



Les pseudo-formats

Informations sur le tutoriel

[Ajouter à mes tutoriels favoris](#) (10 fans)



Auteur : [M@teo21](#)

Licence :

[Plus d'informations](#)

Popularité

Visualisations : 145 619 366

Appréciation 15

des lecteurs :15

60

888

Publicité

Historique des mises à jour

- *Le 08/08/2010 à 03:23:12*
Correction conjugaison.
- *Le 03/08/2010 à 01:07:48*
Correction orthographique, #2605
- *Le 31/07/2010 à 18:20:55*
Correction orthographique, ticket n°2593

Nous venons de passer en revue un grand nombre de propriétés CSS dans les chapitres précédents. Vous savez maintenant modifier la taille du texte, sa police, sa couleur etc etc...

Nous allons voir dans ce chapitre un nouvel aspect du langage CSS qu'on appelle... *les pseudo-formats*

Nous n'allons pas apprendre de nouvelles propriétés CSS (vous en savez déjà pas mal), nous allons plutôt voir comment les appliquer à des moments ou des endroits précis. Par exemple, nous allons apprendre à modifier l'apparence d'un lien au passage de la souris, à modifier automatiquement la première lettre d'un paragraphe etc...

Tous ces éléments devraient vous permettre d'ajouter encore plus de dynamisme à votre site web. Que du bon pour vous quoi 😊

Sommaire du chapitre :



- [Des liens sympas](#)
 - [Première lettre et première ligne](#)
 - [Avant / Après](#)
 - [Q.C.M.](#)
-

Des liens sympas

Si vous avez bien suivi les chapitres précédents, vous savez comment modifier l'apparence des liens. Il suffit d'appliquer des styles à la balise `<a>` et le tour est joué.

Par défaut, les liens s'affichent en bleu et sont soulignés. Supposons que vous ne vouliez pas du soulignement, vous allez utiliser :

```
text-decoration:none;
```

... ce qui aura pour effet d'annuler le soulignement. Vos liens ne seront alors plus soulignés.

Bon jusque-là, j'espère que je ne vous apprends rien de trop nouveau 😊

Allez, petite révision pour se chauffer un peu, après on passe aux choses sérieuses. Voici un CSS qui applique 2-3 styles aux liens pour changer des immondes liens bleus soulignés 😊

Code : CSS

```
a
{
  text-decoration: none; /* Les liens ne seront plus soulignés */
  color: red; /* Les liens seront en rouge au lieu de bleu */
  font-style: italic; /* Les liens seront en italique (pourquoi pas ?) */
}
```

[Essayer !](#)

Nous allons apprendre à modifier l'apparence des liens :

- ... lorsque le visiteur pointe dessus avec la souris.
- ... lorsque le visiteur clique dessus.
- ... lorsque le visiteur a sélectionné le lien.
- ... lorsque le visiteur a déjà vu la page.

Au passage de la souris

Le premier pseudo-format que je vais vous montrer s'appelle `:hover`. Comme tous les autres pseudo-formats que nous allons voir, c'est une information que l'on rajoute après le nom de la balise (ou de la class) dans le CSS, comme ceci :

Code : CSS

```
a:hover
{
}
}
```

`":hover"` signifie "Dessus".

`"a:hover"` signifie donc : "Quand la souris est sur le lien" (quand on pointe dessus).

A gauche, vous tapez comme d'habitude la balise concernée (en l'occurrence `<a>`, la balise de lien), et à droite vous mettez le pseudo-format.

A partir de là, c'est à vous de définir l'apparence que doivent avoir les liens lorsqu'on pointe dessus. Laissez libre cours à votre imagination, il n'y a pas de limite 😊

Voici un exemple de présentation des liens, mais n'hésitez pas à inventer le vôtre :

Code : CSS

```
a
{
  text-decoration: none; /* Les liens ne seront plus soulignés */
  color: red; /* Les liens seront en rouge au lieu de bleu */
  font-style: italic; /* Les liens seront en italique (pourquoi pas ?) */
}
a:hover /* Quand le visiteur pointe sur le lien */
{
  text-decoration: underline; /* Le lien deviendra souligné quand on pointera dessus */
  color: green; /* Le lien sera écrit en vert quand on pointera dessus */
}
```

[Essayer !](#)

Sympa , n'est-ce pas ? 😊

La bonne nouvelle, c'est que vous pouvez appliquer le pseudo-format ":hover" à n'importe quelle balise. La mauvaise nouvelle, c'est que pour Internet Explorer 6 le ":hover" ne marche que sur les liens. Sous Internet Explorer 7, il marche en théorie sur toutes les balises, mais d'après mes tests il y a quand même quelques bugs.

Sur tous les autres navigateurs (dont Mozilla Firefox, que j'espère que vous avez téléchargé depuis le temps 😊) ça fonctionne à la perfection.

Voici un exemple d'utilisation du :hover sur un paragraphe (à tester avec un autre navigateur que IE) :

Code : CSS

```
p:hover /* Quand on pointe sur un paragraphe */
{
  background-color: #CFE1EB; /* Le fond du paragraphe change de couleur */
  text-indent: 20px;
}
```

[Essayer !](#)

Si vous pointez sur les paragraphes, vous voyez donc qu'ils changent de couleur. Pas très utile, mais après tout l'objectif du CSS c'est de décorer non ? 😊

Au moment du clic

Le pseudo-format **:active** permet d'appliquer un style particulier au moment du clic. Le lien gardera cette apparence très peu de temps : juste pendant que vous cliquez avec le bouton de la souris. En clair, ce n'est pas forcément toujours bien visible.

Pour ma part, un effet que je trouve bon d'appliquer avec **:active**, c'est de changer la couleur de fond du lien. Comme ça, on voit bien qu'on a cliqué 😊 :

Code : CSS

```
a:active /* Quand le visiteur clique sur le lien */
{
    background-color: #FFCC66;
}
a:hover /* Quand le visiteur pointe sur le lien */
{
    text-decoration: underline;
    color: green;
}
a /* Lien normal */
{
    text-decoration: none;
    color: red;
    font-style: italic;
}
```

[Essayer !](#)

Lorsque le lien est sélectionné

Là, c'est à peine différent. Le pseudo-format `:focus` applique un style lorsque le lien est sélectionné.

C'est-à-dire ?

C'est-à-dire... que ça revient un peu comme `:active`, c'est-à-dire au moment du clic (en tout cas pour un lien). Ce pseudo-format, appliqué à d'autres balises XHTML que nous n'avons pas encore vues, permettra de créer des effets assez sympas, vous verrez 😊

Pour ce qui est des liens, je vous mets un exemple très similaire au précédent (on change le fond) pour que vous puissiez comparer :

Code : CSS

```
a:focus /* Quand le visiteur sélectionne le lien */
{
    background-color: #FFCC66;}
a:hover /* Quand le visiteur pointe sur le lien */
{
    text-decoration: underline;
    color: green;
}
a /* Lien normal */
{
    text-decoration: none;
    color: red;
    font-style: italic;
}
```

[Essayer !](#)

Vous le voyez : le lien garde sa couleur de fond un peu plus longtemps. Personnellement, je préfère utiliser un `:focus` à la place d'un `:active` car je trouve ça plus visible.

Mais comme IE (dans ses anciennes versions) ne comprend pas le focus, j'applique le même style au `:focus` et au `:active`, comme on a appris à le faire dans le premier chapitre CSS, en séparant les noms par une virgule :

Code : CSS

```
a:active, a:focus /* Appliquer le même style aux liens actifs et focus */
{
    background-color: #FFCC66;
}
```

L'avantage de cette technique, c'est que si le navigateur est IE, il prendra uniquement le :active (et le fond sera un peu coloré), tandis que si c'est un autre navigateur le fond restera plus longtemps grâce au :focus.

A vous de vous faire votre propre idée, je ne vous oblige à rien 😊

Quand la page a déjà été vue...

Il est possible d'appliquer un style à un lien vers une page qui a déjà été vue. Par défaut, le navigateur colore le lien en un violet immonde (encore plus immonde que le bleu souligné :p)

Personnellement, je préfère éviter de modifier l'apparence des liens qui ont déjà été vus car je trouve que c'est assez peu agréable au final visuellement. Mais là encore, c'est une opinion personnelle, je ne veux pas vous influencer 😊

Le pseudo-format qui nous intéresse s'appelle **:visited** (qui signifie "visité"). Une application marrante (mais pas très utile) de ce pseudo-format pourrait être de barrer tous les liens qui ont été vus :

Code : CSS

```
a:visited /* Quand le visiteur a déjà vu la page concernée */
{
    text-decoration: line-through;
}
a:focus /* Quand le visiteur sélectionne le lien */
{
    background-color: #FFCC66;
}
a:hover /* Quand le visiteur pointe sur le lien */
{
    text-decoration: underline;
    color: green;
}
a /* Lien normal */
{
    text-decoration: none;
    color: red;
    font-style: italic;
}
```

[Essayer !](#)

Si vous avez cliqué sur tous les liens, ils seront tous barrés et vous n'y verrez plus grand chose, forcément 😊

Bon, plus sérieusement, si vous voulez appliquer un style précis à des liens déjà visités, je peux vous conseiller de colorer les liens en légèrement plus clairs que la normale. Si vos liens sont verts en temps normal, mettez-les en vert clair lorsqu'ils ont été visités.

Enfin, c'est une suggestion comme une autre, mais c'est un effet qui passe plutôt bien je trouve.

Bon allez, j'arrête de vous proposer mes idées, sinon tous vos sites vont se ressembler 😊

D'ailleurs, on a fini de voir les pseudo-formats qu'on utilise généralement sur les liens. On va maintenant s'intéresser aux pseudo-formats modifiant la première lettre ou la première ligne.

Première lettre et première ligne

En CSS, il est possible de modifier automatiquement l'apparence de la première lettre et de la première ligne du texte contenu dans une balise.

Il s'agit là encore de pseudo-formats. Si on a tendance à utiliser ceux-là plutôt sur des paragraphes, n'oubliez pas que ça fonctionne aussi bien sur d'autres balises (comme les titres).

La première lettre

Pour modifier l'apparence de la première lettre, vous devez utiliser le pseudo-format **:first-letter**

On peut s'en servir pour écrire en gras et en plus gros la première lettre de tous les paragraphes de votre page. Si en plus vous indentez votre texte avec *text-indent*, on va finir par se croire dans un roman ou un article de journal 😊

Code : CSS

```
p:first-letter /* La première lettre de chaque paragraphe */
{
    font-weight: bold; /* En gras */
    font-size: 1.2em; /* Ecrit légèrement plus gros que la normale */
    color: blue; /* En bleu */
}
p
{
    text-indent: 20px;
}
```

[Essayer !](#)

La première ligne

Un autre pseudo-format intéressant permet de modifier cette fois la première ligne. Il s'agit de **:first-line**.

Appliqué aux paragraphes, il permet d'inciter à la lecture du texte. Par exemple, vous pourriez automatiquement écrire en petites capitales la première ligne de chaque paragraphe, afin de les rendre plus attrayants :

Code : CSS

```
p:first-line /* La première ligne de chaque paragraphe */
{
    font-variant: small-caps; /* En petites capitales */
}
p
{
    text-indent: 20px;
}
```

[Essayer !](#)

Bref, pas besoin d'y passer 3/4 d'heure non plus, c'est facile à utiliser donc je vous laisse faire joujou avec ces nouvelles possibilités 😊

Avant / Après

Les pseudo-formats que nous allons voir maintenant sont très intéressants et changent un peu de ce qu'on a l'habitude de faire en CSS. C'est toutefois un peu plus difficile que le reste, aussi ouvrez grand vos oreilles parce

que ça en vaut la peine.

Bon, de quoi parle-t-on ici ?

On parle de la possibilité d'ajouter automatiquement du texte au début et à la fin de certains paragraphes, afin de diminuer la quantité de texte que vous avez à écrire.

Par exemple, supposons que je pose beaucoup de questions dans ma page web. En temps normal, j'écrirai ceci en XHTML pour indiquer qu'une personne pose une question :

Code : HTML

```
<p>Question : quel est l'âge du webmaster ?</p>
```

[Essayer !](#)

Maintenant, supposons que je sois un webmaster fainéant (et vous n'avez pas idée du nombre de webmasters fainéants qui existent, j'en fais d'ailleurs partie :p). Je souhaite automatiser l'insertion du texte "Question : " au début et du point d'interrogation à la fin, pour ne pas avoir à les répéter 50 fois dans mon code XHTML. Je vais créer une class "question" qui permettra de mettre en forme automatiquement toutes les questions de mon texte.

Voici le code XHTML qui va vous permettre de comprendre le truc :

Code : HTML

```
<h1>Plein de questions</h1>

<p>J'ai créé le Site du Zér0 parce que j'en avais marre de ne pas trouver

<p class="question">De quand date ce site</p>

<p>Il a été créé aux alentours de 1999</p>

<p class="question">Quel a été le premier nom du site</p>

<p>Il n'a jamais changé de nom, il s'est toujours appelé "Site du Zér0"</p>

<p class="question">Combien de temps as-tu passé pour trouver un nom aussi

<p>Environ 40 secondes. Oui je sais, c'est beaucoup vu le résultat, mais c

<p class="question">Quel est l'âge du webmaster</p>

<p>Eh oui ? Qui ça moi ? Mon âge ne vous intéresse pas, allez, allez, ch
```

[Essayer !](#)

Utilisons maintenant les pseudo-formats suivants :

- **:before** : qui signifie "avant"
- **:after** : qui signifie "après"

On va utiliser une propriété CSS nouvelle et un peu particulière qui nous permettra d'insérer du texte : content. Regardez comment j'utilise :before et :after sur les paragraphes équipés de la class "question" :

Code : CSS

```
.question:before /* Avant chaque question */
{
    content: "Question : "; /* Commencer par Question : */
}
.question:after /* Après chaque question */
{
    content: " ?"; /* Rajouter un point d'interrogation */
}
```

[Essayer !](#)

content est donc une propriété CSS vraiment particulière, vu qu'elle permet d'indiquer le texte que l'on veut mettre "Avant" (before) ou "Après" (after). C'est donc assez différent de ce qu'on a fait jusque-là (et je vous rassure, c'est la seule propriété CSS vraiment "à part" qui existe).

Vous n'êtes pas obligés de mettre uniquement la propriété *content* avec un *:before* ou *:after*, vous pouvez aussi utiliser tous les styles CSS que vous connaissez. Ces styles s'appliqueront au texte rajouté par le *:before* ou *:after*.

Code : CSS

```
.question:before
{
    content: "Question : ";
    font-weight: bold; /* "Question" sera en gras */
    color: blue; /* "Question" sera en bleu */
}
.question:after
{
    content: " ?";
}
```

[Essayer !](#)

Comme vous le voyez, la mise en gras et en bleu s'appliquera donc au texte rajouté par le *:before*.

Si vous appliquez le pseudo-format *:first-letter* à la class *question*, pour mettre en gras par exemple la première lettre, c'est la première lettre de "Question :" (le Q) qui sera mise en gras, et non pas la lettre qui se trouve juste après "Question :": *:first-letter* s'appliquera donc à la première lettre du texte écrit avec le *:before*.

Une image au lieu du texte ?

Il est aussi possible d'indiquer une image à la place du texte avec le *:before* et le *:after*.

On utilise toujours la propriété *content*, mais au lieu de taper du texte entre guillemets, on va mettre une valeur "url" comme ceci :

Code : CSS

```
.question:before
{
    content: url("../images/qst.gif"); /* Mettre une image de question */
}
.question:after
{
    content: " ?";
}
```

[Essayer !](#)

N'oubliez pas que le chemin relatif vers l'image se fait à partir de la position du fichier CSS et non pas du fichier XHTML. Si vous vous trompez dans le chemin, l'image ne s'affichera pas.

Ce que je vous montre là ne sont que des exemples d'utilisation, vous en trouverez certainement une utilité plus appropriée à votre site web 😊

Q.C.M.

Quel pseudo-format permet d'indiquer un style lorsque la souris pointe sur une balise précise (comme un lien) ?

- ☐ :hover
- ☐ :focus
- ☐ :pointer

Comment faut-il faire pour indiquer un style particulier aux liens ni visités, ni cliqués, ni survolés avec la souris ?

- ☐ Utiliser le pseudo-format :normal
- ☐ Utiliser le pseudo-format :standard
- ☐ Appliquer le style à la balise de lien <a>

Que fera ce CSS ?

Code : CSS

```
blockquote
{
    font-style:italic;
}

blockquote:first-line
{
    font-size:1.1em;
}
```

- ☐ Ecrira la première ligne des citations en italique et en un peu plus grand.
- ☐ Ecrira la première ligne des citations en un peu plus grand, et le reste en italique.
- ☐ Ecrira toute la citation en italique, et la première ligne sera en plus écrite en plus grand.

Quel pseudo-format permet d'appliquer un style particulier aux liens vers des pages qui ont déjà été vues ?

- ☐ :seen
- ☐ :visited
- ☐ :active

Que fait ce CSS ?

Code : CSS

```
h2:before
{
    content:"Sous-titre : ";
    font-style:italic;
}
```

- ☐ Les titres h2 seront entièrement écrits en italique et précédés de la mention "Sous-titre : "
- ☐ Les titres h2 seront précédés de la mention "Sous-titre : " qui sera la seule à être en italique.
- ☐ Les titres h2 seront écrits en italique, sauf la mention "Sous-titre : " qui est automatiquement rajoutée.

Correction !

[Statistiques de réponses au QCM](#)

Les pseudo-formats sont un aspect intéressant du CSS bien qu'ils amènent parfois à des confusions. Si vous avez répondu tout juste au QCM c'est que vous avez tout compris. Si vous avez fait des erreurs, sachez que c'est un peu normal de se planter la première fois. Essayez de faire des tests avec les CSS proposés pour essayer de comprendre "le truc".

En pratique, on voit assez peu sur le web de pseudo-formats autres que ceux concernant les liens. Il est vrai qu'avoir des liens qui changent de couleur quand on pointe dessus par exemple, c'est assez sympa 😊
:first-line et :first-letter fonctionnent bien sur tous les navigateurs (dont Internet Explorer) mais sont assez peu connus et c'est dommage car ils permettent de donner un "petit plus" à la présentation de votre page web sans faire d'efforts.
Enfin, :before et :after sont très peu utilisés (ils ne sont pas toujours compatibles avec tous les navigateurs)

La partie II traitant uniquement du CSS s'arrête là. Eh oui, déjà 😊
Mais il faut reconnaître que ces 4 chapitres ont été riches en nouveautés pour vous 😊

Si on résume ce qu'on a fait jusqu'ici :

- I. La première partie vous a enseigné toutes les bases du XHTML, de quoi faire une première page web avec des liens et des images. La page était certes un peu moche, mais...
- II. on a introduit le CSS dans cette partie II. Désormais, sans rien changer au fichier XHTML, vous avez appris à donner une belle présentation à votre page web par le biais des fichiers .css. Les possibilités sont quasi-infinies, c'est à vous de mélanger les propriétés CSS à votre goût.
- III. Toutefois, on n'a pas encore tout vu sur la création de site web. Il reste des balises XHTML un peu plus complexes, notamment pour créer des tableaux ou des formulaires (mixés à du CSS pour leur donner une belle présentation). Il reste aussi à apprendre à manier les "calques", un passage obligatoire si vous voulez créer le design de votre site web.

En clair, la partie III mélangera XHTML et CSS et vous permettra de maîtriser enfin le design de votre site web. C'est la dernière ligne droite et il vous reste encore plein de nouvelles choses à découvrir 😊

Quand vous êtes prêts, rendez-vous dans la partie III 😊