POMPAGE - Web design puisé à la source. [http://www.pompage.net]

CSS : on reprend tout à zéro ! (13ème épisode)

Par *Joe Gillespie [http://www.wpdfd.com/]*

Cet article est le treizième d'une série qui en compte 15. Chaque épisode sera publié bimensuellement dans nos pages.

Pour de nombreux designers, le style des formulaires dans les pages Web ne constitue pas une grande priorité. Les formulaires sont généralement laissés « tels quels », et pour de bonnes raisons d'ailleurs.

Les conventions qui sont utilisées par défaut pour les formulaires fonctionnent assez bien en l'état.
Si un formulaire n'a pas l'aspect attendu, les lecteurs pourraient ne pas s'y retrouver.
Le style des formulaires peut devenir très différent d'un navigateur à l'autre, selon l'interprétation qu'ils en font.

Voyons tout cela de plus près.

« Les conventions qui sont utilisées par défaut pour les formulaires fonctionnent assez bien en l'état. »

Bon, peut-être. Pourtant on trouve encore un peu partout beaucoup de tableaux mal conçus, que ce soit sur papier ou sur le Web.

Moi en tous cas, je déteste remplir des formulaires, et je sais que je ne suis pas seul à réagir ainsi.

Mais les formulaires sont bien pratiques pour récolter de l'information, d'autant qu'ils la récoltent d'une manière qui en rend l'exploitation assez facile. Alors, quelle est l'erreur commise par les « designers » de formulaires ?

Le premier gros écueil apparaît lorsque le formulaire **semble** compliqué. L'utilisateur est intimidé et se place en position défensive.

Et puis, on nous pose des questions qui peuvent paraître soit inutiles, soit indiscrètes. Pourquoi voulez-vous absolument connaître ma date de naissance ?

L'aspect compliqué est une question de mise en forme. On peut donner à un formulaire une apparence simple en le présentant de manière propre et ordonnée. La première chose dont on doit s'assurer, c'est de bien contrôler les tailles de police pour ne pas les laisser démolir la mise en page.

La taille et la position des légendes ou des textes explicatifs doivent aussi être réfléchies. Le mieux est de les placer soit au-dessus ou en dessous des champs de saisie, soit sur la droite. Si vous tenez vraiment à mettre des textes sur la gauche, utilisez `align: right;` pour éviter de vous retrouver avec

d'énormes trous qui déconnectent visuellement le texte de son champ, et augmentent la complexité visuelle de la mise en page. Essayez également de ne pas avoir trop de tailles différentes pour vos champs de saisie, et ne remplissez pas les espaces vides à tout prix. Si le champ du code postal est plus court que les autres champs d'adresse, ce n'est pas un problème : ne placez pas pour autant différentes tailles de champs sur une même ligne horizontale juste pour gagner de l'espace.

Sur une page Web, on n'a pas besoin de gagner de l'espace. Choisissez intelligemment les tailles de vos champs de saisie. Alignez-les verticalement. Gardez-les propres et dégagés.

Ci-dessus, un formulaire qui semble compliqué. Il y a deux axes verticaux, et le déplacement du regard est perturbé par la nécessité de sauter horizontalement et verticalement pour trouver le champ suivant.

Voici un design plus simple, avec un seul axe vertical en plein milieu, ce qui n'interrompt pas le déplacement du regard.

L'avantage par rapport à un formulaire papier, c'est qu'un formulaire Web masque tout ce qui n'est pas immédiatement visible dans la fenêtre du navigateur, évitant ainsi de noyer le visiteur sous la masse. Néanmoins, un formulaire très long sur une seule page exigera de nombreux défilements verticaux, ce qui peut être encore pire. Mieux vaut diviser les formulaires longs sur toute une série de pages, à

condition toutefois de penser à permettre aux utilisateurs de revenir en arrière s'ils veulent modifier quelque chose.

« Si un formulaire n'a pas l'aspect attendu, les lecteurs pourraient ne pas s'y retrouver. »

Il est clair qu'on ne peut pas « s'éclater » dans le design de formulaires. Il faut bien que le lecteur comprenne la fonction d'un champ de saisie de texte, comme pour une fenêtre pop-up ou un bouton d'envoi. Ça ne signifie pas pour autant que nous sommes condamnés au texte noir sur fond blanc (encore que certains navigateurs n'acceptent rien d'autre !)

Ce serait une erreur d'assimiler un champ de saisie à du texte ordinaire, la fenêtre doit avoir son apparence propre, tout en indiquant sa fonction.

Les éléments que l'on peut styler sont `input`, `textarea` et `select`, mais attention : tout style attribué à l'élément `input` affectera non seulement le champ de saisie de texte, mais également les autres outils de saisie que sont les boutons radio, les cases à cocher, ainsi que les boutons « Envoyer » et « Effacer ».

On peut sans risque jouer sur les styles `font-family` et `font-size` des polices. La couleur du texte (`text-color`) peut varier si on le souhaite. Pour la couleur de fond (`background-color`), c'est plus discutable. Si elle est d'une teinte plus pâle que celle de la page, c'est acceptable, et cela peut rendre l'aspect du formulaire moins agressif.

Si on change le contour d'un champ de saisie de données (« `input` »), cela modifie également le contour du bouton d'envoi, qui du coup peut se mettre à ressembler à n'importe quelle zone de texte si on n'y prend garde. Pour éviter cela, il faut donc « envelopper » le champ de saisie dans une `div` qui lui sera propre, et se contenter d'appliquer des styles sur `#saisie input` comme ci-dessous :

```
#saisie { }

#saisie input
        {
        color: #633;
        font-size: 10px;
        background-color: #ebebd9;
        padding: 3px;
        border: double 3px orange
        }
```

NdT : les éléments `input` et `textarea` ne peuvent être traduits, car il s'agit de code HTML (HyperText Markup Language). Rappelons que `input` fait référence à de la saisie de données de formulaire (un nom, une date, un chiffre...), tandis que `textarea` permet de saisir des commentaires, des remarques ou autres textes longs.

Si maintenant vous voulez qu'un champ `textarea` ressemble à un champ `input`, vous pouvez faire

ceci :

```
#saisie input, #saisie textarea
        {
        color: #633;
        font-size: 10px;
        background-color: #ebebd9;
        padding: 3px;
        border: double 3px orange
        }
```

Quant à l'élément `select` (il s'agit des listes déroulantes), son rendu varie d'un navigateur et d'une plateforme à l'autre. Tout ce que nous pouvons faire, c'est fixer une taille et une famille de police. De toute manière, vous ne pouvez généralement avoir aucune idée de l'aspect final, une fois qu'ils auront été cliqués.

« Le style des formulaires peut devenir très différent d'un navigateur à l'autre. »

La meilleure des preuves réside dans ces quelques *copies d'écran [/IMG/html/tests_formulaires.html]*.

Les mêmes styles interprétés sous Explorer (Windows), Mozilla, Safari (Mac) et Opera montrent quelle variété on peut trouver. Je reviendrai dans le prochain épisode sur les différences entre les interprétations des navigateurs.

L'important ici est que nous ne cherchons pas à enjoliver les formulaires. Nous essayons de les rendre plus faciles à utiliser et moins impressionnants grâce à une présentation visuelle réfléchie.

On peut encore faciliter davantage la vie de l'utilisateur avec des scripts de validation de formulaires qui sont bien utiles, mais c'est une tout autre histoire, dans laquelle je ne compte pas m'engager tout de suite.

Episode 14 [/pompe/cssdezero-14/]

Fiche technique

- *Article original : CSS from the Ground Up -13 [http://www.wpdfd.com/editorial/basics/cssbasics13.html]*
- *Auteur : Joe Gillespie [http://www.wpdfd.com/]*
- *Date de parution :* 2004
- *Traduction :* Sylvain Lelièvre, Frédéric Chotard
- *Date de traduction :* décembre 2005

À propos de l'auteur

Joe Gillespie est un web designer qui partage ses passions pour le design, les CSS et la typographie

depuis 8 ans sur son site *Web Page Design for Designers*. Il a pris sa retraite du Web cette année, au terme de 80 publications mensuelles. Dommage pour nous !