

Programmation Web

Le CSS

Pierre Blarre - Florian Rodriguez

Cascading Style Sheets

- Séparer contenu et présentation
- <https://www.csszengarden.com>

Définition d'un style

style

```
selecteur {  
  déclaration  
  déclaration  
}
```

Copy

déclaration

```
propriété:valeur;
```

Copy

Exemple

```
h1 {  
  color: red;  
  font-size: 5em;  
}
```

Copy

Association document HTML - CSS

- Feuille de style externe
- Feuille de style interne
- Styles en ligne

Feuille de style externe

- Méthode la plus commune
- Recommandé dans la plupart des cas
- Les différentes pages d'un site ont à peu près la même apparence
→ une seule feuille de style commune
- Un élément HTML `<link>` fait référence à ce fichier.

html

```
1 <!doctype html>
2 <html>
3   <head>
4     <meta charset="utf-8" />
5     <title>Une expérience avec CSS</title>
6     <link rel="stylesheet" href="styles.css" />
7   </head>
8   <body>
9     <h1>Hello World!</h1>
10    <p>ceci est mon premier exemple CSS</p>
11  </body>
12 </html>
```

Copy

style.css

```
1 h1 {
2   color: blue;
3   background-color: yellow;
4   border: 1px solid black;
5 }
6
7 p {
8   color: red;
9 }
```

Copy

Feuille de style interne

- Les règles CSS peuvent être écrites directement dans l'en-tête HTML <head> dans un élément <style>

```
1 <!doctype html>
2 <html>
3   <head>
4     <meta charset="utf-8" />
5     <title>Mes expérimentations CSS</title>
6     <style>
7       h1 {
8         color: blue;
9         background-color: yellow;
10        border: 1px solid black;
11      }
12
13      p {
14        color: red;
15      }
16    </style>
17  </head>
18  <body>
19    <h1>Hello World!</h1>
20    <p>Ceci est mon premier exemple CSS</p>
21  </body>
22 </html>
```

Copy

Styles en ligne

- Les styles en ligne sont des déclarations CSS qui n'affectent qu'un seul élément, elles sont déclarées grâce à l'attribut style

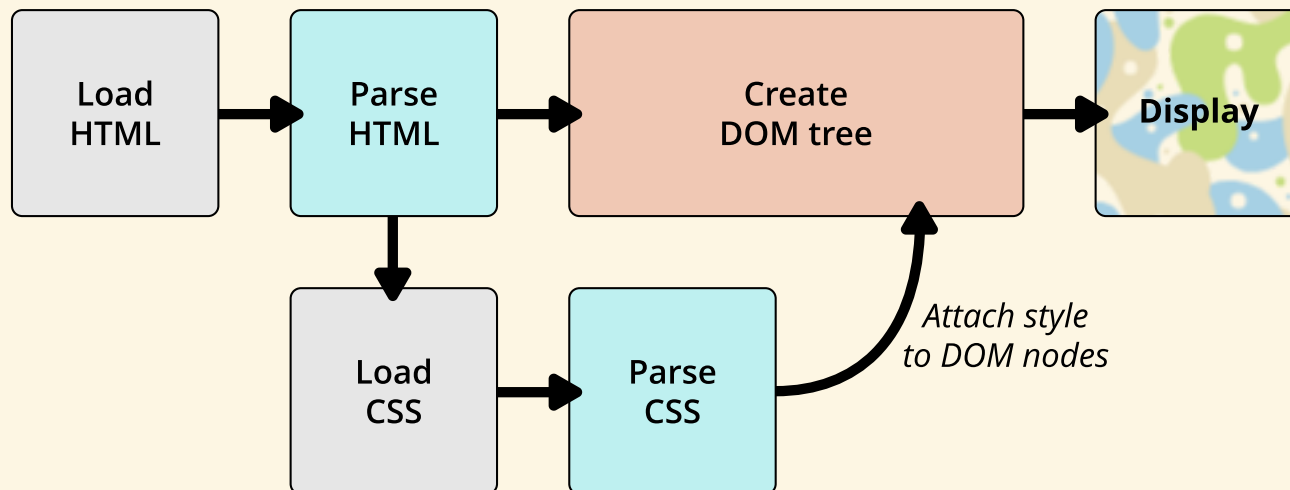
```
1 <!doctype html>
2 <html>
3   <head>
4     <meta charset="utf-8" />
5     <title>Mes expérimentations CSS</title>
6   </head>
7   <body>
8     <h1 style="color: blue;background-color: yellow;border: 1px solid black;">
9       Hello World!
10    </h1>
11    <p style="color:red;">Ceci est mon premier exemple CSS</p>
12  </body>
13 </html>
```

Copy

CSS: comment ça marche

Le navigateur:

1. **Charge le HTML** (par exemple, il le reçoit à travers le réseau).
2. **Traduit le HTML en un DOM** (Document Object Model)
→ une représentation du document HTML stockable en mémoire sur votre ordinateur.
3. **Récupère** ensuite la plupart des ressources attachées au document HTML
→ images, vidéos, feuilles CSS externes, fichiers javascript
4. **Parse le CSS**, classe les différentes règles par types de sélecteur (élément, classe, ID, etc.). et **calcule quelle règle s'applique à quel nœud du DOM**.
→ L'arbre ainsi obtenu s'appelle l'arbre de rendu (**render tree**).
5. Pour chaque nœud, **calcule l'effet visuel de la règle CSS associée**.
6. Le visuel ainsi produit est affiché à l'écran (cette étape s'appelle « **painting** »).



Le DOM

- Tout DOM a une **structure d'arbre**.
- **Chaque élément**, attribut et bout de texte du langage de balises **est traduit en un nœud du DOM** dans la structure en arbre. Les nœuds sont définis par leurs relations à d'autres nœuds du DOM. Certains éléments sont les parents de nœuds enfants (child nodes) et les nœuds enfants peuvent avoir des frères et sœurs (siblings).
- Comprendre le DOM vous aidera à concevoir, déboguer et maintenir votre CSS, car le DOM est le point de jonction entre CSS et le contenu HTML du document.
- Avec les DevTools de votre navigateur, vous pourrez naviguer à travers le DOM en sélectionnant les éléments pour les inspecter et voir quelles règles s'appliquent sur eux.

Représentation concrète du DOM

html

```
<p>
  Let's use:
  <span>Cascading</span>
  <span>Style</span>
  <span>Sheets</span>
</p>
```

Copy

DOM

```
P
├ "Let's use:"
├ SPAN
│   └ "Cascading"
├ SPAN
│   └ "Style"
└ SPAN
    └ "Sheets"
```

Copy

- Voir les DevTools → click droit → inspecter

Sélecteurs CSS

- Toute règle CSS commence par un sélecteur
- Un sélecteur est une expression qui indique au navigateur à quelle entité HTML s'applique la règle CSS correspondante.
- Il y a une large palette de sélecteurs CSS disponibles permettant avec une fine granularité de désigner les éléments auxquels appliquer des styles:
 - Sélecteurs de type, de classe et d'ID
 - Sélecteurs d'attribut
 - Pseudo-classes et pseudo-éléments
 - Combinateurs

Type de sélecteurs

- Sélecteurs de type, de classe et d'ID
→ ciblent les éléments HTML

```
/* cible des éléments HTML */  
h1 {  
}  
/* cible une classe */  
.box {  
}  
/* cible un id (unique) */  
#unique {  
}
```

Copy

- Sélecteurs d'attribut
→ ciblent des éléments en fonction de la présence d'un attribut

```
a[title] {  
}  
/* cible un attribut avec une valeur bien précise */  
a[href="https://example.com"]{  
}
```

Copy

Type de sélecteurs

- Pseudo-classes

→ ciblent des éléments dans un état donné.

```
/* pseudo-classe :hover cible un élément seulement s'il est survolé par le pointeur de la souris */  
a:hover {  
}
```

Copy

- :link, :visited
- :hover, :active, :focus
- :first-child, :last-child, :nth-child(n)

- Pseudo-éléments

→ ciblent une certaine partie d'un élément plutôt que l'élément tout entier.

```
/* ::first-line sélectionne la première ligne d'un texte contenu dans un élément */  
p::first-line {  
}
```

Copy

- ::first-line
- ::first-letter
- ::before, ::after
- ::selection

Type de sélecteurs

- Combineurs

→ combine les sélecteurs pour cibler plus finement les éléments dans nos documents.

```
/* sélectionne les paragraphes enfants directs de <article> grâce au combineur enfant (>) */  
article > p {  
}
```

Copy

- `selecteur selecteur` **Combineur descendant**
→ combine deux sélecteurs de sorte que les éléments choisis par le second sélecteur sont sélectionnés s'ils ont un élément ancêtre (parent, parent de parent, parent de parent de parent, etc...) qui correspond au premier sélecteur.
- `selecteur > selecteur` **Combineur enfants (directs)**
→ correspond uniquement aux éléments correspondant au second sélecteur qui sont les enfants directs des éléments correspondants au premier
- `selecteur + selecteur` **Combineur frère adjacents**
→ sélectionner quelque chose s'il est juste à côté d'un autre élément au même niveau de la hiérarchie.
- `selecteur ~ selecteur` **Combineur général de frères**
→ sélectionner les frères d'un élément même s'ils ne sont pas directement adjacents

Règles conflictuelles - Cascade, Spécificité et Héritage

- CSS est l'acronyme pour Cascading Style Sheets (soit « feuilles de style en cascade » en français).
- Il arrive souvent que deux règles appliquent différentes valeurs pour la même propriété sur le même élément.
- **L'héritage** décrit comment certaines propriétés CSS héritent ou non par défaut des valeurs appliquées aux éléments parents.
- **La cascade** et **la spécificité** sont des mécanismes qui contrôlent quelle règle s'applique lorsqu'un tel conflit se produit.

Héritage

- Certaines propriétés dont les valeurs sont définies sur des éléments parents sont héritées par leurs enfants
- L'héritage ne concerne pas toutes les propriétés. Ainsi, si on fixe width à 50% sur un élément, cela ne signifie pas que tous ses descendants auront une largeur égale à 50% de celle de leur parent. Si c'était le cas, CSS serait inutilement complexe.

Source

```
<!doctype html>
<html>
  <head>
    <meta charset="utf-8" />
    <title>Cascade</title>
    <style>
      body {
        color: blue;
      }

      span {
        color: black;
      }
    </style>
  </head>
  <body>
    <p>On a configuré la valeur de la propriété color
    <p>On peut changer la couleur en ciblant un éléme
  </body>
</html>
```

Copy

Résultat

On a configuré la valeur de la propriété color sur le body à blue. Cette valeur est héritée par ses descendants.

On peut changer la couleur en ciblant un élément avec un selecteur comme span.

Cascade

- Lorsque deux règles de la même couche de cascade s'appliquent et ont la même spécificité, c'est celle qui est définie dans la dernière feuille de style qui sera utilisée.

Source

```
<!doctype html>
<html>
  <head>
    <meta charset="utf-8" />
    <title>Cascade</title>
    <style>
      h1 {
        color: red;
      }
      h1 {
        color: green;
      }
    </style>
  </head>
  <body>
    <h1>Hello World!</h1>
  </body>
</html>
```

Copy

Résultat

Hello World!

Spécificité

- La spécificité est l'algorithme utilisé par le navigateur pour décider la valeur qui est appliquée à un élément pour une propriété donnée.

Source

```
<!doctype html>
<html>
  <head>
    <meta charset="utf-8" />
    <title>Cascade</title>
    <style>
      .main-heading {
        color: red;
      }
      h1 {
        color: green;
      }
    </style>
  </head>
  <body>
    <h1 class="main-heading">Hello World!</h1>
  </body>
</html>
```

Copy

Résultat

Hello World!

Spécificité - calcul

- La spécificité d'un sélecteur est mesurée selon 3 composantes différentes, qu'on peut voir comme des colonnes de centaines, dizaines et unités qui correspondent respectivement aux :
 - **Identifiant**
On marque un point dans cette colonne pour chaque sélecteur d'identifiant contenu dans le sélecteur composite.
 - **Classe**
On marque un point dans cette colonne pour chaque sélecteur de classe, d'attribut ou de pseudo-classe contenu dans le sélecteur composite.
 - **Élément**
On marque un point dans cette colonne pour chaque sélecteur d'élément ou de pseudo-élément contenu dans le sélecteur composite.

Spécificité - calcul

Sélecteur	Identifiants	Classes	Éléments	Spécificité totale
h1	0	0	1	0-0-1
h1 + p::first-letter	0	0	3	0-0-3
li > a[href*="en-US"] > .inline-warning	0	2	2	0-2-2
#identifier	1	0	0	1-0-0
button:not(#mainBtn)	1	0	1	1-0-1

[Calculateur de spécificité](#)

Spécificité - styles en ligne

- Les styles en ligne (attribut style) l'emportent sur toutes les règles déclarées dans les feuilles de style, quelle que soit leur spécificité. (spécificité 1-0-0-0)

Spécificité - exception

- Il existe une méthode permettant de passer outre toutes ces règles, y compris les styles en ligne : `!important`.
- **Il est fortement recommandé de ne pas l'utiliser sauf en cas d'extrême nécessité.**
- Le marqueur `!important` change la façon dont la cascade fonctionne et peut largement complexifier le débogage de problèmes CSS, notamment pour les grandes feuilles de style.
- **Toujours** trouver un moyen d'utiliser la spécificité avant de se résoudre à utiliser `!important`
- Utiliser `!important` **uniquement** sur des CSS spécifiques à une page pour remplacer des CSS étrangères (provenant de bibliothèques externes comme Bootstrap ou normalize.css).
- **Ne jamais** utiliser `!important` lorsque vous codez un plugin, une extension.
- **Ne jamais** utiliser `!important` sur des CSS appliquées à un site.