

# L'héritage en CSS

## 1. Qu'est-ce que l'héritage en CSS ?

L'héritage en CSS est un mécanisme qui permet à certains styles appliqués à un parent de se transmettre automatiquement à ses enfants. Cela simplifie l'écriture du code en évitant de répéter les mêmes styles pour chaque élément.

Cependant, toutes les propriétés CSS ne sont pas héritées. L'héritage est limité principalement aux propriétés relatives au texte.

### 1.1. Propriétés héritées par défaut

Les propriétés liées au texte sont automatiquement héritées :

- `color` (couleur du texte)
- `font-family` (police de caractère)
- `font-size` (taille de police)

### 1.2. Propriétés non héritées par défaut

Les propriétés comme les marges, les bordures ou les dimensions ne sont **pas héritées** par défaut. Exemple :

- `margin`
- `padding`
- `border`
- `background`

## 2. Forcer ou empêcher l'héritage

### 2.1. Utiliser `inherit`

Le mot-clé `inherit` permet de forcer une propriété à hériter de la valeur de son parent, même si elle n'est pas héritée par défaut.

HTML :

```
<div>
  <p class="bold">Texte gras</p>
  <p>Texte normal</p>
</div>
```

CSS :

```
div {
  font-weight: bold;
}
p {
  font-weight: inherit;
}
```

Le second paragraphe hérite de la propriété `font-weight` de son parent malgré l'absence d'héritage par défaut.

### 2.2. Utiliser `initial`

Le mot-clé `initial` rétablit la valeur par défaut de la propriété, indépendamment des styles appliqués au parent.

HTML :

```
<div>
  <p class="default">Texte normalisé</p>
</div>
```

CSS :

```
p {
  color: red;
}
.default {
  color: initial;
}
```

Le texte de la classe "default" retrouvera la couleur standard (généralement noire), ignorante du `color` rouge appliqué au `<p>` parent.

### 2.3. Utiliser `unset`

Le mot-clé `unset` supprime toute valeur appliquée. Si la propriété est héritée par défaut, elle retournera à son état hérité. Sinon, elle reprendra sa valeur initiale (`initial`).

HTML :

```
<div>
<p>Texte rouge</p>
</div>
```

CSS :

```
div {
  color: red;
}
p {
  color: unset;
}
```

Le `<p>` perd la couleur rouge héritée et revient à la valeur par défaut (noire).

### 3. Exemple pratique

HTML :

```
<div class="container">
<h1>Héritage en CSS</h1>
<p>Ce texte hérite des styles de son parent.</p>
<p class="non-herite">Ce texte a une couleur différente.</p>
</div>
```

CSS :

```
.container {
  color: blue;
  font-family: Arial, sans-serif;
}
.non-herite {
  color: initial;
}
```

Dans cet exemple :

- Le texte des `<p>` hérite de la couleur `blue` du parent ( `.container` ) sauf pour celui ayant `.non-herite` .
- La police `Arial` est appliquée partout car elle est héritée.

### 4. À RETENIR

- Les propriétés héritées par défaut concernent principalement le texte :
  - `color` , `font-family` , `font-size` , etc.
- Les **propriétés non héritées par défaut** nécessitent une intervention explicite pour être transmises.
- Pour gérer l'héritage :
  - `inherit` : force l'héritage.
  - `initial` : restaure la valeur CSS par défaut (comme défini par le navigateur).
  - `unset` : mélange entre les deux, dépend si la propriété est héritée par défaut ou non (hérité sinon initialisé).

Résumé des mots-clés :

Mot-clé	Effet
inherit	Forcer la propriété à hériter de la valeur parent.
initial	Réinitialise avec la valeur définie par défaut (CSS standard).
unset	Hérité si possible, sinon valeur par défaut comme initial.