

# Les listes à puces

Alors comme ça, vous voilà dans la partie III de ce tutoriel ? Bravo, on avance bien à ce que je vois 😊

Vous avez encore pas mal de choses à apprendre sur le XHTML et le CSS. Notamment, et c'est important, vous découvrirez dans cette partie comment réaliser le design de votre site web.

La particularité de cette partie III, c'est que désormais nous allons apprendre de nouvelles choses en XHTML et en CSS. Le plan de notre cours pourrait donc se résumer à ceci :

- I. XHTML
- II. CSS
- III. XHTML & CSS, la puissance combinée de ces deux langages

Ah, et tenez justement : vous voyez ce que je viens de faire ci-dessus ? Ca s'appelle une liste à puce, et c'est justement ce que nous allons apprendre à faire dans ce chapitre ! (ouah c'te transition de fou 🤪)

Ce n'est pas bien compliqué à comprendre, et ça fait un bon petit chapitre pour démarrer doucement la partie III qui va se compliquer un peu par la suite.

Car oui, pour ceux qui se poseraient des questions, la partie III est bien la partie la plus "difficile". Mais il n'y a pas de quoi s'en faire, du temps que vous restez attentifs ça ne peut que bien se passer 😊

---

## Différents types de listes (XHTML)

Ce chapitre a le mérite d'être très clair dans sa construction : dans un premier temps nous apprendrons à créer des listes à puces d'un point de vue XHTML, et dans un second temps nous apprendrons à décorer ces listes via de nouvelles propriétés CSS. Tout d'abord, il faut savoir qu'il existe 3 types de listes à puces :

- Les listes non ordonnées
- Les listes ordonnées
- Les listes de définitions (plus rares)

Nous allons voir comment on crée chacune de ces listes à puces. C'est facile, mais soyez attentifs quand même car vous allez avoir besoin des listes à puces dans les chapitres suivants. En effet, contrairement à ce qu'on pourrait croire les listes à puces sont très utilisées : elles servent notamment à créer le menu de votre site web !

### Liste non ordonnée

Une liste non ordonnée ressemble à ceci :

- Fraises
- Framboises
- Cerises

C'est un système qui nous permet de faire une liste d'éléments, sans notion d'ordre (il n'y a pas de "premier" ni de "dernier"). Créer une liste à puce non ordonnée est très simple. Il suffit d'utiliser la balise `<ul>` que l'on referme un peu plus loin avec un `</ul>`.

Commencez donc à taper ceci :

#### Code : HTML

```
1 <ul></ul>
```

Et maintenant, voilà ce qu'on va faire : on va écrire chacun des éléments de la liste entre 2 balises `<li></li>`. Toutes ces balises doivent se trouver entre `<ul>` et `</ul>`. Vous allez comprendre de suite avec cet exemple :

#### Code : HTML

```
1 <ul>
2   <li>Fraises</li>
3   <li>Framboises</li>
4   <li>Cerises</li>
5 </ul>
```

[Essayer !](#)

`<ul></ul>` indique le début d'une liste à puce  
`<li></li>` indique un nouvel élément de la liste à puce.

Vous pouvez mettre autant d'éléments que vous voulez dans la liste à puces, vous n'êtes pas limité à 3 éléments bien entendu 😊

Et voilà, vous savez créer une liste à puce non ordonnée !

C'était dur hein ? 🤔

Pour ceux qui ont besoin de faire des listes complexes, sachez que vous pouvez *imbriquer* des listes à puces (créer une liste à puce DANS une liste à puces). Si vous voulez faire ça, ouvrez une seconde balise `<ul>` à l'intérieur d'un élément `<li></li>`

Si vous fermez les balises dans le bon ordre, il ne devrait pas y avoir de problème 😊

## Liste ordonnée

Déjà que la liste à puces non ordonnée c'était de la rigolade... Eh bien là vous allez vous rendre compte que, décidément, faire un site web c'est suuuuuper dur 😊

Pour faire une liste ordonnée, on va pas se fouler. On ne change que la balise `<ul></ul>` qui devient cette fois `<ol></ol>`

A l'intérieur, on ne change rien : on utilise toujours des balises `<li></li>`.

L'ordre dans lequel vous mettez les éléments de la liste est important. Le premier `<li></li>` sera l'élément n°1, le second sera le n°2 etc...

Comme c'est particulièrement intuitif, je vous laisse admirer la simplicité de cet exemple :

#### Code : HTML

```
1 <ol>
2   <li>Je me lève</li>
3   <li>Je mange et je bois</li>
4   <li>Je retourne me coucher</li>
5 </ol>
```

[Essayer !](#)

Par rapport à l'exemple précédent, tout ce qu'on a eu à changer donc c'est la balise `<ol>`

## Liste de définitions

Là par contre, c'est différent des exemples précédents. C'est "un peu" plus difficile (mais alors franchement juste un peu ;)).

Le principe de base est le même. On indique le début et la fin de la liste à l'aide d'une balise. Ici il ne s'agit pas de `<ul>`, ni de `<ol>` mais de... `<dl>` (c'est l'abréviation de "Definition List")

Commençons donc par taper ceci :

#### Code : HTML

```
1 <dl>
2
3
4 </dl>
```

Bien, maintenant la petite différence c'est qu'on n'utilise plus la balise `<li>` pour indiquer les différents éléments de la liste.

En effet, dans le cas d'une liste de définitions, il y a 2 types d'éléments différents :

- Les mots
- Et leur définition

Les mots se trouvent entre `<dt></dt>`, et les définitions entre `<dd></dd>`.

Ce qu'il faut bien comprendre, c'est qu'on doit mettre d'abord le mot (`<dt>`), ensuite sa définition (`<dd>`). Si vous voulez mettre une seconde définition, il faut recommencer : un `dt` et ensuite un `dd`.

Regardez bien cet exemple :

### Code : HTML

```
1 <dl>
2   <dt>Chat</dt>
3   <dd>Animal à 4 pattes qui fait "Miaou !"</dd>
4   <dt>Chien</dt>
5   <dd>Animal à 4 pattes qui fait "Ouaf Ouaf !"</dd>
6   <dt>Prof de maths</dt>
7   <dd>Extraterrestre venu d'une autre planète qui enseigne des
8   choses que personne ne comprend</dd>
</dl>
```

[Essayer !](#)

Comme vous pouvez le voir, les définitions sont mises légèrement en retrait par rapport aux mots. C'est pas très joli pour le moment, mais tout le monde sait qu'avec un petit coup de CSS on peut faire en sorte que les mots soient écrit en gras et en bleu et que les définitions soient clignotantes sur fond rouge. Si vous ne trouvez pas tout seul comment faire ça en CSS, c'est que vous êtes passés trop vite sur la partie II du tutoriel 😊

Les éléments de la liste à puce fonctionnent donc ici par paire. Une fois qu'on a compris qu'il faut taper d'abord le mot, puis sa définition, bah on sait se servir des listes de définitions 😊

Tenez, si c'est pas beau ça, on a déjà fini de voir la partie "XHTML" des listes à puces ! 😊  
Que diriez-vous de pimenter tout ça avec un peu de CSS, mmh ? Ca devrait avoir encore meilleur goût ensuite 😊

---

## Décorez vos listes (CSS)

Non non, je ne vais pas vous apprendre à faire en sorte que vos définitions soient clignotantes sur fond rouge 🤖 Ca, vous savez déjà le faire :

### Code : CSS

```
1 dd
2 {
3   background-color:red;
4   /* ... */
5 }
```

Bref ça c'est du vu et revu, en plus je viens de vous mettre sur la piste 🤖  
Ce que nous allons voir maintenant ce sont des propriétés CSS spécialisées dans la

présentation des listes à puces. Il en existe 3. Nous allons étudier le fonctionnement de chacune d'elles.

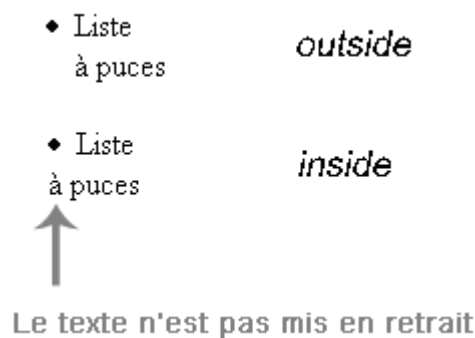
## Retrait des listes

La première des propriétés que je veux vous montrer est très facile à utiliser. Elle permet d'indiquer si on veut que la liste soit mise en retrait ou non.

Cette propriété s'appelle *list-style-position*, et peut prendre 2 valeurs :

- inside : la liste n'est pas mise en retrait.
- outside : la liste est mise en retrait (par défaut).

Ce schéma qui vous explique la différence entre inside et outside :



Voici le CSS qui nous permettra de tester le retrait :

### Code : CSS

```
1 .pas_de_retrait
2 {
3     list-style-position: inside;
4 }
5
6 .retrait
7 {
8     list-style-position: outside;
9 }
```

Et un code XHTML banal de liste à puces :

### Code : HTML

```
1 <p>Voici une liste avec retrait (par défaut) :</p>
2
3 <ul class="retrait">
4     <li>Liste<br />à puces</li>
```

```

5     <li>Ligne 1<br />Ligne 2</li>
6     <li>Vous pouvez voir<br />que le texte est décalé sur la
7 droite</li>
8 </ul>
9
10 <p>Voici une liste sans retrait :</p>
11
12 <ul class="pas_de_retrait">
13     <li>Liste<br />à puces</li>
14     <li>Ligne 1<br />Ligne 2</li>
15     <li>Vous pouvez voir<br />que le texte n'est pas décalé sur
    la droite</li>
    </ul>

```

Essayer !

Pour bien voir la différence au niveau du retrait, il faut que les éléments de la listes comportent au moins 2 lignes (c'est pour ça que j'ai mis un `<br />`). En pratique, on ne désactive pas le retrait des listes à puces, sauf quand on se sert des listes pour créer le menu de son site comme on le verra plus tard.

## Représentation de la puce

Nettement plus intéressante, la propriété *list-style-type* vous permet de changer l'apparence de la puce. En effet, vous n'êtes pas obligés d'avoir une puce sous forme de rond noir, pas plus que vous n'êtes obligés d'avoir des chiffres pour les listes numérotées (on peut utiliser des lettres : a, b, c, d...)

La propriété *list-style-type* peut prendre de nombreuses valeurs. Certaines d'entre elles concernent uniquement les listes non ordonnées, d'autre concernent uniquement les listes ordonnées :

- Pour les listes non ordonnées (`<ul>`) :
  - [disc](#) : un disque noir (par défaut).
  - [circle](#) : un cercle.
  - [square](#) : un carré.
  - [none](#) : aucune puce ne sera utilisée.
  
- Pour les listes ordonnées (`<ol>`), le choix est vaste :
  - [decimal](#) : des nombres 1, 2, 3, 4, 5... (par défaut)
  - [decimal-leading-zero](#) : des nombres commençant par zéro (01, 02, 03, 04, 05...). **Ne fonctionne pas sur Internet Explorer**
  - [upper-roman](#) : numérotation romaine, en majuscules (I, II, III, IV, V...)
  - [lower-roman](#) : numérotation romaine, en minuscules (i, ii, iii, iv, v...)
  - [upper-alpha](#) : numérotation alphabétique, en majuscules (A, B, C, D, E...)
  - [lower-alpha](#) : numérotation alphabétique, en minuscules (a, b, c, d, e...)

- o lower-greek : numérotation grecque. **Ne fonctionne pas sur Internet Explorer**

Il existe en réalité d'autres valeurs possibles pour les listes ordonnées, comme la numérotation arménienne, géorgienne, hébraïque, japonaise etc... Mais je vous en fais grâce (et je ne pense pas que vous m'en voudrez !), ce que je vous propose là devrait amplement vous suffire 😊

Et si on testait quelques-unes de ces listes à puces modifiées ? Voici le code XHTML qui va nous servir de test (un peu gros, mais ne prenez pas peur c'est un peu répétitif c'est tout 😊)

### Code : HTML

```
1 <h2>Quelques listes non ordonnées</h2>
2
3 <p>Voici une liste à puce non ordonnée sans modification (=
4 <em>disc</em>) :</p>
5 <ul>
6   <li>Liste</li>
7   <li>à</li>
8   <li>puces</li>
9 </ul>
10
11 <p>Avec <em>circle</em> :</p>
12 <ul class="cercle">
13   <li>Liste</li>
14   <li>à</li>
15   <li>puces</li>
16 </ul>
17
18 <p>Avec <em>square</em> :</p>
19 <ul class="carre">
20   <li>Liste</li>
21   <li>à</li>
22   <li>puces</li>
23 </ul>
24
25 <p>Avec <em>none</em> (sans puces) :</p>
26 <ul class="rien">
27   <li>Liste</li>
28   <li>à</li>
29   <li>puces</li>
30 </ul>
31
32 <h2>Quelques listes ordonnées</h2>
33
34 <p>Liste à puces ordonnée sans modification (=
35 <em>decimal</em>)</p>
36 <ol>
37 <li>Un</li>
38 <li>Deux</li>
39 <li>Trois</li>
40 <li>Quatre</li>
41 </ol>
42
```

```

43 <p>Avec <em>decimal-leading-zero</em> (<strong>Ne fonctionne pas
44 sous IE)</strong>) :</p>
45 <ol class="commence_a_zero">
46 <li>Un</li>
47 <li>Deux</li>
48 <li>Trois</li>
49 <li>Quatre</li>
50 </ol>
51
52 <p>Avec <em>lower-alpha</em> :</p>
53 <ol class="lettres_minuscules">
54 <li>Un</li>
55 <li>Deux</li>
56 <li>Trois</li>
57 <li>Quatre</li>
58 </ol>
59
60 <p>Avec <em>upper-roman</em> :</p>
61 <ol class="chiffres_romains">
62 <li>Un</li>
63 <li>Deux</li>
64 <li>Trois</li>
65 <li>Quatre</li>
66 </ol>
67
68 <p>Avec <em>lower-greek</em> (<strong>Ne fonctionne pas sous
69 IE</strong>) :</p>
70 <ol class="lettres_grecques">
71 <li>Un</li>
    <li>Deux</li>
    <li>Trois</li>
    <li>Quatre</li>
72 </ol>

```

... Et le CSS qui va avec :

### Code : CSS

```

1 /* Listes à puces non ordonnées */
2
3 .cercle
4 {
5     list-style-type: circle;
6 }
7 .carre
8 {
9     list-style-type: square;
10 }
11 .rien
12 {
13     list-style-type: none;
14 }
15
16 /* Listes à puces ordonnées */
17
18 .commence_a_zero
19 {
20     list-style-type: decimal-leading-zero;
21 }
22 .lettres_minuscules

```



```

23 {
24     list-style-type: lower-alpha;
25 }
26 .chiffres_romains
27 {
28     list-style-type: upper-roman;
29 }
30 .lettres_grecques
31 {
32     list-style-type: lower-greek;
33 }
34
35 /* Quelques styles supplémentaires juste pour la présentation
36 En plus ça vous fait quelques rappels ;o) */
37
38 h2
39 {
40     text-indent: 20px;
41     font-family: Arial, Verdana, "Times New Roman", serif;
42 }
43 em
44 {
45     background-color: yellow;
46 }
47 strong
48 {
49     color: red;
50 }

```

Essayer !

Je n'ai pas mis tous les styles possibles pour ne pas trop alourdir le code source, mais je pense que vous êtes assez grands pour tester les autres styles tous seuls non ? 🤖

## Changer l'image de la puce

Si aucune des représentations de puces ci-dessus ne vous convient, pourquoi ne pas créer la vôtre ? C'est justement ce que vous propose la propriété *list-style-image*, qui vous permet d'utiliser n'importe quelle image à la place d'une puce.

Vous devez lui indiquer comme valeur "url" suivi de l'adresse de l'image à utiliser. Ça fonctionne exactement comme la propriété *background* (qui permet d'indiquer une image de fond, si vous vous souvenez bien).

On va par exemple utiliser une petite image de dossier à la place de la puce, vous allez voir que ça fait beaucoup plus joli ! 😊

### Code : CSS

```

1 ul /* Ma liste aura des puces en forme de dossiers */
2 {

```

```
3     list-style-image: url("dossier.png");
4 }
5
6 /* Juste pour améliorer la présentation : */
7 a
8 {
9     color: blue;
10    text-decoration: none;
11 }
12
13 a:hover
14 {
15     text-decoration: underline;
16 }
```

[Essayer !](#)

Sympatoche non ? 😊

L'image de puce peut être de n'importe quel type (PNG, GIF ou JPEG).  
Attention cependant, l'image ne doit pas être trop grande sinon elle sera coupée. Je vous  
conseille d'utiliser une taille d'environ 15x15 pixels.