



# Cours d'intégration

## 3W Academy

### CSS 8. L'héritage et la cascade

---

#### [8. L'héritage et la cascade](#)

##### [8.1. L'héritage](#)

##### [8.2. La cascade](#)

###### [8.2.1. L'ordre dans la source](#)

###### [8.2.2. La spécificité \(ou priorité\)](#)

###### [8.2.3. L'importance](#)

## 8. L'héritage et la cascade

### 8.1. L'héritage

L'héritage repose sur l'arborescence des éléments au sein du document. Une propriété héritable sera transmise automatiquement du parent à ces enfants. Si la même propriété est redéfinie à un des enfants, alors c'est la seconde valeur qui l'emporte.

Par exemple. La propriété `color` définie sur l'élément `body` sera héritée de proche en proche et appliquées à tous les enfants de `body`. Si un article voit sa propriété `color` redéfinie, alors l'article et les éléments enfants de l'article auront la couleur définie au niveau de l'élément article et non celle définie au niveau du `body`.



**Attention** Seules, certaines propriétés sont héritables par défaut, il s'agit essentiellement des propriétés concernant la typographie (texte, fonte et couleur).

Il est possible de contrôler l'héritage avec les valeurs suivantes:

<code>inherit</code>	la propriété aura la même valeur que l'élément parent.
<code>initiale</code>	la propriété aura la valeur initiale prévue par le navigateur
<code>unset</code>	la propriété aura sa valeur naturelle ( <code>inherit</code> si naturellement héritable ou <code>initiale</code> sinon)
<code>revert</code>	annule toute valeur appliquée dans la feuille de style

### 8.2. La cascade

La cascade est une notion essentielle du CSS (Cascading Style Sheets / feuilles de style en cascade). C'est l'ensemble des règles qui permet de déterminer quelle valeur sera appliquée à un élément si celle-ci est définie à plusieurs endroits dans la feuille de style. Combinée à l'héritage, cela permet de comprendre quelle valeur sera appliquée finalement à un élément.

La cascade dépend de trois facteurs du plus prioritaire au moins prioritaire:

1. l'importance

2. la spécificité (ou priorité)
3. l'ordre dans la source

### 8.2.1. L'ordre dans la source

C'est le facteur le moins fort de la cascade. Si deux styles ayant la même spécificité (voir section suivante) sont en conflit, alors c'est l'ordre de déclaration dans le code source qui s'applique.

Par exemple:

Dans le CSS

```
.message {  
    padding: 1em;  
    border: 1px solid;  
}  
.info {  
    color: blue;  
}  
.warning {  
    color: orange;  
}
```

Dans le document HTML

```
<p class="message warning info">  
    ...  
</p>
```

Dans cet exemple, le paragraphe aura la couleur orange car la déclaration de la classe `.warning` est postérieure à celle de la classe `.info` bien que "info" soit noté après "warning" dans l'attribut.



**Attention** C'est l'ordre de déclaration dans la feuille de style qui compte pas l'ordre des classes au sein de l'attribut `class` de l'élément

### 8.2.2. La spécificité (ou priorité)

C'est le facteur intermédiaire de la cascade mais c'est le plus souvent ce facteur qui est déterminant.

Si plusieurs styles entrent en conflit, c'est le sélecteur ayant la spécificité la plus grande qui l'emporte. Les règles sont les suivantes:

- Une classe (ou pseudo-classe) est toujours plus spécifique qu'un élément (ou pseudo-élément).
- Un ID est toujours plus spécifique qu'une classe
- Un style déclaré dans l'attribut style est toujours plus spécifique qu'un ID.

Pour évaluer facilement la spécificité d'un sélecteur, on peut utiliser le système suivant:

- Chaque élément (ou pseudo-élément) vaut 1 point
- Chaque classe (ou pseudo-classe) vaut 10 points
- Chaque ID vaut 100 points
- Chaque déclaration dans l'attribut style vaut 1000 points

Par exemple:

Dans le CSS

```
#outer p {
    color: cyan;
}
#outer #inner p {
    color: violet;
}
#inner.colored p {
    color: red;
}
article.colored .info {
    color: blue;
}
article:first-child p {
    color: green;
}
section article p {
    color: black;
}
```

Dans le document HTML

```
<section id="outer">
  <article id="inner" class="colored">
    <p class="info">...</p>
  </article>
</section>
```

Calcul de la spécificité:

Sélecteur	Style	ID	Classe	Elément	Total
-----------	-------	----	--------	---------	-------

#outer p	0	1	0	1	101
#outer #inner p	0	2	0	1	201
#inner.colored p	0	1	1	1	111
article.colored .info	0	0	2	1	21
article:first-child p	0	0	1	2	12
section article p	0	0	0	3	3

Le paragraphe est de couleur "violet".



**Attention** Cette méthode simple de calcul ne représente pas fidèlement la réalité: 10 classes ne valent pas un ID, l'ID est toujours plus spécifique.

### 8.2.3. L'importance

C'est le facteur le plus fort de la cascade. L'importance se définit via un élément syntaxique particulier, la déclaration *!important*.

En effet, il est possible de forcer une valeur en lui ajoutant la déclaration *!important* à la fin de la valeur.

Dans le CSS

```
#banner {
    color: blue;
}
.colored {
    color: green;
}
article {
    color: red !important;
}
```

Dans le document HTML

```
<article id="banner" class="colored">
    ...
</article>
```

Dans cet exemple, l'article aura la couleur rouge car la déclaration *!important* est placée derrière la valeur "red".



**Attention** Le seul moyen de modifier la valeur ayant une déclaration *!important* est d'ajouter une autre déclaration *!important* avec les autres facteurs de la cascade plus forts (particularité ou ordre dans la source).

En conclusion, il est fortement recommandé de ne pas utiliser cette déclaration.