

Tutoriel simplifié pour CSS

UCBL - Département Informatique de Lyon 1 – BDW - automne 2024

Ce tutoriel reprend le tutoriel HTML du cours BDW.

1. Pour la mise en page et mise en forme de notre page web, nous créons un fichier `tutoriel.css` dans lequel nous mettrons nos styles CSS.

Dans notre page web (`page.html`), il faut **attacher le fichier CSS à la page HTML** grâce à la balise `<link>`. Comme il n'y a pas encore de style, la page n'a pas changé.

Les étapes suivantes modifient essentiellement le fichier `tutoriel.css`. Ce sont les **sélecteurs CSS** qui déterminent à quels éléments de la page HTML s'applique un style.

```
<link rel="stylesheet"
↳ type="text/css"
↳ href="tutoriel.css">
```



2. Commençons par mettre des **couleurs de fond** (propriété `background-color`). Les couleurs s'expriment par leur nom (anglais) ou leur code RGB. Voici une **liste des couleurs CSS**.

Pour avoir une page avec un fond blanc^a, on utilise un **sélecteur de balise** (`body`) et on renseigne la propriété de couleur de fond entre accolades.

Personnalisons de la même manière le titre principal (balise `<h1>`) avec un fond bleu ciel (code `#AACCFF`).

^aLa majorité des navigateurs utilisent par défaut une page blanche, donc aucun changement visible.

```
body {
  background-color: white;
}

h1 {
  background-color: #AACCFF;
}
```

Tutoriel BDW

3. Un autre sélecteur est basé sur un **nom de classe**. On précède le nom de la classe par un point. La propriété `color` permet de **colorier un texte**, ici en violet.

Désormais, toutes les balises avec la classe *violet* ont un texte en violet. Le sélecteur de classe permet d'appliquer un style à des balises différentes, comme ici sur `` et sur ``.

```
.violet {
  color: darkviolet;
}
```

```
<p>L'objectif de ce tutoriel est
↳ d'apprendre le <strong
↳ class="violet">HTML</strong> :
```

```
<li class="violet">balises de
↳ contenu</li>
```

L'objectif de ce tutoriel est d'apprendre le **HTML** :

- balises structurantes
- balises de contenu
- balises de formulaire

4. Imaginons que nous voulons mettre le mot *structurantes* en **gras et en police plus grande**. Pour appliquer le style uniquement à ce mot, nous allons l'entourer d'une balise `` avec un identifiant.

Dans le CSS, on utilise un **sélecteur basé sur l'identifiant** (valeur de l'attribut `id` précédée par un `#`). Le style se compose de deux propriétés, `font-size` (taille) et `font-weight` (gras).

```
<li>balises <span  
↳ id="id-structurantes">structurantes  
↳ </span></li>
```

• balises **structurantes**

```
#id-structurantes {  
  font-size : 1.5em ;  
  font-weight : bold ;  
}
```

5. Étudions maintenant l'**alignement de contenu** avec la propriété `text-align`.

On complète le style appliquée à la balise `<h1>` pour centrer le titre principal, et les paragraphes qui sont premiers enfants de leur balise parente (sélecteur `p:first-child`) s'alignent à droite.

```
h1 {  
  background-color: #AACCF; ;  
  text-align: center ;  
}  
  
p:first-child {  
  text-align: right ;  
}
```

Tutoriel BDW

L'objectif de ce tutoriel est d'apprendre le [HTML](#) :

- balises **structurantes**
- balises de contenu
- balises de formulaire

6. Améliorons le rendu du **tableau**.

La propriété `border-collapse` avec la valeur `collapse` permet de fusionner les bordures adjacentes.

Pour les cellules du tableau, on utilise deux **sélecteurs de descendance** pour appliquer le style aux balises `<th>` et `<td>` qui sont des descendantes d'une balise `<table>`. Le style spécifie une **bordure** noire, à trait plein de 1 pixel d'épaisseur. Pour que la bordure ne soit pas collée au contenu, une marge interne (`padding`) est ajoutée.

```
table {  
  border-collapse: collapse ;  
}  
  
table th, table td {  
  border: 1px solid black ;  
  padding: 0.5em ;  
}
```

PHP	JavaScript	Java	Python	Autre
10	20	30	40	50

7. Un exemple de **sélecteurs avec pseudo-classe**, qui prend en compte l'état d'un élément (e.g., *survolé* ou *sélectionné*).

Quand un lien est survolé (`hover`), il n'est pas souligné et s'affiche sur fond jaune.

```
a:hover {  
  text-decoration: none ;  
  background-color: yellow ;  
}
```

[Crédits image](#)

[Agrandir l'image](#)

8. Pour espacer les éléments, on utilise les **marges** internes (**padding**) et extérieures (**margin**).

Un **sélecteur conditionnel** permet de sélectionner les `<input>` qui ont un attribut `type` avec la valeur `submit`.

Une marge extérieure à gauche est ajoutée au bouton *Envoyer*, ce qui le décale vers la droite. La marge interne ajoute un espacement entre son contenu (texte) et sa bordure.

```
input[type="submit"] {
  margin-left: 3em;
  padding: 1em;
}
```

Quel(s) framework(s) aimeriez-vous apprendre ?
 Flask Django Laravel

Envoyer

9. L'**agencement des éléments** repose avant tout sur le flux (ordre des balises), leur dimensions et marges, et le type de rendu. Pour agencer un élément de façon *responsive*, on utilise **flexbox**.

On transforme le pied de page en élément flex (`display: flex`), et on précise que ses éléments enfants auront des espacements autour et entre eux (`justify-content: space-around`). La taille de ces espacement est automatiquement calculée. Notez l'ajout d'une marge extérieure en haut pour "détacher" le pied de page du tableau.

```
footer {
  margin-top: 1em;
  display: flex;
  justify-content: space-around;
}
```

PHP	JavaScript	Java	Python	Autre
10	20	30	40	50

[Crédits image](#) - [Agrandir l'image](#)

10. Dernier exemple d'**agencement avec flexbox**, pour positionner la liste et l'image côte à côte.

En HTML, il faut d'abord ajouter une balise qui imbrique ces deux éléments. Une balise `<div>`, identifiée ici par `div-regroupement`, est adaptée.

Dans le code CSS, on utilise le sélecteur basé sur l'identifiant, et on ajoute les propriétés de flexbox. La valeur `space-between` ne met des espacements qu'entre les éléments.

```
<div id="div-regroupement"><ul>
  <li>balises <span
    ↳ id="id-structurantes">structurantes
    ↳ </span></li>
  <li class="violet">balises de
    ↳ contenu</li>
  <li>balises de formulaire</li>
</ul>
</div>
```

- balises **structurantes**
- balises de **contenu**
- balises de formulaire



```
#div-regroupement {
  display: flex;
  justify-content: space-between;
}
```

Résultat final

Ci-dessous le code source complet et l'aperçu de la page avec CSS.

Ce tutoriel est volontairement limité pour en faciliter la prise en main. Des détails supplémentaires sont donnés dans les diapositives de cours (<https://perso.liris.cnrs.fr/fabien.duchateau/BDW>) ou sur d'autres ressources ([référentiel MDN](#), [tutoriel Flexbox](#) ou [tutoriel sélecteurs CSS](#)).

```
1  body {
2    background-color: white;
3  }
4
5  h1 {
6    background-color: #AACCFE;
7  }
8
9  .violet {
10   color: darkviolet;
11 }
12
13 #id-structurantes {
14   font-size : 1.5em ;
15   font-weight : bold;
16 }
17
18 h1 {
19   background-color: #AACCFE;
20   text-align: center;
21 }
22
23 p:first-child {
24   text-align: right;
25 }
26
27 table {
28   border-collapse: collapse;
29 }
30
31 table th, table td {
32   border: 1px solid black;
33   padding: 0.5em;
34 }
35
36 a:hover {
37   text-decoration: none;
38   background-color: yellow;
39 }
40
41 input[type="submit"] {
42   margin-left: 3em;
43   padding: 1em;
44 }
45
46 footer {
47   margin-top: 1em;
48   display: flex;
49   justify-content: space-around;
50 }
51
52 #div-regroupement {
53   display: flex;
54   justify-content: space-between;
55 }
56
```

Tutoriel BDW

L'objectif de ce tutoriel est d'apprendre le **HTML** :

- balises **structurantes**
- balises de contenu
- balises de formulaire



Sondage

Votre nom :

Avez-vous déjà programmé une page web ? Oui Non

Langages utilisés (en plus de HTML et CSS) :

Quel(s) framework(s) aimeriez-vous apprendre ?
Flask Django Laravel

Résultats du sondage

PHP	JavaScript	Java	Python	Autre
10	20	30	40	50

[Crédits image](#) - [Agrandir l'image](#)