

1 Introduction

Le CSS (Cascading Style Sheets, aussi appelées feuilles de style) est un langage utilisé pour décrire l'apparence et la mise en forme d'un document écrit en HTML. En d'autres termes, alors que le HTML fournit la structure de la page web (comme les titres, les paragraphes et les liens), le CSS a pour rôle de gérer l'apparence (mise en forme et style) de la page web (taille de la police, couleur, positionnement, décoration, arrangement, espacement et même des animations, ...).

1.1 Intégration du CSS dans le document HTML

- Si le code CSS est commun à plusieurs éléments à l'intérieur d'un même document HTML, mieux vaut l'intégrer dans l'entête de la page. Pour cela, il suffit de réunir le code CSS à l'intérieur de la balise « head » avec l'attribut « style »
- S'il s'agit d'un morceau de code CSS spécifique, le plus simple est de l'intégrer directement dans la balise HTML d'un élément avec l'attribut « style ».
- Si le code CSS est commun à plusieurs pages, il est recommandé de faire un fichier à part, c'est-à-dire de créer un fichier CSS externe relié à la feuille HTML. Cela permet d'assurer la cohérence des styles entre toutes les pages d'un site.
- Si on va créer un fichier de style CSS externe relié à la feuille HTML, ce fichier aura l'extension .css
- Pour lier un fichier HTML avec un fichier CSS, on écrit dans la partie <head> :

```
<link rel="stylesheet" href="nomfichier.css">
```

1.2 Déclaration d'une règle en CSS

```
Selecteur {  
    propriete : valeur ;  
    Propriete : valeur ;  
    ....  
    Propriete : valeur ;  
}
```

- Un **sélecteur** détermine l'élément (nom de balise HTML) dont on veut modifier l'apparence (le style).
 - Une ou des **propriété(s)** CSS détermine(nt) le style qui va être appliqué à un élément
 - Une **valeur** détermine le comportement d'une propriété
- N.B. : Pour grouper plusieurs sélecteurs dans une règle, il suffit de les lister en les séparant par des virgules « , ».

1.2.1 Les commentaires en CSS

Un commentaire en CSS s'écrit entre /* et */
Exemple : /* Je suis un commentaire CSS */

2 Les sélecteurs

Sélecteur	Rôle
*	Permet de sélectionner (cibler) tous les éléments
element	Permet de sélectionner (cibler) tous les éléments de type element
#id	Permet de cibler un élément en fonction de la valeur de son attribut id
.class	Permet de cibler des éléments en fonction de la valeur de leur attributs class
element.class	Permet de cibler tous les éléments de type element en fonction de la valeur de leurs attributs class
Sélecteur :hover	Permet de cibler un élément lorsqu'il est survolé par la souris.
Sélecteur :active	Permet de cibler un élément lorsqu'il est activé.
Sélecteur :focus	Permet de cibler un élément qui a le focus

3 Styliser l'arrière plan

Propriété CSS	Rôle
background-color	Cette propriété permet d'appliquer une couleur de fond pour un élément. Comme valeur, on doit renseigner une couleur
background-image	Cette propriété permet d'appliquer une image de fond à un élément. Comme valeur, on doit renseigner l'adresse indiquant où se trouve l'image de fond. Elle peut être écrite en absolu (http :// ...) ou en relatif (paysage.jpg) si l'image de fond se trouve dans le même dossier que le fichier .css
background-size	Cette propriété permet de définir la taille de l'image de fond sur l'arrière-plan. On peut exprimer la taille de l'image en pixels ou en % Exemples : <ul style="list-style-type: none">• background-size : 200px 100px ;• background-size : 50% ;
background-repeat	Cette propriété permet de contrôler la répétition d'une image en arrière-plan. Les valeurs possibles sont repeat, repeat-x, repeat-y ou no-repeat
background	Super-propriété d'arrière-plan combine background-color, background-image, background-size et background-repeat

4 Mise en forme des textes (les polices)

4.1 Les propriétés des polices

Propriété CSS	Rôle
font-family	Cette propriété permet de définir la police du texte. Exemples de police : Arial Black, Futura, Verdana, Helvetica, Impact, Trebuchet MS, ...
font-size	Cette propriété spécifie la taille d'une police (en px ou en %)
font-weight	Cette propriété spécifie la manière dont les caractères d'un texte doivent être affichés (bold, bolder, lighter)
font-style	Cette propriété spécifie le style d'un texte (normal, italic)
font	Super-propriété, combine font-family, font-size, font-weight, font-style

4.2 Les propriétés des textes

Propriété CSS	Rôle
color	Cette propriété spécifie la couleur du texte. Elle peut accepter comme valeur : Un nom de couleur en anglais (red, yellow, green, blue, etc..) ou une notation hexadécimale (#FFFFFF, #000000, #FF0000, #00F000, #0000FF, #FFFF00, etc..)
text-align	Cette propriété spécifie l'alignement horizontal d'un texte (left, center, right, justify)
text-transform	Cette propriété spécifie la façon de transformer les caractères d'un texte en majuscules ou en minuscules (uppercase, lowercase, capitalize)
text-shadow	Cette propriété définit une ombre au texte, elle nécessite quatre valeurs dans l'ordre suivant : La projection horizontale de l'ombre (en px) La projection verticale de l'ombre (en px) Le rayon de propagation de l'ombre (le « radius », en px) La couleur de l'ombre (comme color)

5 Les listes

Propriété CSS	Rôle
list-style-type	Cette propriété définit le type de puces à utiliser pour la liste, par exemple des puces carrées (square) ou rondes (circle) pour une liste non ordonnée ou bien des nombres, des lettres ou des chiffres romains pour une liste ordonnée (lower-alpha, upper-roman, ...).
list-style-position	Cette propriété définit si les puces seront disposées à l'intérieur ou à l'extérieur du début de l'élément de liste. (outside, inside)
list-style-image	Cette propriété Spécifie une image comme marqueur d'éléments de liste
list-style	Super-propriété de liste. Combine list-style-type, list-style-position, list-style-image

Exemples :

Code CSS	Description
<pre>ul { list-style-position: inside; }</pre>	Permet de positionner les puces à l'intérieur de la zone d'affichage de l'élément de liste. <ul style="list-style-type: none">• Café• Thé• Jus
<pre>ul { list-style-position: outside; }</pre>	Permet de positionner les puces à l'extérieur de la zone d'affichage de l'élément de liste. <ul style="list-style-type: none">• Café• Thé• Jus
<pre>ul { list-style-image: url('smiley.gif'); }</pre>	Permet d'utiliser l'image 'smiley.gif' comme puce

6 Les bordures

En CSS, les bordures sont définies généralement par trois caractéristiques : un style, une épaisseur (ou largeur) et une couleur. On peut aussi créer des bordures arrondies.

- La propriété CSS « border-style » permet de définir le style d'une bordure. (solid, double, dotted, dashed, ...)
- La propriété CSS « border-width » permet de définir la largeur (ou « l'épaisseur) d'une bordure. (medium, thin, thick ou bien valeur en px)
- La propriété CSS « border-color » permet de définir la couleur d'une bordure. Elle peut accepter comme valeur : Un nom de couleur en anglais ou une notation hexadécimale.
- La propriété CSS « border-radius » permet d'arrondir les coins d'un élément. (Valeur en px ou en %)

Exemples :

Code CSS	Description
<pre>.p1 { border-style: solid; }</pre>	Permet de mettre tous les éléments HTML qui possèdent la classe p1 avec une bordure solide (trait simple)
<pre>.p1 { border-style: solid; border-width: medium; }</pre>	Permet de mettre tous les éléments HTML qui possèdent la classe p1 avec une bordure solide d'épaisseur moyenne
<pre>.p1 { border-style: solid; border-width: medium; border-color: yellow; }</pre>	Permet de mettre tous les éléments HTML qui possèdent la classe p1 avec une bordure solide d'épaisseur moyenne avec couleur jaune
<pre>.p1 { border-style: solid; border-width: medium; border-color: yellow; border-radius: 10px; }</pre>	Permet de mettre tous les éléments HTML qui possèdent la classe p1 avec une bordure solide d'épaisseur moyenne avec couleur jaune avec des coins arrondis de rayon 10 pixels

6.1 Les propriétés des bordures

Propriété CSS	Rôle
border-style	Cette propriété permet de définir le style des bordures d'un élément (solid (un trait simple), double (un double trait), dotted (un trait en pointillés), dashed (un trait en tirets) ou autre)
border-width	Cette propriété permet de définir la largeur des bordures d'un élément (medium, thin, thick ou bien valeur en px).
border-color	Cette propriété permet de définir la couleur des bordures d'un élément. Elle peut accepter comme valeur : Un nom de couleur en anglais ou une notation hexadécimale.
border-radius	Cette propriété permet de définir le rayon des coins arrondis d'un élément (valeur en px ou en %)
border	Super-propriété de bordure combine border-style, border-width, border-color et border-radius.

7 Les images : Les effets de filtres

En CSS, on peut appliquer des effets de filtre sur un élément HTML comme des images, des blocs de texte, ou tout autre élément. Ces effets peuvent inclure des modifications de couleur (niveaux de gris, sépia, saturation, ...), d'inversement des couleurs, du flou, des ombres portées, etc.

En CSS, la propriété « filter » permet d'appliquer des effets visuels (filtres) à un élément HTML, et utilise des fonctions spécifiques pour chaque type de filtre.

Exemples des effets de filtre :

Effet de filtre de flou gaussien en utilisant la syntaxe : `filter: blur()`

Code HTML	Code CSS
<pre><body> </body></pre>	<pre>.image { filter: blur(5px); }</pre>

Image initiale sans filtre

Image avec filtre de flou gaussien

Effet de filtre de niveaux gris en utilisant la syntaxe : `filter: grayscale()`

Code HTML	Code CSS
<pre><body> </body></pre>	<pre>.image { filter: grayscale(20%); }</pre>

Image initiale sans filtre

Image avec filtre de niveaux de gris

Effet de filtre d'inversion des couleurs, `filter: invert()`

Code HTML	Code CSS
<pre><body> </body></pre>	<pre>.image { filter: invert(90%); }</pre>

Image initiale sans filtre

Image avec filtre d'inversion des couleurs

7.1 Propriété des images (Les effets de filtres)

Propriété CSS	Rôle
filter	Cette propriété permet de définir des filtres sur un élément. <code>filter : blur(valeur)</code> : Effet de filtre de flou gaussien, valeur en px ou en % <code>filter : grayscale(valeur)</code> : Effet de filtre de niveaux de gris, valeur en % <code>filter : invert(valeur)</code> : Effet de filtre d'inversion des couleurs, valeur en %

8 Les tableaux

8.1 Propriétés des tableaux

Propriété CSS	Rôle
table-layout	Cette propriété permet de définir la façon de disposer les cellules, les lignes et les colonnes d'un tableau. (auto, fixed)
border-collapse	Cette propriété permet de définir si les bordures des cellules sont fusionnées ou séparées (separate, collapse)

Exemples

`border-collapse: collapse;` : les bordures des cellules sont partagées

9 Les transitions

Les transitions CSS permettent de modifier la valeur d'une propriété CSS de manière fluide et selon une durée qu'il faut définir.

Pour créer une transition, il faut définir deux valeurs pour une propriété (une première valeur de départ ou valeur par défaut et une seconde valeur d'arrivée) et faire en sorte que la valeur change progressivement de la valeur de départ à la valeur d'arrivée.

Exemples : Changer progressivement la couleur des textes des éléments HTML ou modifier la taille d'un élément, etc.

N.B. : Il n'est pas toujours strictement indispensable de définir des valeurs de départ pour les propriétés pour lesquelles on définit des transitions puisque la plupart des propriétés ont des valeurs par défaut. Par exemple pour la propriété color, si la valeur de départ n'est pas définie, la transition va donc se faire entre la valeur par défaut (noire) et la valeur d'arrivée spécifiée.

Pour créer des transitions en CSS, on doit utiliser ces différentes propriétés de type transition-* ou la propriété raccourcie transition :

— La propriété « **transition-delay** » permet d'indiquer la durée à attendre avant de débuter la transition qui s'applique pour un changement de propriété.

Exemples :

Code CSS	Description
<code>transition-delay: 250ms;</code>	Attendre 250 milliseconde avant de débuter la transition.
<code>transition-delay: 1s;</code>	Attendre 1 seconde avant de débuter la transition.

— La propriété « **transition-duration** » permet de définir le nombre de secondes ou de millisecondes que doit durer une transition. La valeur par défaut, 0s, indique qu'il n'y aura aucune animation.

Exemples :

Code CSS	Description
<code>transition-duration: 500ms;</code>	La transition dure 500 milliseconds.
<code>transition-duration: 1s;</code>	La transition dure 1 seconde.

— La propriété « **transition-property** » désigne les noms des propriétés CSS sur lesquelles un effet de transition devrait être appliqué.

Exemples :

Code CSS	Description
<code>transition-property: background-color, border;</code>	La transition sera appliquée sur les deux propriétés background-color et border
<code>transition-property: all;</code>	La transition sera appliquée sur toutes les propriétés

Exemple de transition :

Code HTML	Code CSS
<pre><body> <h1>Exemple de transitions</h1> </body></pre>	<pre>h1 { color: purple; text-align: center; margin: 20px; text-transform: uppercase; transition-property: text-shadow, font-size, color; transition-duration: 5s; } h1:hover { font-size: 12px; text-shadow: 10px 10px 8px brown; color: blue; }</pre>

Dans cet exemple, on crée une transition pour le titre h1.
On définit de nouvelles valeurs pour les propriétés font-size, text-shadow et color lors de l'état hover.
La transition est effectuée donc lorsque l'utilisateur va passer la souris sur le titre h1.
La transition va durer 5 secondes.

9.1 Propriétés des transitions

Propriété CSS	Rôle
transition-delay	Cette propriété permet de spécifier le délai nécessaire avant que la transition commence.
transition-duration	Cette propriété permet de spécifier la durée de la transition.
transition-property	Cette propriété permet de spécifier les propriétés CSS sur lesquelles l'effet de transition sera appliquée.
transition	Super-propriété de transition. Combine transition-delay, transition-duration et transition-property.

10 Les animations

Les animations CSS permettent de créer des transitions entre deux états de mise en forme.
Une animation est décrite par deux choses : des propriétés propres à l'animation d'une part et un ensemble d'étapes (keyframes) qui indiquent l'état initial, final et éventuellement des états intermédiaires d'autre part.

N.B. :

- Les transitions et les animations permettent toutes les deux de modifier la valeur de propriétés de manière progressive, au cours du temps.
- La différence entre les transitions et les animations en CSS est que les animations nous laissent à la fois une plus grande liberté et un plus grand contrôle sur le déclenchement et la progression du changement de valeur des propriétés animées.
- En effet, dans le cas d'une transition, nous ne pouvons que préciser une valeur de départ et une valeur d'arrivée pour les propriétés pour lesquelles nous souhaitons créer notre transition, tandis que dans le cas d'une animation nous allons pouvoir indiquer de manière explicite comment la « transition » entre les différentes valeurs doit se passer et pouvoir préciser différentes valeurs intermédiaires.

Pour créer une animation CSS, il faut utiliser la propriété raccourcie animation ou les propriétés détaillées correspondantes sur un ou plusieurs éléments HTML. Ces propriétés permettent de configurer la durée, le minutage et d'autres détails à propos de l'animation.

- La règle « **@keyframes** » permet de créer, nommer et définir les étapes qui composent la séquence d'une animation CSS. Cela permet de contrôler une animation plus finement que ce qu'on pourrait obtenir avec les transitions.
- La propriété « **animation-name** » définit une liste d'animations qui doivent être appliquées à l'élément HTML ciblé. Chaque nom indique une règle @keyframes qui définit les valeurs des propriétés pour la séquence.

Pour créer une animation CSS on doit :

Créer et nommer l'animation (avec @keyframes)

Attacher cette animation à un élément HTML (avec animation-name)

- La propriété « **animation-delay** » permet de définir le délai entre le moment où l'élément est chargé et le moment où l'animation commence. La valeur par défaut, 0s, (0 seconde) indique que l'animation doit démarrer dès qu'elle est appliquée.

Exemples :

Code CSS	Description
<code>animation-delay: 500ms;</code>	L'animation doit démarrer après 500 milliseconds après le chargement de l'élément HTML.
<code>animation-delay: 2s;</code>	L'animation doit démarrer après 2 secondes après le chargement de l'élément HTML.

- La propriété « **animation-duration** » permet de définir la durée d'une animation pour parcourir un cycle. La valeur par défaut est 0s, ce qui indique qu'aucune animation ne doit avoir lieu.

Exemples :

Code CSS	Description
<code>animation-duration: 750ms;</code>	La durée de l'animation est 750 milliseconds
<code>animation-duration: 2s;</code>	La durée de l'animation est 2 secondes

- La propriété « **animation-iteration-count** » indique le nombre de répétition de l'animation avant que celle-ci s'arrête. Par défaut, une animation est exécutée une seule fois

Exemples :

Code CSS	Description
<code>animation-iteration-count: 2;</code>	L'animation s'arrête après 2 répétitions
<code>animation-iteration-count: 1.5;</code>	L'animation s'arrête après 1.5 répétitions

- La propriété « **animation-direction** » indique le sens de l'animation. Par défaut une animation va dans le sens normal mais on peut aussi l'obliger d'aller vers le sens inverse ou alterné. On dispose de ces valeurs : normal, reverse, alternate, alternate-reverse.

Exemples :

Code CSS	Description
<code>animation-direction: normal;</code>	L'animation va dans le sens normal. (valeur par défaut)
<code>animation-direction: reverse;</code>	L'animation va dans le sens inverse.
<code>animation-direction: alternate;</code>	L'animation alterne d'un sens à l'autre
<code>animation-direction: alternate-reverse;</code>	L'animation alterne d'un sens à l'autre en commençant par le sens inverse.

Exemple d'animation : Glissement de textes

Code HTML	Code CSS
<pre><body> <h1>Robotique</h1> <p class="aller"> La robotique est la science... </p> <p class="retour"> La robotique trouve des applications... </p> </body></pre>	<pre>h1 { text-align: center; color: brown; } img { width: 400px; height: 500px; display: block; margin: auto; } p { animation-delay: 1s; animation-duration: 2s; animation-name: glissement; animation-iteration-count: 2; } .aller { color: blue; animation-direction: normal; } .retour { color: green; animation-direction: reverse; } @keyframes glissement { from { margin-left: 100%; width: 300%; } to { margin-left: 0%; width: 100%; } }</pre>

Exemple d'animation : Changement de couleur

Code HTML	Code CSS
<pre><body> <div class="d"></div> </body></pre>	<pre>div{ width: 90%; height: 200px; margin: 0 auto 20px auto; border: 2px solid black; } .d{ background-color: yellow; animation-name: couleur; animation-delay: 10s; animation-duration: 20s; animation-iteration-count: 2; } @keyframes couleur{ from{ background-color: yellow; } 20%{ background-color: pink; } 40%{ background-color: purple; } 60%{ background-color: blue; } 80%{ background-color: orange; } to{ background-color: yellow; } }</pre>

10.1 Propriétés des animations

Propriété CSS	Rôle
@keyframes	Cette propriété permet de spécifier les états clés de l'animation.
animation-name	Cette propriété permet de spécifier le nom de l'animation.
animation-delay	Cette propriété permet de spécifier le délai nécessaire avant que l'animation commence.
animation-duration	Cette propriété permet de spécifier la durée de l'animation
animation-iteration-count	Cette propriété permet de spécifier le nombre de répétition de l'animation
animation-direction	Cette propriété permet de définir la direction de l'animation (normal, reverse, alternate, alternate-reverse).
animation	Super-propriété de l'animation. Combine animation-name, animation-delay, animation-duration, animation-iteration-count et animation-direction