

Beste de savoir

Faciliter la navigation au clavier
avec :focus, outline et box-shadow

19 avril 2021

Table des matières

1. Styliser le focus des liens avec outline	1
2. Des boutons arrondis avec une bordure extérieure au focus	3
3. Bonus : supprimer les pointillés de Firefox	5

Beaucoup de sites utilisent les mêmes styles au survol avec une souris (via `:hover`) que lors de la navigation au clavier (via `:focus`)¹.

Or cela n'a **pas le même impact** et **ne répond pas au même besoin**:

- En naviguant avec une souris, l'utilisateur sait où est son pointeur et cherche à confirmer qu'un élément est réactif grâce aux styles du `:hover`
- En naviguant au clavier l'utilisateur cherche plutôt à aller au prochain élément réactif sans forcément savoir où celui-ci se trouve.



Il faut donc **rendre le `:focus` facile à localiser** pour l'utilisateur.

Bien que les navigateurs fournissent des styles par défaut, ceux-ci ne sont pas toujours très adaptés (bordure très fine, parfois sans contraste, au style parfois pas très cohérent, etc.). À vous, donc, de **fournir une solution adaptée** tout en vous assurant que le rendu soit cohérent avec votre charte graphique.



Certaines personnes ont l'habitude de supprimer les styles par défaut des navigateurs via `outline: none` sans fournir de solution alternative. Ceci rend la navigation au clavier impossible (et par conséquent votre site moins accessible). Ne faites jamais ça !

Je vais donc vous partager quelques astuces pour mettre en place des liens et des boutons facilement utilisables au clavier sans casser votre superbe design. 🍊

1. Styliser le focus des liens avec outline

Imaginons le HTML suivant, une simple phrase avec un lien :

```
1 <p>
2   Connaissez-vous <a href="https://zestedesavoir.com">Zeste de
   Savoir</a> ?
```

1. Je vous rassure : j'ai également été coupable de cette flemmardise pendant des années !

1. Styliser le focus des liens avec outline

```
3   C'est un super site !
4 </p>
```

Lorsque l'on navigue au clavier, le style par défaut n'est pas très notable, il se voit assez peu:

Connaissez-vous [Zeste de Savoir](#)? C'est un super site !

FIGURE 1.1. – Aperçu du rendu par défaut du focus avec Firefox

Commençons avec l'option la plus simple: [outline](#) ↗ .

```
1 a:focus {
2   outline: 3px solid red;
3 }
```

On a donc une bordure qui s'affiche sans impacter les dimensions de notre élément (c'est d'ailleurs un des avantages de la propriété `outline` sur `border` qui a une syntaxe similaire).

Connaissez-vous [Zeste de Savoir](#)? C'est un super site !

FIGURE 1.2. – Aperçu du rendu avec un outline de 3px rouge sous Firefox

Mais c'est pas très beau d'avoir un si gros cadre collé au texte... que faire?

Eh bien on utilise [outline-offset](#) ↗ !

```
1 a:focus {
2   outline: 3px solid red;
3   outline-offset: 4px;
4 }
```

Et voilà, lors de la navigation au clavier nos liens sont mis en valeur sans casser le design avec un cadre :

Connaissez-vous [Zeste de Savoir](#)? C'est un super site !

FIGURE 1.3. – Aperçu avec 'outline-offset'

2. Des boutons arrondis avec une bordure extérieure au focus

Et **bonus** : ça s'adapte automatiquement aux saut de lignes !

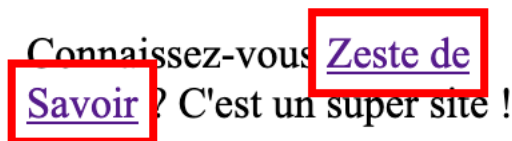


FIGURE 1.4. – Aperçu avec un saut de ligne dans le lien

?

Oui mais si j'ai des boutons avec un arrondi, comment je fais puisque `outline` n'a pas d'option `radius` ?

2. Des boutons arrondis avec une bordure extérieure au focus

Je suppose que la création d'un bouton ne vous pose pas de problème :

```
1 <button>Un bouton</button>
```

On peut en profiter pour le styliser un peu, en lui arrondissant les angles...

```
1 button {  
2   appearance: none;  
3   font-size: inherit;  
4   padding: 1em;  
5   border: 0;  
6   background-color: orange;  
7   border-radius: 2em;  
8 }
```



FIGURE 2.5. – Aperçu du bouton orange, sans effet de survol ou focus

Seulement voilà: au focus le contour par défaut n'est pas de la bonne couleur et surtout ne suit pas la forme du bouton puisque `outline` n'a pas d'option `radius`.

2. Des boutons arrondis avec une bordure extérieure au focus

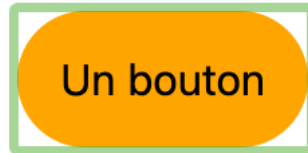


FIGURE 2.6. – Aperçu du bouton avec le style de focus par défaut de Firefox

Admettons que l'on applique les mêmes styles au `:hover` comme au `:focus` :

```
1 button: hover,  
2 button: focus {  
3     outline: none; /* On supprime le contour par défaut du  
         navigateur */  
4     box-shadow: 2px 3px 7px rgba(0,0,0,2); /* On met une petite  
         ombre grise */  
5 }
```

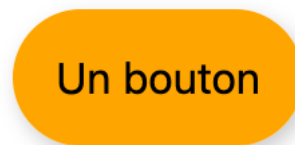


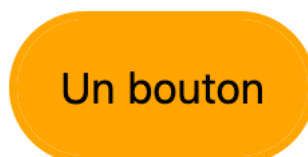
FIGURE 2.7. – Aperçu du bouton avec le style du survol

Comme vous le voyez, ça peut être suffisant au survol (quoique l'ombre n'est pas très visible), mais si on ne sait pas où est le focus actuel sur la page... difficile de s'y retrouver.

Heureusement, `outline` n'est pas la seule propriété qui permet de dessiner des contours sans impacter les dimensions des éléments !

```
1 button: focus {  
2     outline: none;  
3     box-shadow: 0 0 0 3px orange;  
4 }
```

On profite ici de la propriété `spread-radius` de l'ombre [↗](#) pour la rendre solide.



3. Bonus : supprimer les pointillés de Firefox

FIGURE 2.8. – Aperçu du bouton avec une ombre de 3px au focus

Vous ne voyez pas la différence? Ah oui, l'ombre est collée au bouton!

Malheureusement il n'y a pas d'option d'`offset` pour les ombres... c'est fâcheux.

Mais il y a une astuce ! Si notre fond est blanc, et que l'on applique une première ombre blanche avant celle qui est orange... on peut simuler un décalage !

```
1 button:focus {  
2     outline: none;  
3     /*  
4     Une première ombre fine (largeur = décalage) de la couleur du  
5     fond, une deuxième pour notre cadre (largeur = décalage +  
6     cadre)  
7     */  
8     box-shadow: 0 0 0 2px white, 0 0 0 5px orange;  
9 }
```



FIGURE 2.9. – Aperçu du bouton avec focus et deux ombres (une fine blanche, une large orange)

3. Bonus : supprimer les pointillés de Firefox

Si vous testez sous Firefox, vous avez pu remarquer que par défaut apparaissent des pointillés à l'intérieur de votre bouton sous Windows. Il ne s'agit pas d'un `outline` par défaut, mais d'un pseudo élément propre à Firefox qui affiche une bordure personnalisée.

Vous pouvez le supprimer avec le sélecteur `::-moz-focus-inner`:

```
1 button::-moz-focus-inner {  
2     border: 0;  
3 }
```



Supprimez cette bordure **uniquement** après avoir mis en place une alternative suffisante !

Et voilà, avec quelques lignes de CSS vous avez rendu votre site plus facile à utiliser au clavier !

3. Bonus : supprimer les pointillés de Firefox

Voici au passage un aperçu de ce que cela peut donner avec un peu plus de styles autour :

!(<https://jsfiddle.net/wiki53/y721c0jx/54/>)

i

Note : Pour tester sous macOS il faut activer la navigation au clavier

1. Préférences Système
2. Clavier
3. Raccourcis
4. Utiliser la navigation au clavier pour déplacer la cible sur les différents commandes

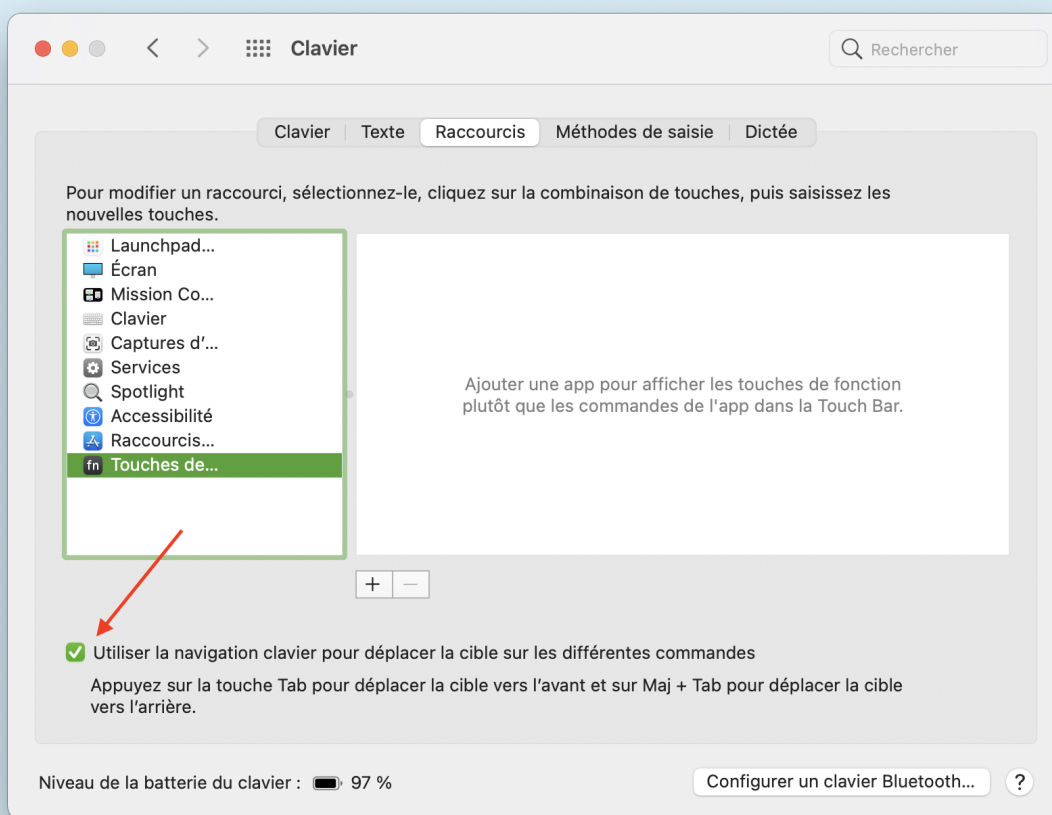


FIGURE 3.10. – Capture d’écran des réglages du clavier sous macOS

N’oubliez pas de relancer votre navigateur par la suite pour prendre en compte ce nouveau réglage.

Ce tutoriel a été inspiré par [cette vidéo de Kevin Powell](#) [↗](#), qui m’a rappelé que j’avais fait cette erreur sur mon propre site (oups !).