

JS : Manipuler le CSS

L2 MPCIE - UE Développement Web

David Lesaint
david.lesaint@univ-angers.fr



Janvier 2019

Manipuler le CSS

Cascading Style Sheets (CSS)

Langage de feuilles de style

Pour décrire la présentation d'un document écrit dans un langage de balisage (HTML, XHTML, SVG, XML).

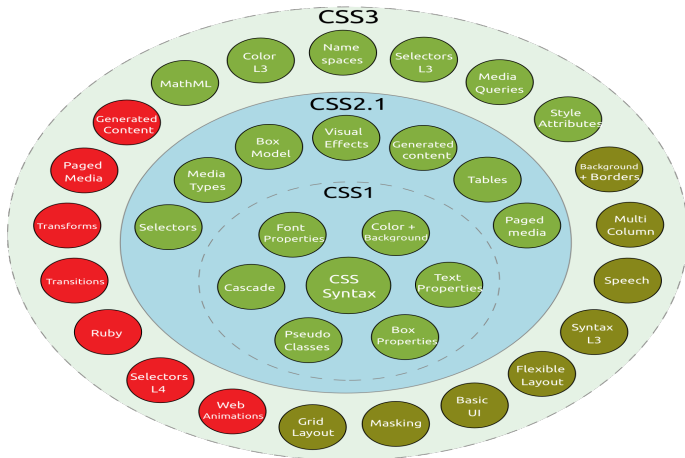
Objectifs

- Séparer la forme du contenu : mise en page, couleurs, polices ...
- Apporter plus de flexibilité et de contrôle sur la présentation.
- Permettre la réutilisation de fichiers CSS de mise en forme dans différentes pages HTML.
- Réduire la complexité des pages HTML (eg. répétition).
- Adapter la présentation aux capacités du terminal et à différents rendus (écran, impression, ...).

Les versions CSS

- CSS 1 : 1996
 - Polices, couleurs, espacements, alignements, marges ...
 - Statut : Obsolète.
- CSS 2 : 1998-2011
 - Positionnement, types média ...
 - Statut : Recommandation.
- CSS 3 : 1999-2018
 - Décomposition en ~50 modules.
 - Statut : Normalisation en cours de chaque module.
- CSS 4 : 2007-2018
 - Evolution séparée des modules CSS 3 et spécification de nouveaux modules.
 - Statut : Normalisation en cours de chaque module.

De CC 1 à CSS 3



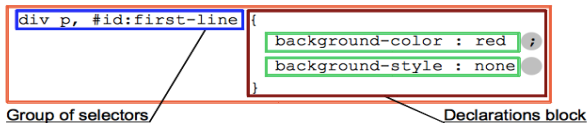
By Krauss - Own work, CC BY-SA 4.0, <https://commons.wikimedia.org/w/index.php?curid=44954967>

Déclarations, blocs et règles CSS

Déclaration = paire propriété:valeur

- Propriétés et valeurs sont insensibles à la casse.
- Toute déclaration invalide est ignorée (eg. `font-size:red`).

A CSS ruleset (or rule):



Bloc = liste entre { } de déclarations séparées par des ;

Règle = sélecteur(s) précédant un bloc de déclarations

- Le moteur de rendu CSS appliquera les déclarations du bloc aux éléments satisfaisant le groupe de sélecteurs.
- Toute règle comportant un sélecteur invalide est ignorée.

Une règle ou une @-règle (@-rule)

Différents types de @-règles :

- Méta-données relatives au fichier, eg. `@charset`, `@import`.
- Règles conditionnelles, eg. `@media`, `@document`.
- Règles descriptives, eg. `@font-face`.

Chaque @-règle a sa propre syntaxe et sémantique.

Avec

- L'attribut `style` d'un élément HTML donné.
- Des règles placées dans un élément `style` de l'en-tête du document HTML.
- Des règles placées dans un fichier `.css` importé dans le document HTML.

Editer le CSS d'éléments

feuille.css

```
1 @charset "UTF-8";
2
3 div {
4     margin: auto;
5     border: solid 1px green;
6 }
```

css-application.html

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4 <meta charset="utf-8" />
5 <title>Editer le CSS d'éléments HTML</title>
6 <link rel="stylesheet" href="feuille.css">
7 </head>
8 <style>
9 div {
10     background-color: orange;
11 }
12 </style>
13 </head>
14 <body>
15 <div style="text-align: center;">Je suis un DIV.</div>
16 </body>
17 </html>
```

Lecture de propriété CSS en JS

Accès avec `Window.getComputedStyle(element)`

Renvoie un objet contenant toutes les propriétés CSS de l'élément après application des règles (importées ou non).

- En lecture seule.

css-getcomputedstyle.html

```
1 <!DOCTYPE html>
2 <html><head><meta charset="utf-8" />
3 <title>Accès à propriété CSS avec getComputedStyle</title>
4 <style>#div1 {color: orange;}</style>
5 </head>
6 <body>
7   <div id="div1">Orange.</div>
8   <div id="div2" style="color:lightblue">Bleu.</div>
9 </script>
10 var div1 = document.getElementById('div1');
11 var div2 = document.getElementById('div2');
12 alert('Couleur du DIV1 : ' + getComputedStyle(div1).color); //rgb(255, 165, 0)
13 alert('Couleur du DIV2 : ' + getComputedStyle(div2).color); //rgb(173, 216, 230)
14 </script>
15 </body></html>
```

Ecriture de propriété CSS en JS

Avec `element.style.propriété`

- CamelCase pour propriété CSS comportant un tiret.
- Ne pas utiliser pour lire la propriété CSS calculée car n'intègre pas les règles importées !

css-style-vs-feuille.html

```
1 <!DOCTYPE html>
2 <html><head><meta charset="utf-8" />
3 <title>Accès à propriété CSS via attribut style d'éléments HTML</title>
4 <style>#div1 {background-color: orange;}</style>
5 </head>
6 <body>
7   <div id="div1">Fond orange.</div>
8   <div id="div2" style="background-color:lightblue">Fond bleu.</div>
9   <script>
10    var div1 = document.getElementById('div1');
11    var div2 = document.getElementById('div2');
12    alert('Couleur de fond du DIV1 : ' + div1.style.backgroundColor);//
13    alert('Couleur de fond du DIV2 : ' + div2.style.backgroundColor);//lightblue
14  </script>
15 </body></html>
```