Les sélecteurs

Sélecteurs de type

```
a{
    /*la règle s'applique à tous les éléments a*/
}

h1, h2{
    /*la règle s'applique à tous les éléments h1 et h2*/
}
```

Appliquer une mise en forme différente à deux éléments ayant le même nom

1° Sélecteurs de classe

```
.texte{
    /*la règle s'applique à tous les éléments qui ont l'attribut class="texte". */
}
```

L'attribut class permet de classer, catégoriser les éléments, notamment en vue de leur mise en forme :

```
div class="menu"
div class="actu"
p class="chapo"
```

Un élément peut avoir plusieurs classes. Dans ce cas, il a un attribut `class`, avec plusieurs valeurs séparées par des espaces

```
div class="menu top"
```

→ Le menu peut être stylé avec des règles

```
.menu{
  background-color:grey;
}
```

qui s'appliquera aussi au menu du bas ou du côté

et une règle

```
.top{
  width:1000px;
}
```

qui ne s'appliquera qu'à ce menu en particulier

Appliquer une mise en forme différente à deux éléments ayant le même nom
2° Sélecteurs descendants

```
h1 a{
    /*La règle s'applique à tous les éléments a contenus dans un élément h1 (distinguer les liens
    inclus dans un titre des autres liens*/
}
```

Combinaison d'un sélecteur de classe et d'un sélecteur descendant

```
.menu a{
   /*la règle s'applique à tous les éléments a contenus dans un élément qui a l'attribut
   class="menu" (permet par exemple de mettre en forme de manière différente les liens du
   menu et les autres)*/
}
```

Autres combinaisons

```
div.texte{
    /*la règle s'applique à tous les éléments div qui ont l'attribut class="texte"*/
}

.menu.top{
    /*la règle s'applique à tous les éléments div qui ont l'attribut class="menu top"*/
}


#top{
    /*sélecteur d'identifiant : la règle s'applique à l'élément (il ne doit y en avoir qu'un !) qui a
    l'attribut id="top"*/
}
```

Pseudo-classes

```
a:hover{

    /*
    la règle s'applique aux éléments a quand ils sont survolés par la souris. C'est une pseudo-classe :
    :hover → au survol
    :active → une fois cliqué
    :visited → lien déjà visité
    */

}
```

Règles de préséance

À l'intérieur d'une feuille de style, les dernières règles l'emporte sur les premières

```
p{
   color:red;
}
p{
   color:blue;
}
```

→ sera bleu

Mais les règles les plus *spécifiques* l'emportent même sur celles qui les suivent

– un sélecteur de classe l'emporte sur un sélecteur de type

– hiérarchie complète des sélecteurs dans la « cascade » de règles : https://developer.mozilla.org/fr/docs/Apprendre/CSS/Introduction_%C3%A0_CSS/La_cascade_et_l_h%C3%A9ritage

```
.red{
   color:red;
}
p{
   color:blue;
   border-style:solid;
}
```

→ un élément `p` possédant l'attribut `class=red` restera rouge.

→ mais comme tous les autres `p`, il possèdera une bordure pleine (cette règle n'entre pas en contradiction avec une règle plus spécifique)

On peut lier plusieurs feuilles de style à un même fichier HTML

```
<link rel="stylesheet" href="style.css"/>
<link rel="stylesheet" href="perso.css"/>
```

Les règles de la deuxième feuille supplantent celles de la première, sauf si celles-ci sont plus spécifiques

L'arme ultime

```
p{
   color:blue !important;
}
```

!important indique que la règle doit supplanter toutes les autres (précédentes et suivantes), sauf si elles ont elles aussi cette indication.

Ne pas en abuser, à employer en dernier recours.

Les valeurs relatives

Possibilité d'exprimer les valeurs de *largeur* (pour `width`, `margin`, `padding`)

– de façon absolue, en px

– de façon relative, en % de la largeur de l'élément parent → comme la largeur de l'élément `body` est par défaut égale à la largeur de la fenêtre, la largeur des éléments s'adaptera en fonction de la largeur de l'écran ou de la fenêtre.

HTML :

```
<body>
    <div></div>
</body>
```

CSS :

```
div{
    background-color:red;
    width:50%;
    height:500px;/*la hauteur ne peut pas s'exprimer en %*/
}
```

Voir la différence si on ajoute :

```
body{
   width:50%;/*voir la différence avec une valeur absolue en redimensionnant la
   fenêtre*/
   background-color:blue;
}
```

On peut aussi exprimer une propriété `margin` ou `padding` en % : cette marge (qu'il s'agisse d'une marge supérieure, inférieure, gauche ou droite) sera alors proportionnelle à la *largeur* de l'élément parent (et non à sa hauteur, même pour une marge supérieure ou inférieure !)
Les dimensions relatives ne fonctionnent pas pour les cellules d'un tableau (voir au besoin la propriété `table-layout`)

Centrer horizontalement un élément de type block

HTML :

```
<body>
   <h1>Titre</h1>
</body>
```

CSS :

```
h1{
   background-color:red;
   width:50%;
   margin:0 auto;/*0 pour haut et bas, auto pour droite et gauche*/
}
```

- *Rappel : un élément block occupe toute la largeur de son parent*
- *Rappel : l'élément body occupe toute la largeur de la fenêtre*
- Pour appliquer la propriété `margin:0 auto;` à un élément il faut avoir défini la largeur de cet élément
- La différence entre cette largeur et la largeur de son parent sera répartie sous forme de marge de manière égale à droite et à gauche → l'élément sera donc centré
- Pour centrer du texte ou un élément *inline* à l'intérieur d'un élément block, comme « Titre » à l'intérieur de `h1` (et non un élément block à l'intérieur d'un autre, comme `h1` à l'intérieur de `body`), utiliser la propriété `text-align:center;`

Les images

- Attention : l'élément `img` est de type inline (par défaut, elle se comporte comme un flux de texte)
- On ne peut pas centrer une image avec la règle `margin:auto;` qui ne s'applique qu'à un élément de type block.
- Mais on peut tout de même lui appliquer la propriété `width` pour régler sa largeur.
- Pour manipuler une image (la centrer, par exemple), il faut l'inclure dans un élément conteneur `div` auquel on donne une classe pour la distinguer des autres `div` (`class="container"`, par exemple)
- On peut ensuite appliquer `text-align:center;` à l'élément `div` qui contient une image (pas à l'image elle-même)
- Pour donner à une image de fond la même taille que l'élément auquel elle sert de fond, écrire :

```
body{
   background-image:url("image.jpg");
   background-size:100%;
}
```

La propriété `background-size` peut recevoir une valeur en %, relative à la largeur de l'élément auquel s'applique la propriété.

Méthode

Travailler en équipe

Par étapes (solution professionnelle)

(1) Une personne dessine la maquette des pages (graphiste)
(2) Une personne code le HTML
(3) Une personne code le CSS (intégrateur)

Simultanément

(1) Chacun code un fichier HTML + un personne code dans un fichier temporaire les éléments qui apparaîtront sur toutes les pages (header et footer)
(2) On copie les éléments header et footer sur toutes pages créées
(3) On code le CSS
– Une personne crée un fichier CSS pour les propriétés avec héritage
– Une personne crée un fichier CSS pour les dimensions
– Une personne crée un fichier CSS pour les marges
Chacun vérifie que le CSS s'applique bien à sa copie des fichiers HTML (interdiction de toucher au code HTML pendant ce temps)
(4) On fusionne les fichiers CSS en un seul
Utiliser les outils de partage pour synchroniser le dossier de travail (Dropbox, Google, etc.)

Dissocier trois étapes

- Régler les propriétés dont les enfants héritent + les bordures
- Régler les dimensions (largeur, hauteur)
- Régler margin et padding

Régler les propriétés dont les enfants héritent

Pour être le plus bref possible

– Si une règle s'applique à la majorité des éléments contenus dans un autre, appliquer la règle à ce dernier (ses enfants en hériteront)

```
body{
   font-family:Arial;
   /*police majoritairement utilisée : pour les h1, h2, p, les li, les td qui héritent tous de body; j'écrirai une règle spécifique pour les titres h3*/
}
```

– Si une même règle s'applique à deux éléments, les regrouper

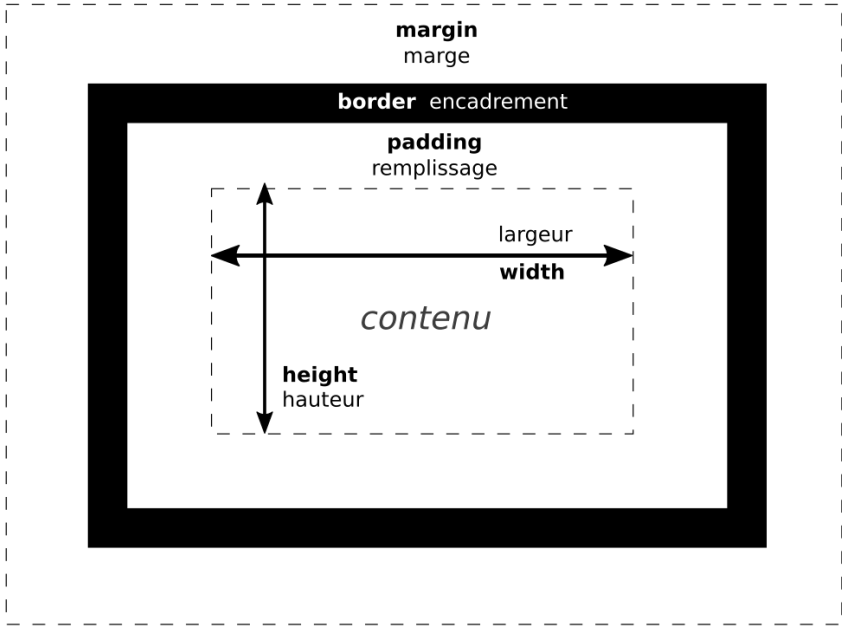
```
h1, h2, h3{
   font-family:Georgia;/*je regroupe ici h1, h2, h3, qui partagent une règle*/
}
h2{
   border: 10px solid silver;/*en revanche, cette règle est spécifique à h2*/
}
```

→ Plus économique que d'écrire

```
h1{
   font-family:Georgia;
}
h2{
   font-family:Georgia;
   border: 10px solid silver;
}
h3{
   font-family:Georgia;
}
```

→ Si je décide de remplacer Georgia par une autre police, je voudrai certainement que ce changement s'applique d'un seul coup à h1, h2 et h3.

Régler margin et padding



- Margin : espace entre la bordure d'un élément et
  - les éléments voisins
  - son élément parent (dans lequel il est contenu)
- Padding : espace entre la bordure d'un élément et
  - ses enfants (son contenu)

Outils

– Utiliser l'explorateur de document

– Rajouter couleur et bordure aux éléments pour visualiser (la règle `*{border-style:solid;}` applique une bordure à tous les éléments, * fonctionne comme un sélecteur joker valant tout élément)

– Indenter le code CSS

On part de l'élément le plus englobant, **`body`**

1° Les éléments contenus dans le `body` sont-ils tous à la même distance des bords de la fenêtre ?

Oui

→ `padding` du `body` = cette distance

Non

→ `padding` du `body` = plus petite distance (distance entre le bord de la fenêtre et les éléments les plus proches du bord de la fenêtre)

→ ajouter une propriété `margin` (marge extérieure) aux éléments qui sont plus éloignés du bord

2° Quel est l'espacement haut et bas entre les éléments contenus dans `body` ?

→ régler la propriété `margin-top` et `margin-bottom` pour chaque type d'élément

Pour le premier et le dernier éléments, cette propriété peut vous obliger à revoir le réglage du padding haut et bas du conteneur.

Attention, les éléments `p`, `h1`, `ul`, `li`, etc. peuvent avoir par défaut des propriétés `margin` et `padding` qui ne sont pas réglées à 0 !

On descend au niveau inférieur (éléments **`div`** ou **`header`**, **`section`**, etc. contenus dans **`body`**)

1° Les éléments (autre `div`, `p`, `h1`, `table`, etc.) contenus dans les `div`, `header`, `section` de premier niveau sont-ils tous à la même distance du bord de ces `div`, `header`, `section` ?

Oui

→ `padding` des `div`, `header`, `section` = cette distance

Non

→ `padding` des `div`, `header`, `section` = plus petite distance (distance entre le bord des `div`, `header`, `section` et les éléments les plus proches de ce bord)

→ ajouter une propriété `margin` (marge extérieure) aux éléments qui sont plus éloignés du bord des `div`, `header`, `section`.

2° Quel est l'espacement haut et bas entre les éléments contenus dans les `div`, `header`, `section` *?*

→ régler la propriété `margin-top` et `margin-bottom` pour chaque type d'élément

Pour le premier et le dernier éléments, cette propriété peut vous obliger à revoir le réglage du padding haut et bas du conteneur)

Attention, les éléments `p`, `h1`, `ul`, `li`, etc. peuvent avoir par défaut des propriétés `margin` et `padding` qui ne sont pas réglées à 0 !

On descend au niveau inférieur (`h1`, `div` contenues dans une `div` de premier niveau, etc.)

1° Les éléments (autre `div`, `p`, `h1`, `table`, etc.) contenus dans les `div` de premier niveau sont-ils tous à la même distance du bord des div ?

…

2° Quel est l'espacement haut et bas entre les éléments contenus dans les `div` ?

…

Et ainsi de suite…