

TD/TP n° 2

Exercice 1.

On considère le code XML suivant :

```
<racine>
  <branche taille="gros">A
    <brindille>AA
      <fruit>Banane</fruit>
      <fruit>Kaki</fruit>
    </brindille>
    <branche>AB
      <brindille>ABA
        <fruit>Pomme</fruit>
        <fruit>Poire</fruit>
      </brindille>
      <brindille>ABB
        <fruit>Coing</fruit>
        <fruit>Prune</fruit>
      </brindille>
    </branche>
    <branche>AC
      <brindille>ACA
        <fruit>Pêche</fruit>
      </brindille>
    </branche>
  </branche>
  <branche taille="petit">B
    <brindille>BA
      <fruit>Nectarine</fruit>
    </brindille>
  </branche>
</racine>
```

Pour chacun des codes CSS suivant, déterminer les éléments qui seront affichés en rouge.

1. `branche:first-child branche { color:red; }`
2. `branche[taille] > brindille { color:red; }`
3. `branche[taille] brindille { color:red; }`
4. `fruit:first-child + fruit { color:red }`
5. `branche { color:red; }
branche > branche { color:black; }`

Exercice 2.

On considère les arbres XML validés par la DTD suivante :

```
<!DOCTYPE toto [
  <!ELEMENT toto (tata,titi)+>
  <!ELEMENT tata (titi?,tata*)>
  <!ELEMENT titi (titi|tutu)*>
  <!ELEMENT tutu (#PCDATA)>
]>
```

1. Construire quelques arbres XML correspondant à cette DTD.
2. Comment faire en CSS pour :
 - a) Mettre en gras le contenu des éléments `titi`, seulement s'ils sont fils d'un `tata`.
 - b) Mettre en vert le contenu du premier élément `titi` qui soit fils direct de `toto`.
 - c) Mettre en police 15pt le contenu des éléments `tutu` qui sont précédés par un élément `titi`.
 - d) Masquer tous les éléments `tata` hormis ceux accrochés directement à la racine `toto`.
 - e) Afficher chaque élément `tutu` entre `tu>` et `<tu`.

Exercice 3. TP

Récupérer (par `wget`) les fichiers `trains.xml` et `trains.png` :

```
wget http://www.lacl.fr/~msassolas/enseignement/XML_Licence/trains.xml
wget http://www.lacl.fr/~msassolas/enseignement/XML_Licence/trains.png
```

1. Construire le fichier `trains.dtd` qui accepte les arbres tels que celui dans `trains.xml`.
2. Construire le fichier `train.css` pour reproduire le résultat fourni dans `trains.png`.

Si vous voulez exactement les paramètres utilisés :

```
Couleur de fond : #412D8D
Taille du titre : 30pt
Taille des noms de gare, etc : 15pt
Taille des infos de correspondance : 12pt
Taille de l'indentation entre les puces rondes et carrées : 25px
Marge en haut et en bas : 10px
Marge sur les cotés : 25px
```

3. Construire un nouveau fichier `trainsTab.css` pour présenter les informations du XML en tableau. Vous ne devez pas changer le XML (hormis l'URI de la nouvelle CSS).

Exercice 4. TP

Récupérer (si vous l'avez perdu depuis la semaine dernière) le fichier `courses-ID.xml` :

```
wget http://www.lacl.fr/~msassolas/enseignement/courses-ID.xml
```

Présenter à l'aide d'une CSS le contenu du fichier de manière lisible et cohérente. Mettre visuellement en évidence les cours qui ont des prérequis (« Prerequisites »). (Bien penser à ajouter le lien vers la CSS dans le fichier!)

Exercice 5. TP

Récupérer le fichier `recipe.xml` :

```
wget http://www.lacl.fr/~msassolas/enseignement/recipe.xml
```

1. Construire une DTD `recipe.dtd` pour ce fichier.
2. Construire un style CSS `recipe.css` pour ce fichier. On pourra par exemple mettre la liste des ingrédients dans un joli encadré.

Rappels

— Validation du XML vis-à-vis d'une DTD référée dans le fichier :

```
xmllint --noout --valid monFichier.xml
```

— Validation du XML vis-à-vis d'une DTD externe :

```
xmllint --noout --dtdvalid maDTD.dtd monFichier.xml
```

— Ressources pour le CSS (toutes les possibilités, dont si peu ont été vues en cours) :

```
http://www.w3schools.com/cssref/
```