

**Objectifs :**

- ⇒ Améliorer sa maîtrise du langage HTML
- ⇒ Savoir ce qu'est CSS et son utilité
- ⇒ Ecrire des pages web en HTML en utilisant CSS



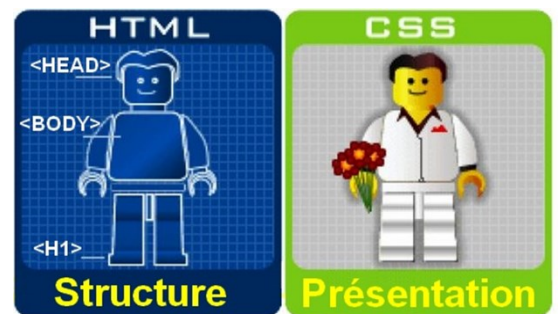
## I - Le langage CSS

### 1) Pourquoi CSS ?

Nous avons vu que HTML est un langage qui décrit à la fois le contenu (le texte brut, les images), le type de contenu (titre, paragraphe, en-tête, citation ...) et la présentation (mise en forme, couleur, ...).

La plupart des sites web utilisent un style de présentation homogène dans tout le site (on parle de [charte graphique](#)) et de ce fait les éléments de présentation seront les mêmes sur toutes les pages du site et dans l'intégralité de chaque page.

Dans ce cas, il pourrait être utile de séparer le contenu de la présentation et définir cette dernière une seule fois pour toute la page, voire pour l'ensemble des pages du site. On pourrait ainsi changer facilement le style d'un site ou d'une page en changeant simplement une fois les éléments de présentation. C'est ce que propose le langage CSS.



D'après [www.lossendiere.com](http://www.lossendiere.com)

### 2) Qu'est-ce que CSS ?

Les feuilles de style en cascade ou CSS (Cascading Style Sheets) est un langage de description de la présentation d'une page HTML. Il est donc complémentaire de HTML et ne s'occupe que de l'aspect visuel des éléments de la page (la présentation).

Lorsque l'on utilise CSS, il n'est plus nécessaire (et cela n'est d'ailleurs pas recommandé : il vaut mieux l'éviter) de spécifier des éléments de présentation (comme `<i>`, `color = ...`, etc ...) dans le code HTML car c'est CSS qui va prendre en charge l'intégralité de l'aspect visuel.

On sépare ainsi complètement le contenu de la présentation.

Le site [CSSgarden](#) propose ainsi une démonstration des possibilités de CSS. Vous pourrez y visualiser le même contenu présenté de plusieurs façons différentes. On peut en effet y changer la présentation en sélectionnant différentes feuilles de style.



**Q1 :** Aller sur le site de [CSSgarden](#) et essayer les différents designs. Quels sont les éléments qui changent et ceux qui ne changent pas ?

**Q2 :** Afficher le code source de la page (clic droit sur le fond de la page) pour deux designs différents. Quelles sont les différences dans les deux codes sources ? Cela confirme-t-il la réponse à la question précédente ?

Une démonstration de ce qui peut être réalisé via la conception basée sur CSS. Sélectionnez n'importe quelle feuille de style de la liste pour l'utiliser sur cette page.

Téléchargez les exemples [HTML](#) et [CSS](#)

**LE CHEMIN VERS L'ÉVEIL**  
Les reliques dues aux balises spécifiques aux navigateurs, aux DOMS incompatibles, au support corrompu de CSS et aux navigateurs abandonnés, jonchent un chemin sombre et lugubre.

Aujourd'hui, nous devons nous libérer de l'esprit du passé. La véritable nature du web a été révélée grâce aux efforts continus des gens du W3C, du WAP et des créateurs des principaux navigateurs.

Le Jardin Zen CSS vous invite à vous relaxer et méditer sur les importantes leçons des maîtres. Commencez à voir clairement. Apprenez à utiliser ces techniques consacrées dans une nouvelle méthode stimulante. Ne faites qu'un avec le web.

- MID CENTURY MODERN par Andrew Lohman
- GARMENTS par Dan Mill
- STEEL par Steffen Knöller
- APOTHECARY par Trent Walton
- SCREEN FILLER par Elliot Jay Stocks
- FOUNTAIN KISS par Ivarine Carlson

Le site [CSSgarden](#) permet d'essayer différentes feuilles de style en cliquant ici sur un modèle (design) dans le menu de droite.

### 3) Comment utiliser CSS ?

CSS peut être utilisé de 3 manières différentes :

1. Directement dans le code html en ajoutant un attribut `style="..."` à une balise. Dans ce cas les instructions ne sont valables que pour la balise en cours.

Ex : `<h2 style="text-decoration:underline;">Titre de niveau 2 souligné</h2>`

2. Dans l'en-tête du fichier HTML (déclaré par une balise `<style>`). Dans ce cas ces instructions sont valables pour toute la page html.

Ex : `<head>  
<style>  
h2 { background-color: blue; text-align: left; color: white; }  
</style>  
</head>`

3. Dans un fichier `.css` dont on donne la référence dans l'en-tête du fichier html. Dans ce cas les instructions sont valables pour toutes les pages qui font référence à ce fichier (souvent pour tout le site).

Ex : `<head>  
<link href="feuille.css" media="all" rel="stylesheet" type="text/css" />  
</head>`

*Le fichier « feuille.css » sera lu et on appliquera les règles CSS s'y trouvant à la page en cours.*

On peut utiliser ces 3 méthodes simultanément dans le même document et même définir plusieurs feuilles de style (fichiers `.css`) pour une même page html.

### 4) La syntaxe

La syntaxe de CSS est assez simple : il s'agit de définir des règles de représentation pour les divers objets de la page. Ainsi, on doit tout d'abord définir à qui la ou les règles vont s'appliquer (c'est le « **sélecteur** »), puis on définit chaque règle se rapportant au sélecteur en indiquant la **propriété** que l'on souhaite fixer et sa (ou ses) **valeur(s)**. La paire propriété/valeur(s) forme une **règle**.

```
p.important {  
font-weight: bold;  
border: 2px solid red;  
}
```

Comme l'indique l'image ci-contre, on fait suivre le sélecteur de l'ensemble des règles entre accolades. Le nom de la propriété est suivi d'un « : » puis de chaque valeur (séparées par des espaces s'il y en a plusieurs). Chaque règle se termine par un « ; ».

Exemple :

```
h1 {  
font-family : Comic sans MS,Verdana,Arial,Sans-serif1 ;  
text-decoration : overline underline ; /*Ligne au dessus et en dessous */  
color : #20A040 ; /* On écrit en vert */  
}
```

*On définit le style des balises h1 (titre de niveau 1) : utilisation de la police « Comic Sans MS » (ou Verdana si on ne trouve pas Comic, ou Arial, s'il n'y a pas non plus Verdana, ou n'importe quelle police sans-serif en dernier recours), des traits en dessus et en dessous et une belle couleur verte.*

On note au passage qu'on peut écrire des commentaires dans un fichier CSS comme ceux de java (débutent avec `/*` et finissent avec `*/`).

<sup>1</sup> Sans-serif est le nom d'une famille de base et sera remplacé par le navigateur par la police qu'il peut trouver qui soit *sans empattement* (sans sérif). Il existe ainsi 5 types de famille de base : Serif, Sans-serif, Monospace (pour les polices non-proportionnelles (dites à chasse fixe) comme *Courier New* sur windows), Cursive et Fantasy.

## 5) Les sélecteurs

Ils permettent de spécifier sur quels éléments vont s'appliquer les règles qui suivent et peuvent être de différents types :

- balise

Permet de viser tous les éléments d'un certain type de balise.

Exemple : `h1` concerne tous les éléments titres de niveau 1 `<h1>` de la page.

```
<style> h1 {font-size : 1.5 Em ;} </style>
<h1>Ce titre aura une police 1,5 fois plus grande</h1>
<h1>Celui-ci aussi</h1>
```

- `.maClasse`

Permet de viser tous les éléments auxquels on a attribué une certaine classe avec l'attribut `class`.

Exemple : `.important` concerne toutes les balises ayant `class="important"`

```
<style>.important {color:red;text-decoration:underline;} </style>
<p class="important">Ce paragraphe est important</p>
<p>Mais pas celui-là</p>
<h2 class="important">Ce titre sera aussi en rouge souligné</h2>
```

- `#monIdentifiant`

Permet de viser l'unique élément auquel on a attribué un certain identifiant avec l'attribut `id`. Contrairement aux classes que l'on peut attribuer à plusieurs éléments, deux éléments ne peuvent avoir le même identifiant.

Exemple : `#titrePage` concerne l'unique balise ayant `id="titrePage"`.



- `[monAttribut]`

Permet de viser tous les éléments qui possèdent un certain attribut.

Exemple : `[autoplay]` concerne tous les éléments ayant l'attribut `autoplay`

- `[monAttribut = "maValeur"]`

Exemple : `[lang="en-US"]` concerne tous les éléments `lang="en-US"`

On peut également combiner ces sélecteurs simples ou utiliser des pseudo-classes. Voir par exemple le site <http://www.yoyodesign.org/doc/w3c/css2/selector.html> pour des informations plus complètes.



**Q3** : Ecrire une feuille de style pour le fichier "exemple.html" qui contient un texte formaté (les types de contenus sont indiqués), mais non mis en forme.

On pourra utiliser [jsbin](#) ou bien créer un fichier "exemple.css" avec notepad++.

Il faut mettre en forme les titres de niveau 1 et 2, les paragraphes et la classe "code" (mais vous pouvez bien sûr aller plus loin).

## 6) Et le C dans CSS ?

Le C de CSS veut dire « Cascading », c'est-à-dire en cascade. En quoi les feuilles de styles sont-elles *en cascade* ? Tout simplement parce que les styles définis par CSS se combinent « en cascade ».

### Exemple :

#### CSS :

```
* {
  background-color : white ;
}
h1 {
  font-size : 2em ;
  text-decoration : underline ;
  color : #507050 ;
  text-align : center ;
}
p {
  text-align : left ;
  font-weight : normal ;
  text-indent : 10 px ;
}
.important {
  color : orange ;
  font-weight : bold ;
}
.crucial {
  color : red ;
  Text-decoration : blink overline underline ;
}
#info_du_jour {
  background-color : #0080f0 ;
  width : 80% ;
  margin-left : 10% ;
  padding : 10px 10px 10px 10px ;
  border : 4px dashed black ;
}
```

#### HTML :

```
<h1>Titre de la page</h1>
<p>Information banale</p>
<p class = "important">Information importante</p>
<p>Une autre info sans importance...</p>
<p class="important crucial">Cette information est extrêmement importante</p>
<h1 class="important">La page des infos importantes</h1>
<div id="info_du_jour">Ici c'est l'information de la journée que l'on veut mettre en évidence</div>
```

On obtient le résultat ci-contre :

On remarque que les règles des classes ont supplanté les règles des balises (texte écrit en orange plutôt que noir pour la classe important) et que lorsque l'on utilise plusieurs classes, celles-ci s'ajoutent, mais c'est la deuxième qui supprime la première en cas d'attribut identique (texte important et crucial en rouge).

On pourrait aussi remarquer que les règles les plus proches de la cible sont prioritaires (une règle dans un fichier CSS est moins prioritaire qu'une règle dans l'en-tête du fichier html qui est elle-même moins prioritaire qu'une règle définie dans la balise).

### Titre de la page

Information banale

Information importante

Une autre info sans importance...

Cette information est extrêmement importante

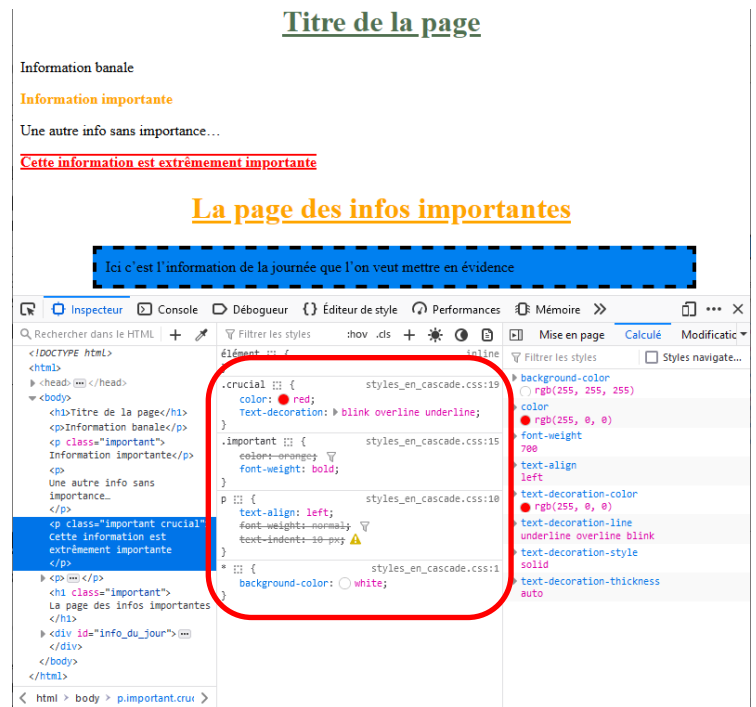
### La page des infos importantes

Ici c'est l'information de la journée que l'on veut mettre en évidence

## 7) Le retour de l'inspecteur

L'inspecteur d'objets du navigateur nous avait été utile pour visualiser le code HTML des pages web. Il nous renseigne également sur les styles appliqués par CSS. Si on examine le paragraphe crucial de l'exemple précédent avec l'inspecteur (voir capture d'écran), il nous affiche tous les styles appliqués en cascade en mettant en évidence (en les barrant) les styles qui ont été supplantés par d'autres.

Un clic sur **Éditeur de style** (dans le bandeau des outils de développement) permet ensuite d'éditer directement les styles CSS (là encore, on édite en réalité que la copie dans la mémoire de l'ordinateur, pas le fichier original sur le serveur).



## II - Au boulot !

Maintenant on en sait suffisamment pour utiliser CSS en combinaison de HTML pour écrire de belles pages bien présentées.

**Q4** : Reprendre votre site sur le mini-projet et définissez tout son aspect visuel avec CSS.

### Références :

<http://www.pierre-giraud.com/html-css/cours-complet/selecteurs-proprietes-css.php>

<https://apprendre-html.3wa.fr/css/css-classes-ids>

<https://css.developpez.com/tutoriels/premiers-pas/>

<https://openclassrooms.com/courses/apprenez-a-creer-votre-site-web-avec-html5-et-css3/mettre-en-place-le-css>

<https://openclassrooms.com/courses/apprenez-a-creer-votre-site-web-avec-html5-et-css3/memento-des-proprietes-css>

[http://isn-marchal.com/site/html\\_css.html#html](http://isn-marchal.com/site/html_css.html#html)

Sur le positionnement des éléments :

<http://cssgridgarden.com/#fr>

<http://fr.learnlayout.com/no-layout.html>

<https://openclassrooms.com/courses/apprenez-a-creer-votre-site-web-avec-html5-et-css3/le-positionnement-en-css>

