

# CSS : Présentation

## Contenu

Objectifs des feuilles de style .....	2
Cascade et héritage .....	3
Le principe de cascade .....	3
L'héritage.....	3
Le mélange des règles .....	3
L'héritage entre propriétés .....	4
Déclaration d'une feuille de style.....	5
Définir des styles dans la balise HTML .....	5
Définir les styles dans <head> .....	5
Définir un style global dans un fichier CSS séparé .....	6
L'essentiel à retenir .....	8

## Objectifs des feuilles de style

Les feuilles de style CSS (Cascading Style Sheet) complètent HTML, en offrant un langage permettant de définir les propriétés graphiques des balises.

Le CSS permet de gérer 3 points :

- L'agencement des balises HTML dans votre page (positionnement)
- La typographie : polices de caractères, taille du texte
- L'habillage graphique : couleurs, bordures, arrière-plans

Les styles peuvent s'appliquer à une ou plusieurs balises. Ceci permet de définir un style une seule fois, et de l'appliquer autant de fois que nécessaire, sans avoir à le redéfinir.

Concrètement, le code CSS est constitué d'un « sélecteur » suivi d'un bloc d'instructions entre accolades :

```
selecteur
{
    instructions CSS
}
```

Exemple :

```
h1
{
    color: red;
}
```

Cet exemple basique permet d'appliquer la couleur rouge au texte de la balise `<h1>`.

Comme on peut l'observer dans l'exemple, une ligne d'instruction CSS se termine toujours par un point-virgule.

Il peut y avoir bien sûr plusieurs lignes d'instructions dans un bloc :

```
/* CSS */
h1
{
    color: red;
    border: 2px solid blue;
    background-color: yellow;
    font-family: verdana;
}
```

# Cascade et héritage

Les feuilles de style CSS (Cascading Style Sheet) sont basés sur 2 principes de base : la cascade et l'héritage.

## Le principe de cascade

La mise en forme finale d'un élément peut être obtenue grâce à plusieurs règles présentes à plusieurs endroits et qui interagissent de façon complexe. C'est cette complexité qui rend CSS si puissant mais c'est également une source de confusion et de difficultés lorsqu'il s'agit de résoudre des problèmes.

Plusieurs sources d'informations principales forment cette cascade. Ces sources sont :

- Les styles par défaut du navigateur pour le langage à balise utilisé
- Les styles définis par l'utilisateur qui consulte le document
- Les styles liés au document par l'auteur (en HTML, cela correspond aux éléments `<link>` qui pointent vers les feuilles de style externes)
- Les styles définis par l'auteur via la balise `<style>` dans l'élément `<head>`
- Les styles définis par l'auteur qui sont directement appliqués depuis les éléments (en HTML, cela correspond à utiliser l'attribut `style`)

Chaque source **surcharge** la ou les précédentes, dans cet ordre. Pour l'ordre de priorité de la cascade, on parle de **poids** des règles de styles.

## L'héritage

L'héritage s'applique de deux façons et chacune a son importance : le mélange des règles et l'héritage entre propriétés.

### Le mélange des règles

Lorsque plusieurs règles correspondent à un même élément, elles s'appliquent toutes à cet élément. Nous avons vu avec la cascade comment une règle en surcharge une autre. Cette surcharge ne concerne que les propriétés en conflit, sinon, toutes les règles sont appliquées.

```
<!-- Code HTML -->
<p>J'adore le <strong>CSS</strong></p>
```

```

/* Code CSS */
p strong {
    background-color: khaki;
    color: green;
}

strong {
    text-decoration: underline;
    color: red;
}

```

À cause de son poids, la première règle surcharge la propriété `color` de la deuxième règle. Toutefois, la couleur d'arrière-plan fournie par la première règle et la décoration fournie par la seconde s'appliquent à la balise `<strong>`. Le texte est mis en gras – comportement natif de `<strong>` - grâce aux styles par défaut du navigateur.

## L'héritage entre propriétés

Certaines propriétés appliquées à un élément peuvent hériter de la valeur d'un des éléments parents. Cela permet par exemple de déterminer la valeur qui sera appliquée pour la propriété d'un élément quand aucune n'est fournie (la valeur du parent se « propage » à l'élément enfant). L'héritage permet d'avoir à éviter de définir toutes les règles pour tous les éléments à chaque fois.

L'héritage peut avoir lieu pour toutes les propriétés mais certaines sont héritées naturellement (par exemple `color` ou `font-family`). La valeur basique, par défaut, est définie par la feuille de style du navigateur, toutes les propriétés qui héritent naturellement prendront cette valeur par défaut si rien d'autre n'est spécifié.

CSS fournit trois valeurs spéciales pour gérer l'héritage :

Valeur	Rôle
<code>inherit</code>	cette valeur indique que la valeur de la propriété de l'élément sera la même que la valeur de la propriété de l'élément parent (c'est la valeur par défaut pour tous les éléments dont l'héritage est naturel).
<code>initial</code>	cette valeur indique que la valeur de la propriété de l'élément sera la valeur définie par la feuille de style du navigateur (si la valeur n'est pas définie par la feuille de style du navigateur et que l'héritage de cette propriété est naturel, la valeur définie sera alors une copie de l'héritage naturel plutôt qu' <i>inherit</i> . Les effets de bords engendrés peuvent être plutôt subtils et nous ne les verrons pas ici).
<code>unset</code>	cette valeur réinitialise la propriété avec sa valeur naturelle. Cela signifie que si la propriété est héritée naturellement, on aura le même effet qu' <i>inherit</i> , sinon on aura le même effet qu'avec <code>initial</code> .

## Déclaration d'une feuille de style

Il existe plusieurs méthodes pour intégrer des feuilles de style CSS à une page web.

△ Rappelez-vous cependant de l'ordre de priorité établi par le principe de cascade.

### Définir des styles dans la balise HTML

La déclaration d'un style CSS peut se faire directement dans une balise HTML via l'attribut `style`. Toutefois, c'est désormais non recommandé, la bonne pratique étant de mettre tous les styles dans des fichiers externes.

```
<!-- Code HTML -->
<html>
<head>
  <title>Titre du fichier</title>
</head>
<body>
  <h1 style="color:blue;">...</h1>
</body>
</html>
```

### Définir les styles dans <head>

La déclaration d'un style peut se positionner dans l'en-tête `<head>` du document, grâce à la balise `style`.

```
<!-- Code HTML -->
<html>
<head>
  <style type="text/css">
    /* ... ici sont définis les formats ... */
  </style>
</head>
<body>...</body>
</html>
```

**A retenir :** la valeur par défaut de l'attribut `type` étant « `type/css` », cet attribut est optionnel. On peut donc simplement écrire :

```
<style>
/* ... ici sont définis les formats ... */
</style>
```

➔ [Documentation de la balise <style>](#)

## Définir un style global dans un fichier CSS séparé

On peut définir les styles dans un fichier texte (.css) séparé qu'il faudra appeler dans les fichiers HTML que l'on veut. Toute mise à jour du fichier .css ira se répercuter sur toutes les pages HTML utilisant cette feuille de style. **C'est la bonne pratique recommandée** (séparation des données et de la présentation) et donc à appliquer.

L'appel du fichier .css s'effectue à l'aide de la balise `<link>` dans la partie `<head>` de la page web :

```
<!-- Code HTML -->
<html>
<head>
  <title>Titre du fichier</title>
  <link rel="stylesheet" href="fichier.css">
</head>
<body>
...
</body>
</html>
```

Il est possible de combiner l'appel à un fichier .css externe et la définition d'un style local avec la balise `<style>`. Dans ce cas, les formats directement définis dans `<style>` deviennent **prioritaires**.

**A retenir :** depuis HTML 5, la balise `<link>` ne prend plus de slash de fermeture « `/>` » et l'attribut `type` est devenu obsolète, donc pas de `type='text/css'`.

L'attribut `href` doit contenir le chemin complet vers le fichier css :

```
<!-- Code HTML -->
<html>
<head>
  <title>Titre du fichier</title>
  <link rel="stylesheet" href="styles/fichier.css">
</head>
<body>
...
</body>
</html>
```

Lorsqu'on parle de chemin, il peut s'agir d'une URL (on appelle donc un fichier CSS hébergé sur un autre serveur) :

```
<!-- Code HTML -->
<html>
<head>
  <title>Titre du fichier</title>
```

```
<link rel="stylesheet" href="http://www.lesite.fr/style/fichier.css">
</head>
<body>
...
</body>
</html>
```

Il est possible d'appeler plusieurs fichiers .css dans une page web :

① Pas de limite de nombre, mais plus ils seront nombreux plus la page sera longue à charger.

```
<!-- Code HTML -->
<html>
<head>
  <title>Titre du fichier</title>
  <link rel="stylesheet" href="fichier1.css">
  <link rel="stylesheet" href="styles/fichier2.css">
  <link rel="stylesheet" href="http://www.autre-
site.fr/assets/fichier3.css">
</head>
<body>
...
</body>
</html>
```

➔ [Documentation de la balise <link>](#)

## L'essentiel à retenir

- Le CSS permet de structurer et d'habiller le code HTML d'une page web
- Le CSS répond à des principes de surcharges : cascade et héritage.
- Le code CSS s'applique via des sélecteurs suivi d'instructions dans un bloc entre accolades
- Il y a 3 manières de déclarer ses styles CSS dans une page web, mais la bonne pratique est de tout mettre dans un fichier externe (ou plusieurs).