

[tutoriel](#)

# Numérotation automatique avec des compteurs CSS

Pour la rédaction de documents, il est souvent utile de numéroter les sections et d'avoir une table des matières.

Le protocole CSS 2.1 automatise la génération de numéros à l'aide de compteurs CSS, et cet article vous guidera dans leur utilisation.

## Pré-requis

### Première étape

La première étape consiste à réinitialiser le compteur et à le nommer.

Pour cela, on peut utiliser la propriété `counter-reset`, comme dans l'exemple suivant :

```
body {  
    counter-reset: section;  
}
```

Le compteur est remis à zéro pour l'élément `<body>`, et le compteur a pour identifiant «section». Ce nom est à votre convenance.

La propriété peut également prendre une deuxième valeur optionnelle qui définit la valeur de départ du compteur.

Voici comment faire pour que le compteur commence à compter à partir de 5 :

```
body {  
    counter-reset: section 4;  
}
```

Cette valeur est fixée à 4 car le compteur est toujours incrémenté juste avant de l'afficher. Si vous le mettez à 4, la valeur passe à 5 avant l'affichage.

## Autres étapes

### Incrémentation du compteur

Vous devez préciser quand le compteur augmente et de quelle valeur.

La propriété **counter-increment** permet de le spécifier.

La valeur par défaut de l'incrément est de 1 :

```
body {
    counter-reset: section 4;
}
h2 {
    counter-increment: section;
}
```

Si vous préférez l'incrémenter de deux, spécifiez-le :

```
h2 {
    counter-increment: section 2;
}
```

La propriété autorise aussi les valeurs 0 ou négatives.

Si le compteur est remis à zéro puis incrémenté sur le même élément, il est d'abord remis à zéro puis incrémenté de la valeur spécifiée, donc dans l'exemple suivant, la valeur sera toujours 2 :

```
h2 {
    counter-reset: section;
    counter-increment: section;
}
```

## Affichage du compteur

Une fois le compteur en place, il faut l'afficher.

Cela peut être fait avec le contenu généré en utilisant la fonction counter, et la pseudo classe :before :

```
body {
    counter-reset: section 4;
}
h2 {
    counter-increment: section;
}
h2:before {
    counter(section) ' ';
}
```

Si vous utilisez un compteur et que vous l'incrémentez sur le même élément, comme dans les exemples ci-dessus sur l'élément <h2>, il est alors incrémenté avant d'être utilisé.

## Utilisation de deux ou plusieurs compteurs

La fonction `counter` peut être utilisée plusieurs fois sur la propriété `content`.

Dans l'exemple suivant, j'utilise deux compteurs pour la numérotation des sections et des sous-sections.

Le compteur de sous-sections est remis à zéro pour chaque nouvelle section.

```
body {  
    counter-reset: section;  
}  
h2 {  
    counter-increment: section;  
    counter-reset: sub-section;  
}  
h2:before,  
h3:before {  
    content: counter(section) ' ' counter(sub-section) ' ';  
}  
h3:before {  
    counter-increment: sub-section;  
}
```

## Utilisation de compteurs imbriqués

La fonction `counters()` permet de spécifier les compteurs imbriqués.

Cela fonctionne bien pour des éléments tels que les listes, où les listes peuvent être imbriquées les unes dans les autres.

Cela ne fonctionnerait pas pour l'exemple ci-dessus car les sous-sections ne sont pas imbriquées dans les sections principales, mais sont plutôt des éléments apparentés.

Les compteurs imbriqués peuvent être spécifiés comme suit :

```
ol {  
    counter-reset: section;  
    list-style-type: none;  
}  
ol li {  
    counter-increment: section;  
}  
ol li:before {  
    content: counters(section, ' ') ' ';  
}
```

Le compteur de chaque élément de liste sera ainsi incrémenté, quel que soit le nombre de listes imbriquées dans le balisage.

Il est important de remettre le compteur au bon endroit, pour qu'il se remette à zéro pour chacun des éléments imbriqués.

Dans ce cas, je réinitialise le compteur sur l'élément `<ol>`.

Dans l'exemple suivant, la première liste imbriquée commencera à 1.1 et comptera jusqu'à 1.2, 1.3 et ainsi de suite.

La deuxième liste imbriquée commencera à 1.1.1 et comptera jusqu'à (1.1.2 et ainsi de suite)

```
<ol>
  <li>Item 1
    <ol>
      <li>Sub item 1
        <ol>
          <li>Sub-sub item 1</li>
          <li>Sub-sub item 2</li>
          <li>Sub-sub item 3</li>
        </ol>
      </li>
      <li>Sub item 2</li>
    </ol>
  </li>
  <li>Item 2</li>
</ol>
```

## Styliser les compteurs

Il est possible de mettre en forme le compteur en mettant un deuxième argument dans les fonctions du compteur.

Les valeurs sont les mêmes que pour la propriété de type liste.

Bien que toutes les valeurs soient possibles, pour les compteurs, seul un sous-ensemble a un sens, car les valeurs telles que disque, carré et cercle restent les mêmes quelle que soit la valeur du compteur.

Le style peut être spécifié comme suit :

```
ol li:before {
  counter(answer, lower-alpha) ' ' ;
}
```

## Conclusion

## Problèmes connus

## Voir aussi

- (fr) <https://dev.opera.com/articles/automatic-numbering-with-css-counters/>

---

Basé sur « [Automatic Numbering With CSS Counters](#) » par David Storey.

From:

<https://nfrappe.fr/doc-0/> - **Documentation du Dr Nicolas Frappé**

Permanent link:

<https://nfrappe.fr/doc-0/doku.php?id=tutoriel:internet:dokuwiki:titres:number:start>



Last update: **2022/08/13 22:27**