivez les effets de l'unique règle décrite. Vérifiez le résultat dans le navigateur.

Question 1.3 : Dans le fichier `ex1.css`, renommez le sélecteur `test1` en `test2` et actualisez l'affichage dans le navigateur. Décrivez et expliquez le changement observé. Répétez l'opération jusqu'au nom `test10`.

Question 1.4 : Précisez, pour chacune des expériences précédentes, combien de nœuds sont sélectionnés dans l'arbre et combien d'éléments HTML sont affectés ? Comment s'appelle ce phénomène ? S'applique-t-il à tous les attributs de style ?

Question 1.5 : Dans `ex1.html`, remplacez `ex1.css` par `ex1b.css`. Donnez la valeur des propriétés de style (couleur de tracé, de fond, style de bordure et police) de l'élément paragraphe d'identifiant `test3`. Quel est le mécanisme mis en oeuvre ici ?

Exercice 2 – Sans tricher !

Question 2.1 : Toujours en considérant le document `ex1.html`, pouvez-vous prédire le style que recevront les éléments paragraphe pour les fichiers CSS suivants ? Justifiez votre prédiction pour chacun. Cela ferait-il une différence si l'ordre des instructions était inversé ?

```
/* Fight1.css */
p {
    background:lightgray;
    color:black;
    border-style: solid;
    border-width: 1px;
    font-family:"Courier New", Times, serif;
}
```

```
#test3 {
    background:white;
    color:blue;
    font-weight: bold;
}
```

```
/* Fight2.css */
p {
    background:lightgray;
    color:black;
    border-style: solid;
    border-width: 1px;
    font-family:"Courier New", Times, serif;
}
```

```
p {
    background:white;
    color:blue;
    font-weight: bold;
}
```

```

/* Fight3.css */
div p {
    background:lightgray;
    color:black;
    border-style: solid;
    border-width: 1px;
    font-family:"Courier New", Times, serif;
}

p {
    background:white;
    color:blue;
    font-weight: bold;
}

```

Question 2.2 : Quelle modification peut-on appliquer au fichier `fight1.css` pour que tous les paragraphes soient sur fond gris ?

Exercice 3 – CSS et les extensions de HTML¹

Récupérez le fichier `mathml.html` depuis l'archive fournie et répondez aux questions suivantes en créant un fichier `mathml.css`. (Il est conseillé d'archiver vos résultats intermédiaires)

Question 3.1 : Faites un croquis rapide de l'arbre syntaxique de ce document.

Question 3.2 : Faites en sorte que les opérateurs de somme de cette formule s'affichent en couleur verte

Question 3.3 : Faites en sorte que les opérateurs de racine s'affichent en couleur bleu, mais que le contenu de ces racines reste dans la couleur par défaut (Indice : l'attribut `color` peut, comme beaucoup d'autre, prendre deux valeurs spéciales. Quelles sont-elles).

Question 3.4 : Dynamisons cela un peu et faisons en sorte que la formule entière, sauf les opérateurs colorés, passe en couleur rouge lorsque le curseur de la souris la survole.

Récupérez maintenant le fichier `svg.html` à partir de l'archive et essayons de donner un peu de forme à cette tâche noire avec style. Attention, si les attributs de mise en forme de MathML correspondent essentiellement à ceux de HTML, SVG dispose de ses propres attributs.

Question 3.5 : Examinez le code svg fourni : inutile cette fois de dessiner l'arbre syntaxique, mais recherchez les éléments auxquels des classes ont été assignées.

Question 3.6 : Appliquez la couleur de remplissage (`fill`) jaune aux classes `feet` et `beak`, la couleur blanche à la classe `belly`, les couleurs en composantes RVB (2b, 2b, 31) à la classe `arms` et (3f, 47, 53) aux classes `.body` et `.head`.

Question 3.7 : Appliquez la couleur noire au tracé de la classe `feet`.

Animons un peu là aussi :

Question 3.8 : Faites en sorte que le bec devienne rouge lorsque la souris passe dessus.

Vous constaterez qu'un effet `BlurFilter` est déclaré dans le flux SVG. Un effet s'applique grâce à l'attribut `filter` par son url (avec la notation fonctionnelle `url(...)`). Ici, vu que l'effet est en ligne dans le code, il sera référence par son nom sous forme de fragment (d'url).

Question 3.9 : Appliquez l'effet en question sur le ventre lorsque le curseur de la souris passe dessus.