

## Exercice 1

Reprenez votre fichier `index.html` du premier TP, et choisissez une balise sur laquelle expérimenter avec le padding, le margin et les border (taille, couleur... soyez créatifs).

## Exercice 2 : jouer avec l'héritage

Créez une page HTML avec la liste imbriquées suivants :

```
<ul class="liste-principale">
  <li> un </li>
  <li> deux </li>
  <ul>
    <li> 1 </li>
    <li> 2 </li>
  </ul>
  <li> trois </li>
  <ul class="sous-liste">
    <li> 1
      <ul>
        <li> oui </li>
        <li> non </li>
      </ul>
    </li>
    <li> 2 </li>
  </ul>
</ul>
```

1. Créez un fichier CSS qui donne un style différent à chacune des deux classes qu'on retrouve dans cette page. Essayez d'appliquer des styles sur plusieurs propriété (modification sur le texte, sur la bordure, sur le padding...).
2. Est-ce que toutes les règles se propagent aux enfants du sélecteur sur qui on applique une règle ?
3. Comment connaître comment se propage une règle ?
4. Est-ce qu'il existe un moyen de forcer l'héritage ? Si oui, lequel ?

Fichier à créer : `heritage.css`

## Exercice 3 : les pseudos-classes

On a vu rapidement les pseudo-classes dans la partie cours en donnant pour exemple la pseudo-classe `:first-child` qui permet de n'appliquer un style qu'au premier enfant d'un sélecteur.

1. Cherchez dans la documentation d'autres pseudo-classes et décrivez à quoi ils servent (on exclut `:hover` et `:focus` dans cet exercice).
2. Créez une liste, et expérimentez avec ces nouvelles pseudo-classes que vous venez de découvrir.

Fichiers à créer : `test-pseudoclass.html` et `pseudoclass-style.css`

## Exercice 4 : menu déroulant partie 1

Soit la liste suivante :

```
<ul>
  <li><a href="">Accueil</a></li>
  <li><a href="">News</a></li>
  <li><a href="">Photos</a>
    <ul>
      <li><a href="">Couleur</a>
        <ul>
          <li><a href="">Normandie</a></li>
          <li><a href="">Paris</a></li>
        </ul>
      </li>
      <li><a href="">Noir et Blanc</a>
        <ul>
          <li><a href="">Portraits</a></li>
          <li><a href="">Paysages</a></li>
        </ul>
      </li>
    </ul>
  </li>
  <li><a href="">Liens</a></li>
  <li><a href="">Contact</a></li>
</ul>
```

Pour empêcher l'affichage d'une balise et de son contenu, il suffit que la propriété CSS `display: none` lui soit appliquée. Pour la faire réapparaître, il faut lui appliquer, par exemple, `display: block`. Cette application est provoquée par la déclaration d'une règle indiquant cette propriété pour un sélecteur désignant cette balise.

Trouver les règles CSS affichant les items de liste de premier niveau mais pas les autres, ceux-ci n'apparaissant que si la souris survole l'item de niveau supérieur dont ils dépendent. On utilisera des sélecteurs composés exclusivement des noms de balises `<ul>` et `<li>`, ainsi que de la pseudo-classe `:hover` : le sélecteur `li:hover` désigne, si elle existe, la balise `<li>` survolée par la souris.

Fichier à créer : `deroule-hover.css`

## ## Exercice 4 : menu déroulant partie 2

On continue sur le même fichier HTML que l'exercice précédent.

On souhaite à présent que chaque item apparaisse dans un rectangle, et que l'éventuelle sous-liste `ul` dépendant d'un item apparaisse non plus en dessous de lui, mais à sa droite au même niveau de départ. Plus précisément :

- chaque liste `ul` aura une largeur de 10em, son padding sera zéro et les puces habituellement présentes ne le seront pas ;
- chaque élément n'ayant qu'une ligne, on spécifiera la hauteur du rectangle via la propriété `line-height`, fixée à 2em ;
- chaque item sera entouré d'une bordure :
  - en ligne pleine, d'épaisseur de 0.1em, grâce à la propriété `border-width`, qui indique 4 épaisseurs dans cet ordre : haut, droit bas, gauche (trouver comment éviter qu'une ligne séparant deux items ait une épaisseur double) ;
- pour qu'une liste de niveau inférieur se place à côté de l'item dont elle dépend, on mettra leur propriété position à `absolute` et on donnera aux propriétés `margin-left` et `margin-top` des valeurs (elles peuvent être négatives) découlant des dimensions données ci-dessus au rectangle mais aussi aux bordures.

On vérifiera qu'au grossissement de la page, les blocs restent lisibles et adjacents (c'est tout l'intérêt d'utiliser l'unité de mesure `em`, plutôt que le pixel ou le point).

Fichier à créer : `deroule-position.css`

## Exercice 4 : menu déroulant partie 3

Pour finir avec ce menu déroulant, on va ajouter quelques éléments pour améliorer la présentation :

- les rectangles auront une couleur de fond ;
- au survol d'un lien hypertextuel par la souris, les lettres passeront du noir au blanc, et le fond passera à une couleur sombre ;

- on enlève le souligné sur les liens hypertextuels ;
- les items de liste seront au centre de leur rectangle.