Les 10 commandements d'un bon design de formulaire

Type : traduction et adaptation enrichie et commentée

Source : Mono

Commentaire personnel : un formulaire est un élément assez commun de nos jours, il ne faut toutefois pas le négliger et le concevoir à la hâte. C'est souvent un point clé (dans le "Lead Generation" - ou Génération de prospects - par exemple) et il doit faire partie intégrante d'une UX réussie. Il y a des bonnes pratiques a prendre en compte, cet article essaye de faire le tour des aspects principaux d'un design de formulaire réussi. Je ne suis pas entièrement d'accord avec tous les points et je le détaille plus bas mais cela reste une source tout à fait utile et une check-list à vérifier lorsque vous implémenterez vos prochains formulaires.

Bonne lecture, et n'hésitez pas à laissez vos questions et commentaires !

---

1. Tu devras fournir des libellés clairs et toujours visibles pour chaque champ

Il existe une mode dans le web design actuel qui est d'afficher le libellé d'un champ seulement quand celui-ci est sélectionné. Autant cela peut sembler "cool" pour un formulaire simple de type "identifiant/mot de passe", pour un formulaire plus long, c'est probablement une mauvaise idée.

Lorsque vous avez la place pour afficher un libellé, faites le.

Dans un formulaire long, l'utilisateur aura tendance à vérifier le texte qu'il aura saisi ; quand vous ne pouvez pas voir quel champ correspond à quel libellé, vous ne pourrez pas effectuer correctement cette vérification.

Illustration 1 : les libellés sont inscris directement dans les champs. Cela a l'air plus clair

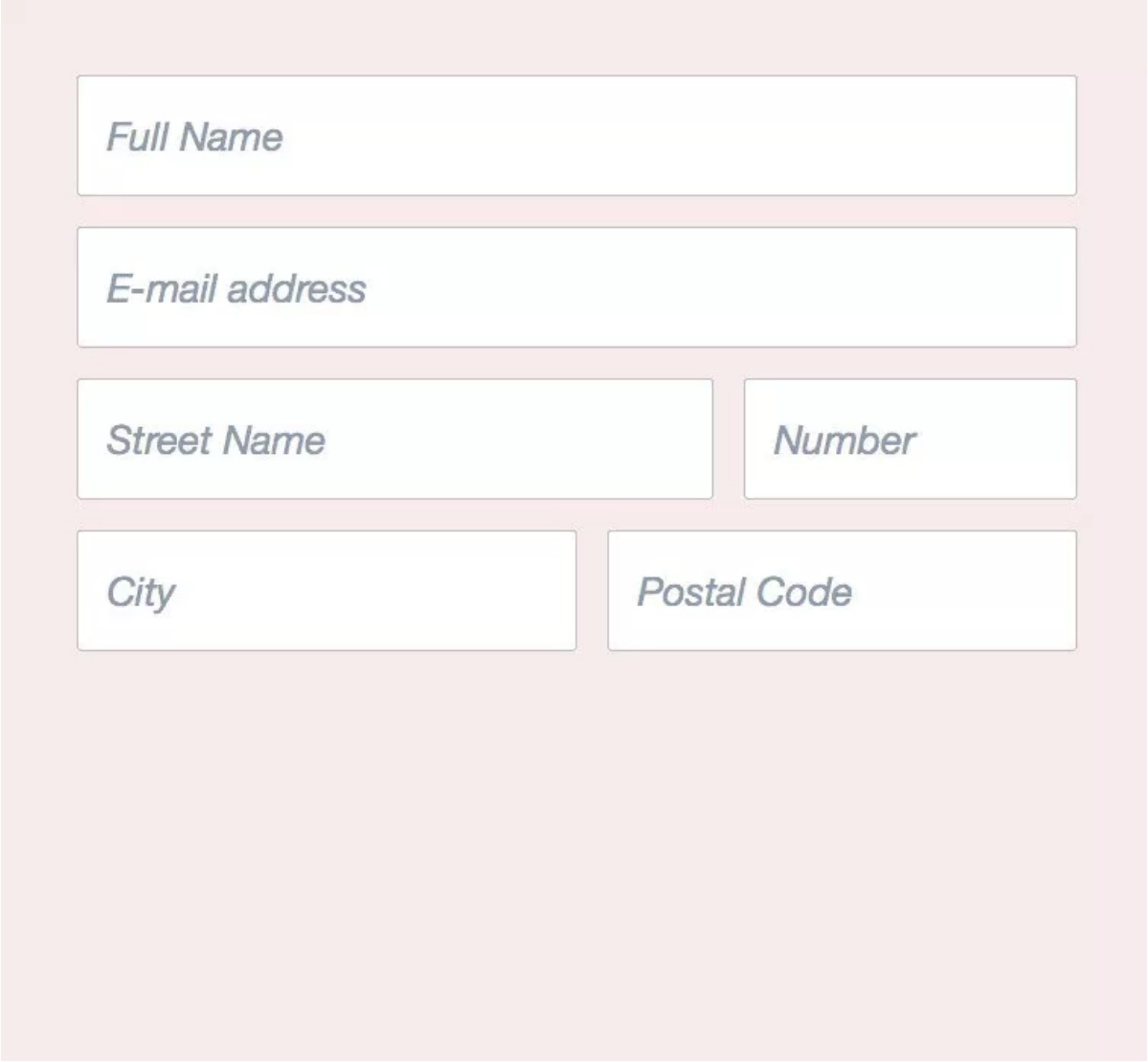
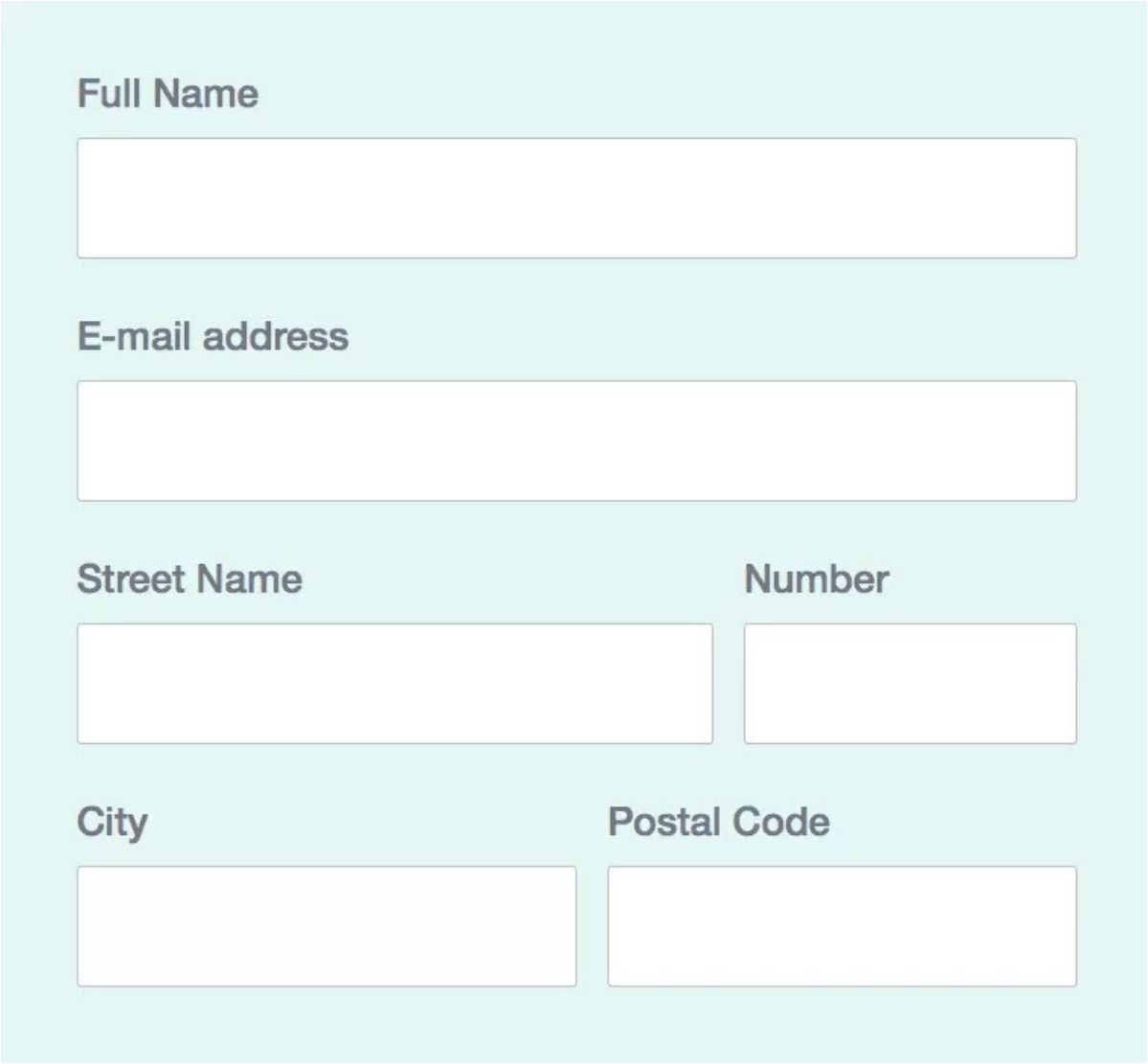


Illustration 2 : les libellés disparaissent lorsqu'un champ est sélectionné. Il est plus compliqué de voir une erreur

Illustration 3 : la meilleure solution est d'écrire des libellés clairs et permanents pour chacun des champs



2. Tu devras utiliser une taille de police suffisamment grande

Vos polices doivent être assez grandes pour être lisibles. 14 pixels est recommandé pour le corps de texte, on peut aller jusqu'à 16 pixels avec certaines polices. La taille dépend aussi du contexte (mobile ou pas ?) et de la quantité d'autres éléments sur la page. Dans le doute, choisissez plus grand.

De plus, si vous choisissez une taille de 16 pixels pour vos champs de formulaires, iOS ne zoomera pas lorsque vous sélectionnerez un champ, ça ne sera simplement plus nécessaire.

Commentaire personnel : à mon avis, 14px est encore trop petit au vu des résolutions actuelles, 16px me semble être une bonne taille de base. Les arguments les plus fréquents en faveur d'une taille de police plus petite sont souvent obsolètes : le scrolling et les pages longues sont devenues un standard et les résolutions d'écran en 640x480 font parti d'un passé (heureusement) révolu depuis longtemps.

Lorsque vous pouvez fournir une Expérience confortable à l'utilisateur, aussi facilement qu'avec un changement de taille de police, ne vous posez pas de question et faîtes le.
Notez que l'interlignage est au moins aussi important que la taille du texte… mais ceci est un autre sujet que j'aborderai certainement par ailleurs.

Illustration 1 : les textes et libellés trop petits sont difficiles à lire

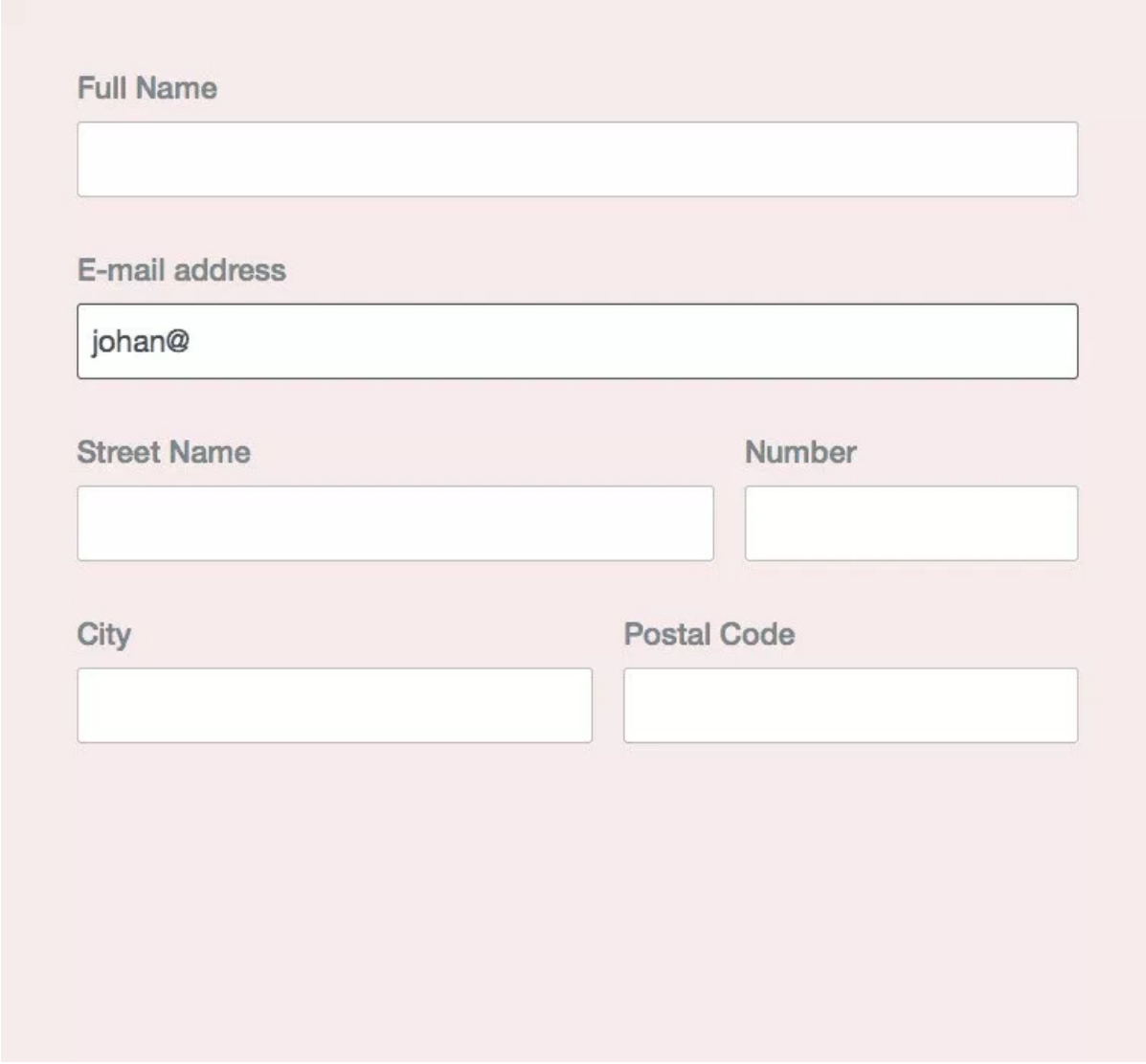
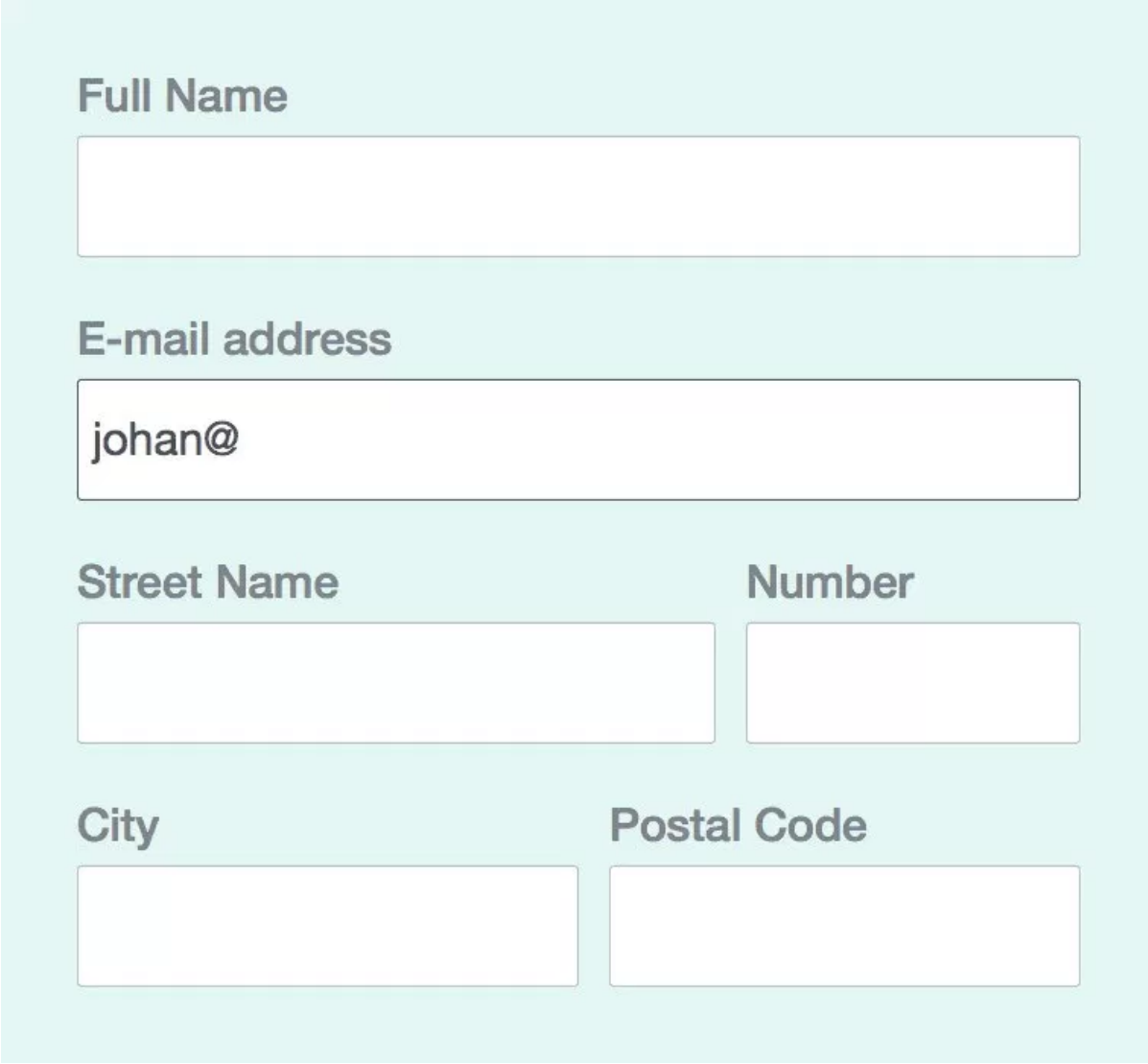


Illustration 2 : utilisez plutôt une taille de police de 14px et plus



3. Tu devras prévoir des zones facilement sélectionnables

De nos jours, il est probable que votre formulaire sera utilisé sur un terminal mobile. Apple recommande des boutons de 44px par 44px car cela correspond à peu près à la taille d'un doigt. C'est une bonne recommandation mais il est inutile de la suivre de manière trop stricte car vous risquez de vous retrouver avec des champs qui peuvent être trop grands dans une application classique. Il est recommandé de choisir une taille comprise entre 32 et 40px de hauteur.

Le champ par défaut dans Bootstrap 3 est de 32px, c'est une bonne taille de base. Les champs plus petits risquent d'être difficilement sélectionnables.

Commentaire personnel : si pour une raison précise, vous désirez malgré tout implémenter des champs plus petits sur votre site web par exemple, il est recommandé de prévoir une version adaptée aux mobiles. Tapoter est bien moins précis que cliquer avec une souris et il serait dommage de

décourager des utilisateurs.
Si leur Expérience sur votre interface mobile est décevante et décourageante, il y a une probabilité qu'ils abandonnent et négligent également la version desktop du site ou de l'application.

Illustration 1 : les champs trop petits sont plus difficilement sélectionnables

Illustration 2 : il est préférable de prévoir des zones plus facile à sélectionner avec le doigt

4. Tu devras dimensionner les champs de saisie selon la longueur attendue

Un formulaire qui présente des champs d'une longueur en accord avec le contenu prévu est plus facile à lire.

Par exemple, pour le champ d'un code postal, ne mettez pas une largeur de 100%. Réfléchissez au nombre minimum de caractères d'un code postal (4 en Belgique, 6 aux pays bas). Il faut que le champ corresponde à un code de 6 caractères, en utilisant la police de votre design mais aussi la police de substitution.

Commentaire personnel : je mettrais un bémol sur point spécifique. Il est difficile d'anticiper et de connaître la longueur de tous les codes postaux. Ils peuvent être très longs, à Londres, un code postal

ressemble à "EC1A 1AH", ceux de Rio de Janeiro sont comme ceci "20000-xxx".

Gardez en mémoire que, sur Linux, les polices sans-serif par défaut sont typiquement larges. Cela a du sens de tester vos designs avec une police large comme Verdana, et cela même si vous avez choisi une police compressé pour votre design. Sur le web, vous ne savez jamais exactement ce que l'utilisateur verra à l'écran.

Commentaire personnel : il est effectivement important de ne pas oublier que l'affichage peut grandement varier en fonction des versions et des navigateurs, de si l'utilisateur a activé javascript ou pas, etc... Même s'il est quasiment impossible de prévoir un affichage idéal sur tous les écrans, il est possible par quelques astuces simples, de limiter les surprises ; tester ses designs avec des polices par défaut en est une.

Illustration 1 : la structure est uniforme, ce formulaire est plus difficile à "scanner"

Illustration 2 : des champs à la taille attendue par l'utilisateur

5. Tu ne devras pas customiser les "checkboxes" et les bouton "radio"

Historiquement, les boutons "radio" ou les "checkboxes" sont personnalisés en utilisant du javascript. De nos jours, il est possible de le faire en utilisant des SVG ou du CSS.

Toutefois, personnaliser l'apparence d'un champ peut entraîner une utilisation perturbée de la navigation au clavier. 90% des formulaires avec des checkboxes ou des boutons radio customisés n'ont pas de styles définis pour les différents états et sont, de ce fait, compliqués à utiliser au clavier.

Tout le monde apprécie un beau formulaire et de jolies animations SVG mais vous ne rendez pas services à l'utilisateur en personnalisant des éléments en n'y apportant aucune valeur ajoutée. Et personnaliser ces

éléments seulement pour des raisons esthétiques n'apporte pas un plus suffisant

Par exemple, des widgets comme select2 ou des "date pickers" (sélecteurs de date) peuvent représenter une valeur ajoutée à un formulaire là où les éléments par défaut ne suffisent pas.

Commentaire personnel : il est certain que garder les éléments par défaut est gage de sécurité. Toutefois, je ne vois pas, pour ma part, de problème à personnaliser des checkboxes ou des boutons radio. Si cela est fait proprement en CSS, pour chaque état il n'y a pas spécialement de contre-indication. Ce commandement me semble exagérément restrictif.

Illustration 1 : des éléments incomplets ou pas assez clairs peuvent être perturbants

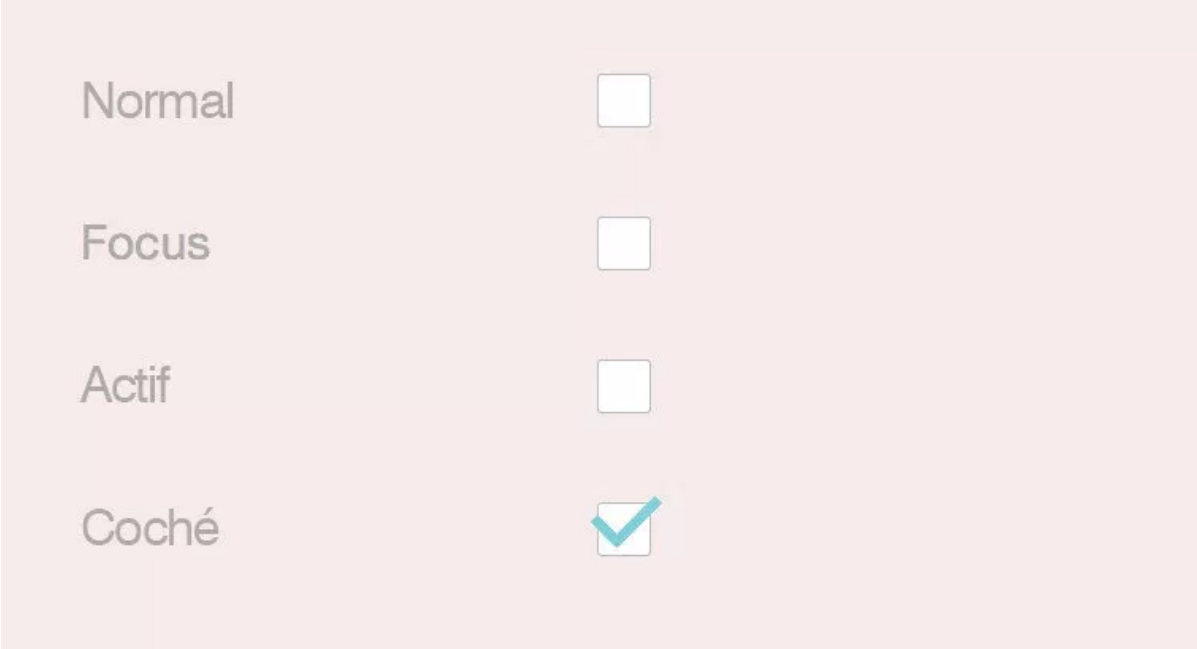
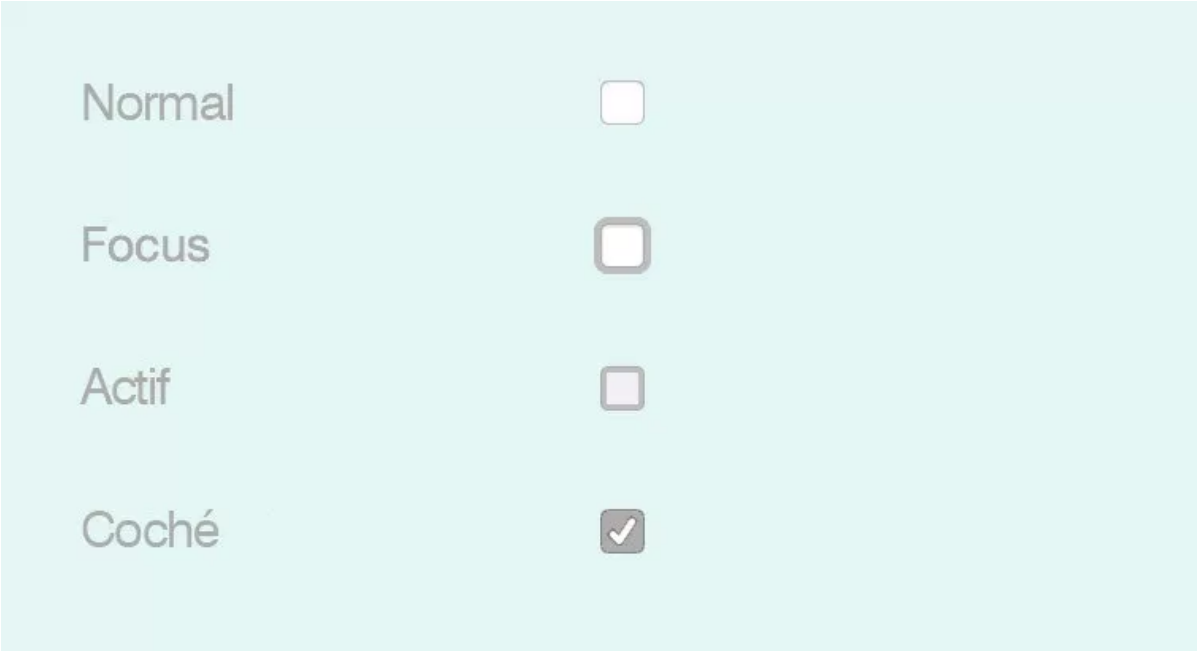


Illustration 2 : il vaut mieux utiliser les éléments par défaut du navigateur



6. Tu devras prévoir des messages d'erreur par champ et un message général

C'est une bonne idée de placer un message d'erreur général en plus de messages spécifiques à chaque champ. Quand un formulaire est validé, l'utilisateur verra la page rechargée avec une erreur mais ne saura pas exactement où elle est et si elle sur la partie visible de l'écran ou non.

Mettez un message d'erreur en haut du formulaire (et assurez vous que celui-ci soit visible sur desktop et mobile) mais vous pouvez également ajouter un message spécifique au niveau du champ ou l'erreur a été commise.

Commentaire personnel : indiquer un erreur est l'information la plus importante du formulaire car cela vous empêche d'aller plus loin. Il faut indiquer au visiteur de la manière la plus claire comment corriger son erreur pour pouvoir avancer.

Illustration 1 : ce formulaire comporte une erreur mais elle n'est pas la partie visible de l'écran