

# Premiers pas avec les CSS

Par "CSS Faciles"

Date de publication : 6 mai 2008

## Qu'est-ce que c'est ? Comment les utiliser ?

Les CSS (Cascading Style Sheets, c'est à dire "feuilles de styles en cascade") permettent de définir la *présentation* des pages Web.

Désormais, votre code HTML ne vous servira qu'à définir la *structure* de vos contenus tandis que les CSS vous permettront de créer vos styles et votre mise en page.

Cet article a été publié à l'origine à l'adresse suivante : **[Premiers pas avec les CSS](#)**

I - L'ancienne façon d'appliquer des styles aux contenus.....	3
II - Comment fonctionnent les CSS.....	4
II-A - Les CSS dans le corps du code (X)HTML (à utiliser avec modération).....	4
II-B - Les CSS dans l'en-tête de la page.....	4
II-C - Les CSS dans une feuille de style totalement séparée du code (X)HTML.....	5
II-D - Avantages des feuilles de styles séparées.....	7
III - Comment les CSS s'appliquent aux éléments (X)HTML.....	8
III-A - Comment appliquer un style aux balises (X)HTML standards.....	8
III-B - Comment utiliser des classes pour appliquer un style.....	8
III-C - Les éléments (X)HTML peuvent avoir plusieurs classes.....	9
III-D - Comment utiliser des "ID" pour appliquer un style.....	9
IV - Priorité des styles, les uns par rapports aux autres.....	11
IV-A - Styles par défaut, styles standards, classes et id.....	11
IV-B - Emboîtement de balises.....	12
Remerciements.....	14

## I - L'ancienne façon d'appliquer des styles aux contenus

Au début de l'histoire du Web, lorsque vous souhaitiez modifier le type de caractère d'une portion de texte, vous deviez utiliser la balise **<font>** en écrivant par exemple **<font face="Times">**.

Si vous souhaitiez définir la couleur de l'arrière plan et la bordure d'un tableau ainsi que le retrait (padding) de chacune de ses colonnes, vous auriez défini votre tableau par le code suivant :

```
<table border="1" bgcolor="silver" cellpadding="3" cellspacing="0">
```

### Cette approche présentait cependant de nombreux inconvénients :

- 1 Vos styles étant définis à l'intérieur même de votre contenu, ils étaient fastidieux à définir ;
- 2 Vous deviez répéter la même définition de style à chaque fois que vous souhaitiez l'utiliser dans votre page où dans les diverses pages de votre site ;
- 3 A chaque fois que vous souhaitiez modifier ce style, vous deviez parcourir le contenu de toutes vos pages pour mettre à jour, une par une, les diverses définitions qu'elles contenaient ;
- 4 Ces répétitions augmentaient la taille de vos pages ;
- 5 Vos styles étaient figés, les contenus se présentaient de la même façon pour tous les usages possibles : écran, imprimante, PDA, lecteurs vocaux, *etc.*

## II - Comment fonctionnent les CSS

Les CSS représentent une nouvelle façon très efficace d'appliquer des styles aux éléments (X)HTML.

Elles vous permettent de définir n'importe quelle propriété de style comme la bordure, le type de caractère, la couleur de fond, l'espace entre les lettres, etc. (nous reviendrons plus tard sur la manière d'y parvenir).

### Il y a trois façons principales d'appliquer des styles CSS :

- 1 Dans le corps du code (X)HTML ;
- 2 Dans l'en-tête de la page ;
- 3 Dans une feuille de style totalement séparée du code (X)HTML.

#### II-A - Les CSS dans le corps du code (X)HTML (à utiliser avec modération)

Vous pouvez définir des styles CSS directement dans la définition d'une balise (X)HTML. Dans l'exemple ci-dessous, nous utilisons une balise **<div>** qui permet de définir une "boîte" à l'intérieur d'un contenu :

##### Exemple de code

```
<div style="background-color:orange; border:1px solid black; color:yellow; font-size:150%; padding:1em;">
  Cette balise div a du style !
</div>
```

##### Ce qui donne :



Cette balise div a du style !

Cette approche est extrêmement proche de l'ancienne façon de définir des styles et présente les mêmes inconvénients.

Elle ne présente un intérêt que lorsque vous êtes certain que le style défini ne sera utilisé à aucun autre endroit ni sur aucune autre de vos pages. S'il y a la moindre chance pour que vous ayez à nouveau besoin de ce style à un autre endroit, vous devriez absolument utiliser l'une des deux autres méthodes proposées plus bas, afin de faciliter la maintenance et l'évolution de votre site.

#### II-B - Les CSS dans l'en-tête de la page

Plutôt que par la méthode précédente, il est préférable de définir vos styles CSS une fois pour toute dans une section particulière de votre page Web (on utilise normalement la section **<head>**).

```
<head>
  <style type="text/css">
    div
    {
      background-color:#339;
      color:#fff;
      padding:15px;
      border-bottom:5px solid red;
      margin-bottom:15px;
    }
  </style>
</head>

<body>
```

```
<div>
  Cette phrase est présentée en fonction du style défini dans l'en-tête
</div>
<div>
  Cette phrase aussi, est pourtant le style n'a été défini qu'une fois !
</div>
</body>
```

### Ce qui donne :



Cette phrase est présentée en fonction du style défini dans l'en-tête

Cette phrase aussi, est pourtant le style n'a été défini qu'une fois !

Grâce à cette nouvelle façon de procéder, vous n'avez besoin de définir votre style qu'une seule fois. Dans notre exemple, le style défini s'appliquera automatiquement à toutes les balises **<div>** de la page.

Avec cette méthode, vous pouvez appliquer le même style plusieurs fois dans la même page, mais pas à plusieurs pages d'un coup. Pour aller plus loin dans la standardisation de vos pages, vous devrez utiliser la troisième méthode.

## II-C - Les CSS dans une feuille de style totalement séparée du code (X)HTML

La façon idéale de définir les CSS consiste à les enregistrer dans un document indépendant de vos pages (X)HTML. Grâce à cette méthode, toutes les pages qui font référence à cette feuille de style externe hériteront de toutes ses définitions.

Un autre intérêt de cette méthode est de pouvoir définir plusieurs feuilles de styles pour le même contenu et de basculer d'une feuille à l'autre en fonction du support sur lequel le contenu est affiché (écran, imprimante, etc.). Nous reviendrons plus tard sur cet aspect.

Une page (X)HTML peut faire référence à plusieurs feuilles de styles en même temps. Dans ce cas, les définitions contenues dans ces différentes feuilles seront combinées entre elles.

Voici un exemple de styles définis dans un document séparé :

#### Document 'mes-styles.css'

```
body
{
  background-color:#ccf;
  letter-spacing:.1em;
}

p
{
  font-style:italic;
  font-family:times, serif;
}
```

#### Document 'ma-page.html'

```
<head>
  <link href="mes-styles.css" media="all" rel="stylesheet" type="text/css" />
</head>

<body>
  <p>Voici un exemple de paragraphe.</p>
  <p>Et voici un deuxième paragraphe.</p>
</body>
```

Document 'ma-page.html'

### Et voici le résultat :



Voici un exemple de paragraphe.  
Et voici un deuxième paragraphe.

Comme dans la méthode précédente (CSS dans l'en-tête de la page), le style n'a été défini qu'une seule fois mais peut être utilisé plusieurs fois. La différence entre cette méthode et la précédente, c'est que notre feuille de style peut être utilisée par un nombre illimité de pages. Il suffira d'ajouter la mention **<link href="mes-styles.css" media="all" rel="stylesheet" type="text/css" />** dans ces pages pour que notre feuille de style s'y applique.

Lorsque les utilisateurs du site chargeront une page, leur navigateur ira également lire la feuille de styles à laquelle cette page fait référence. Cette feuille de style sera gardée en mémoire par le navigateur et n'aura pas besoin d'être rechargée lors de la lecture des pages suivantes. Le résultat est un gain de temps de chargement global et une économie de bande passante pour le serveur de votre site.

La méthode "**<link href=...**" permet également de mettre en place plusieurs feuilles de styles destinées aux différents medias (imprimante, navigateurs de PDA, etc.).

Vous pouvez en effet souhaiter mettre en place une présentation particulière (sans les menus, par exemple) destinée à l'impression de vos documents. Voici une liste des valeurs les plus couramment utilisées pour **link** :

<code>&lt;link href="general.css" rel="stylesheet" type="text/css" media="all"&gt;</code>	Remplacez <i>general.css</i> par le nom que vous souhaitez donner à votre feuille de style. Cette définition vous permettra de mettre en place une feuille de style commune à tous les medias.
<code>&lt;link href="ecran.css" rel="stylesheet" type="text/css" media="screen, projection"&gt;</code>	Remplacez <i>ecran.css</i> par le nom que vous souhaitez donner à votre feuille de style. Cette définition vous permettra de mettre en place une feuille de style destinée aux écrans.
<code>&lt;link href="mobile.css" rel="stylesheet" type="text/css" media="handheld"&gt;</code>	Remplacez <i>mobile.css</i> par le nom que vous souhaitez donner à votre feuille de style. Cette définition vous permettra de mettre en place une feuille de style destinée aux PDA et téléphones mobiles.
<code>&lt;link href="impression.css" rel="stylesheet" type="text/css" media="print"&gt;</code>	Remplacez <i>impression.css</i> par le nom que vous souhaitez donner à votre feuille de style. Cette définition vous permettra de mettre en place une feuille de style destinée aux imprimantes.



**Note :** Le site [blog-and-blues](http://blog-and-blues.com) pourra vous apporter des détails sur ces sélecteurs de media.

Pour clore ce chapitre, précisons enfin que la déclaration "**<link href=...**" n'est pas la seule façon de faire appel à une feuille de style séparée. On peut également utiliser la formulation suivante :

Cette définition de style est à placer dans la section `<head>` de votre page.

```
<style type="text/css">
  @import url(styles.css) all;
</style>
```

Vous devrez remplacer "**styles.css**" par le nom que vous souhaitez donner à votre feuille de style. Vous pouvez également remplacer "**all**" par le type de média auquel se destine votre feuille de style.

Le résultat sera exactement le même que si vous aviez déclaré `<link href="styles.css" media="all" rel="stylesheet" type="text/css" />`, à deux nuances près :

- **Internet Explorer (versions 6 ou inférieures) ne reconnaît pas le type de média avec "@import"**. La feuille de style ne sera ni importée ni appliquée si vous avez précisé un média. Utilisez la formulation suivante : `<style type="text/css">@import url(styles.css)</style>` si vous voulez que votre déclaration soit reconnue par IE6.
- La règle "**@import**" permet d'importer une feuille de styles à l'intérieur d'une autre feuille de style. La déclaration aura alors la forme simple suivante : `@import url(styles.css)`. Vous devrez seulement remplacer "**styles.css**" par le nom que vous souhaitez donner à votre feuille de style.

## II-D - Avantages des feuilles de styles séparées

Il y a de multiples avantages à séparer les feuilles de styles du contenu.

### Citons en particulier :

- 1 **La réduction de la taille des pages** : Les définitions de style ne sont faites qu'une seule fois, même si elles sont utilisées plusieurs fois ;
- 2 **La réduction des temps de connexion** : Les navigateurs garderont en mémoire (en cache) le contenu de la feuille de style CSS qui s'appliquera sur toutes les pages du site. Seuls les contenus des pages devront être chargés au cours de la navigation ;
- 3 **Une mise à jour plus facile** : Vous n'aurez besoin que de changer la feuille de style pour mettre à jour la présentation de l'ensemble de votre site ;
- 4 **Scinder le travail de rédaction et le travail de présentation** : Vous pouvez commencer à rédiger le contenu de vos pages sans vous soucier de leur présentation finale. Pensez simplement à placer correctement vos balises sémantiques (titre, sous-titres, listes, classes et ID, etc.). Vous pourrez travailler votre mise en page et votre design plus tard.

### III - Comment les CSS s'appliquent aux éléments (X)HTML

Quelques règles suffisent pour pouvoir séparer le contenu et sa présentation. L'essentiel est de comprendre comment établir le lien entre un style et la phrase à laquelle s'applique le style. Ce mécanisme est heureusement assez simple.

#### III-A - Comment appliquer un style aux balises (X)HTML standards

Avec les CSS, vous pouvez changer la présentation de toutes les balises (X)HTML standards. Il vous suffit de spécifier le nom de la balise et de faire figurer vos définitions comme suit :

Cette définition de style est à placer dans une feuille de styles ou dans la section <head> de votre page.

```
p
{
  font-weight:bold;
  line-height:1.3em;
}
```

#### Code (X)HTML associé au CSS

```
<p>Ce style va s'appliquer à moi car je suis un paragraphe.</p>
<div>Mais il ne s'appliquera pas à moi.</div>
```

Le code ci-dessus aura pour effet de mettre en gras et d'augmenter la hauteur de ligne de tous les paragraphes

#### La syntaxe générale de définition est la suivante :

- 1 Préciser le nom de la balise en premier ;
- 2 Encadrer les définitions par des accolades { } ;
- 3 Placer un point virgule ; derrière chaque définition.

#### III-B - Comment utiliser des classes pour appliquer un style

Vous pouvez attribuer à chaque élément (X)HTML une ou plusieurs classes. C'est vous qui définirez le nom de ces classes et qui déciderez de leurs styles.

Les styles définis dans les classes remplaceront les styles "normaux" des éléments auxquels ils s'appliquent.

Pour créer une classe, vous devez simplement faire figurer son nom précédé d'un point. Pour éviter toute ambiguïté, votre nom de classe ne doit pas comporter d'espace.

Cette définition de style est à placer dans une feuille de styles ou dans la section <head> de votre page.

```
.mon-style
{
  color:red;
}
```

Pour appliquer le style défini dans votre classe à un élément, ajouter la mention *class="nom-du style"* dans la définition de la balise :

```
<p class="mon-style">Le style s'applique à ce paragraphe</p>
<p>Mais pas à celui-là</p>
```



Cette façon de procéder est très pratique car elle permet d'appliquer les réglages de votre classe à de nombreux éléments, même s'ils ne sont pas du même type :

```
<p class="mon-style">Le style peut s'appliquer à ce paragraphe</p>

<div class="mon-style">Et aussi à cette balise !</div>
```

### III-C - Les éléments (X)HTML peuvent avoir plusieurs classes

Chaque élément (X)HTML peut avoir aucune, une ou plusieurs classes. Pour appliquer plusieurs classes au même élément, précisez simplement la liste de classes en séparant leurs noms par un espace :

Cette définition de styles est à placer dans une feuille de styles ou dans la section <head> de votre page.

```
.mon-style1
{
    color:yellow;
}

.mon-style2
{
    background-color:#A0A0A0;
    font-weight:bold;
}
```

#### Code (X)HTML associé au CSS

```
<p class="mon-style1 mon-style2">Les styles des deux classes s'appliquent à ce paragraphe</p>

<p class="mon-style2">Alors que ce paragraphe n'a qu'une seule classe </p>
```

### III-D - Comment utiliser des "ID" pour appliquer un style

Les éléments (X)HTML peuvent se voir attribuer un *id* (identification) en plus ou à la place d'une classe.

**Le principe de l'id est très similaire à celui de la classe à une exception près :**

- Plusieurs éléments peuvent avoir la même classe ;
- Il ne doit y avoir qu'un seul élément ayant un *id* donné.

Vous pourriez parfaitement vous contenter d'utiliser les classes pour tous vos styles et oublier complètement l'existence des *id*.

**Leur utilisation permet simplement de clarifier les choses et de mieux structurer vos pages :**

- On utilise les classes pour définir l'aspect des mots, phrases et paragraphes ;
- On utilise les *id* pour placer des blocs dans la page (sachant qu'on n'a généralement jamais deux blocs placés exactement au même endroit).

Pour créer un *id*, vous devez simplement faire précéder son nom d'un dièse #. Pour éviter toute ambiguïté, votre nom d'*id* ne doit pas comporter d'espace.

Cette définition de style est à placer dans une feuille de styles ou dans la section <head> de votre page.

```
#mon-style
{
```

Cette définition de style est à placer dans une feuille de styles ou dans la section <head> de votre page.

```
color:red;  
}
```

Pour appliquer le style défini dans votre *id* à un élément, ajouter la mention **id="nom-du style"** dans la définition de la balise

```
<p id="mon-style">Le style s'applique à ce paragraphe</p>  
<p>Mais pas à celui-là</p>
```

## IV - Priorité des styles, les uns par rapports aux autres

### IV-A - Styles par défaut, styles standards, classes et id

Par défaut, chaque type de balise a une présentation particulière dans chaque navigateur. Si vous n'avez défini aucun style particulier pour la balise **<p>**, il est possible que le texte contenu dans ces balises ne se présente pas exactement de la même façon dans tous les navigateurs.

Lorsque vous définissez vous-même le style d'une balise standard avec une définition telle que **p {blablaba}**, vous obligez tous les navigateurs à afficher votre texte de la même façon.

Si vous attribuez une classe à un paragraphe, les styles que vous aurez définis dans cette classe remplaceront également les styles standards du paragraphe et seront même prioritaires par rapport aux styles que vous aurez définis sous la forme **p {blablaba}**.

Si vous attribuez un *id* à un paragraphe, les styles définis dans l'*id* seront prioritaires par rapport à tous les autres styles pouvant s'appliquer à ce paragraphe.

Si vous définissez deux fois le même style dans 2 classes différentes et que les 2 classes s'appliquent à la même balise, c'est la dernière classe citée dans la déclaration `class="classe1 classe2"` qui sera prioritaire.

Si vous définissez 2 fois le même style pour la même balise standard, la même classe ou le même *id*, c'est la dernière définition qui sera prioritaire.

#### En résumé, voici l'ordre des priorités (ceci est valable pour toutes les balises (X)HTML) :

- 1 Style standard défini par le navigateur

```
.mon-style1
{
    color:brown;
    font-weight:bold;
}

.mon-style2
{
    color:green;
}
```

- 2 Style standard redéfini en CSS (la dernière définition est prioritaire)

#### CSS

```
p
{
    color:brown;
}

.mon-style
{
    color:green;
    font-weight:bold;
}
```

#### (X)HTML

```
<p class="mon-style">
    Ce paragraphe s'affiche en vert parce que la classe est prioritaire par rapport au style
    standard de la balise
```

(X)HTML

&lt;/p&gt;

### 3 Style de classe CSS (la dernière définition est prioritaire)

CSS

```
.mon-style
{
    color:brown;
    font-weight:bold;
}

.mon-style
{
    color:green;
}
```

(X)HTML

```
<p class="mon-style">Ce paragraphe s'affiche en vert parce que c'est la dernière
(re)définition de style qui est prioritaire</p>
```

### 4 Style d'id (la dernière définition est prioritaire)

CSS

```
#mon-style1
{
    color:green;
}

.mon-style2
{
    color:brown;
    font-weight:bold;
}
```

(X)HTML

```
<p id="mon-style1" class="mon-style2">
    Ce paragraphe s'affiche en vert parce que l'ID est prioritaire sur la classe
</p>
```

## IV-B - Emboîtement de balises

Dans un cas d'emboîtement de balises tel que celui-ci :

```
<body>
  <div>
    <p>
      bla bla bla
    </p>
  </div>
</body>
```

Les différents styles attribués aux balises **<body>**, **<div>** et **<p>** vont se combiner pour définir quels sont les styles définitifs qui seront appliqués au texte **"bla bla bla"**.

Si le même style est défini, puis redéfini dans ces différentes balises (la couleur du texte, par exemple), c'est la dernière définition qui l'emportera sur les autres.

Si un style particulier n'est défini que dans **<body>** il doit logiquement s'appliquer à l'ensemble du contenu de **<body>**.

En vérité, certains styles (tels que la couleur, par exemple) se transmettent automatiquement aux balises emboîtées, tandis que d'autres (tels que les marges, par exemple) ne se transmettent pas. Lorsqu'ils se transmettent, on dira que le contenu des balises **<div>** et **<p>** aura "hérité" des propriétés de **<body>**.

## Remerciements

Tous mes remerciements à **julp** pour sa relecture.