

Chap 2

Cascading Style Sheets (CSS)

CSS

- Les feuilles de style en cascade CSS (**C**ascading **S**tyle **S**heets) est un langage informatique qui sert à décrire la présentation des documents HTML, XHTML et XML.
- CSS décrit comment les éléments HTML doivent être affichés.
- L'un des objectifs majeurs des feuilles de style CSS est de séparer la structure d'un document de ses styles de présentation.

CSS

❖ **Avantages :**

- La structure du document et la présentation peuvent être gérées dans des fichiers séparés.
- Le code HTML est considérablement réduit en taille et en complexité, puisqu'il ne contient plus de balises ni d'attributs de présentation.

❖ **Avec le CSS, vous pouvez contrôler**

- La couleur, la police, la taille du texte
- L'espacement entre les éléments, la position
- Images d'arrière-plan ou couleurs
- Différents affichages pour différents appareils et tailles d'écran, et beaucoup plus!

CSS



Utilisation de CSS : Il y a **trois** façon de définir des styles CSS

- **Inline** : Dans le corps du code HTML (attribut **style** à l'intérieur des éléments HTML)
- **Interne** : Dans l'en-tête de la page (balise **<style>** dans la section **<head>**)
- **Externe** : Dans une feuille de style totalement séparée du code HTML (balise **<link>** pour créer un lien vers un fichier CSS externe)

CSS

❖ Inline

Consiste à ajouter un attribut style à n'importe quel élément HTML. On insère le code CSS directement dans cet attribut. L'attribut style peut contenir n'importe quelle propriété CSS.

- Exemple

- Code

```
<p style="background-color : BlunchedAlmond; border :1px solid black; color : Brown; font-size:150%; padding:1em;"> Ce paragraphe a du style !</p>
```

- Rendu visuel



Ce paragraphe a du style !

CSS

❖ Interne

Consiste à insérer le code CSS directement dans une balise `<style>` à l'intérieur de l'en-tête `<head>`. Une feuille de style interne peut être utilisée si la page HTML a un style unique.

- Exemple

```
<!DOCTYPE html>
<html>
  <head>
    <style>
      p
      { background-color : BlanchedAlmond;
        border:1px solid black; color : Brown;
          font-size:150%;
          padding:1em;
        }
    </style>
  </head>
  <body>
    <p>Ce paragraphe a du style !</p>
  </body>
</html>
```

— Rendu visuel

Ce paragraphe a du style !

CSS

❖ Externe

- La façon idéale de définir les CSS consiste à les enregistrer dans un document indépendant de vos pages HTML. ¹
- Grâce à cette méthode, toutes les pages qui font référence à cette feuille de style externe hériteront de toutes ses définitions.
- Les styles externes sont définis avec l'élément `<link>`, dans la balise `<head>` d'une page HTML :

CSS

❖ Externe

- Exemple

Fichier test.css:

1

```
p
{
  background-color: BlanchedAlmond;
  border: 1px solid black;
  color: yellow;
  font-size: 150%;
  padding: 1em;
}
```

Code HTML:



```
<head>
  <link href="test.css"
        rel="stylesheet">
</head>

<body>
  <p>Ce paragraphe a du style
  !</p>
</body>
```

Rendu Visuel:

Ce paragraphe a du style !

CSS

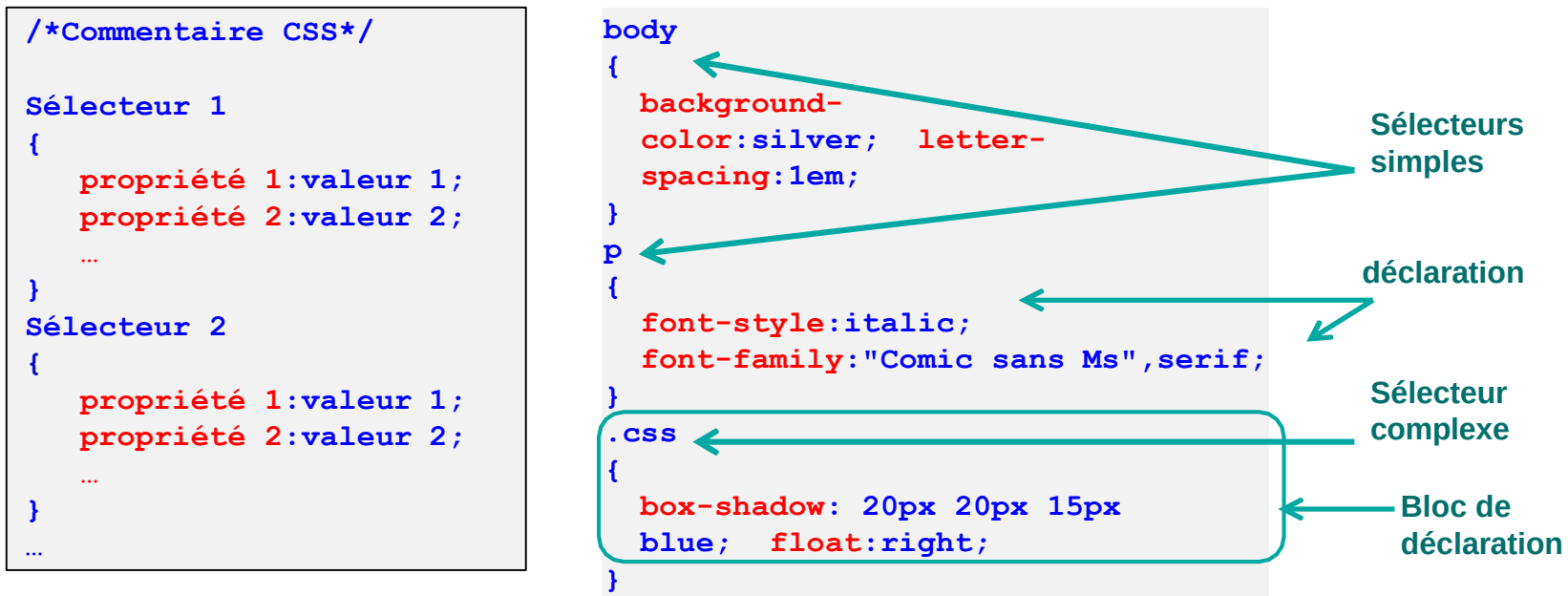
❖ **Avantage du CSS externe ou séparé**

- **Réduction de la taille des pages** : Les définitions de style ne sont faites qu'une seule fois, même si elles sont utilisées plusieurs fois.
- **Mise à jour plus facile** : On n'aura besoin que de changer la feuille de style pour mettre à jour la présentation de l'ensemble d'un site Web.
- **Réduction des temps de connexion** : Les navigateurs garderont en mémoire (en cache) le contenu de la feuille de style CSS qui s'appliquera sur toutes les pages du site. Seuls les contenus des pages devront être chargés au cours de la navigation.
- **Scinder le travail de rédaction et le travail de présentation** : On pourra commencer à rédiger le contenu des pages sans se soucier de leur présentation finale.

CSS

❖ Structure d'un fichier CSS

- Une feuille de style est un ensemble de jeux de règles de mise en forme applicables aux éléments HTML.
- Une règle CSS est composée d'un sélecteur de règle et d'un bloc de déclarations réunies à l'intérieur d'un bloc délimité par les caractères accolades { }.



CSS

❖ Sélecteurs simples

- La manière la plus simple de cibler un type d'éléments HTML en CSS est d'utiliser des sélecteurs simples appelés aussi sélecteurs éléments.
- Par exemple, le sélecteur CSS `p` permet de cibler tous les éléments `p` (c'est-à-dire tous les paragraphes) d'une page HTML pour les mettre en forme tous de la même manière.

Fichier test.html

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8" />
    <title>Page Web stylée</title>
    <link href="test.css" rel="stylesheet">
  </head>

  <body>
    <h1>Un titre de niveau 1</h1>
    <p>Un paragraphe contenant un
    <a href="http://wikipedia.org">lien</a></p>
    <p>Un deuxième paragraphe</p>
  </body>
</html>
```

Fichier test.css

```
/*Les titre h1 va s'afficher en rouge*/
h1{
  color: red;
}
/*Les paragraphes seront bleus*/
p{
  color: blue;
}
/*Le texte des liens sera vert et gras*/
a{
  color: green;
  font-weight: bold;
}
```

Un titre de niveau 1

Un paragraphe contenant un **lien**

Un deuxième paragraphe

CSS

❖ Sélecteurs universel *

- Ce sélecteur permet d'appliquer les mêmes styles à tous les éléments d'une page. Cela peut être très utile pour par exemple définir une police par défaut ou définir les marges de tous les éléments pour ensuite les positionner plus précisément.

Fichier test.html

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8" />
    <title>Page Web stylée</title>
    <link href="test.css" rel="stylesheet">
  </head>

  <body>
    <h1>Un titre de niveau 1</h1>
    <p>Un premier paragraphe</p>
    <p>Un autre paragraphe</p>
    <ul>
      <li>Elément de liste 1</li>
      <li>Elément de liste 2</li>
    </ul>
  </body>
</html>
```

Fichier test.css

```
/*tous les éléments s'afficheront en blue*/

*{
  color: blue;
}
```

Un titre de niveau 1

Un premier paragraphe

Un autre paragraphe

- Elément de liste 1
- Elément de liste 2

CSS

❖ Appliquer le même style à plusieurs éléments

- Pour appliquer un même style à plusieurs types éléments différents sans avoir à recopier le style plusieurs fois en CSS, nous allons simplement séparer nos deux sélecteurs par une virgule.
- Les styles CSS déclarés juste après s'appliqueront ainsi aux deux éléments ou groupes d'éléments sélectionnés.

Fichier test.html

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8" />
    <title>Page Web stylée</title>
    <link href="test.css" rel="stylesheet">
  </head>

  <body>
    <h1>Un titre de niveau 1</h1>
    <p>Un premier paragraphe</p>
    <p>Un autre paragraphe</p>
    <ul>
      <li>Elément de liste 1</li>
      <li>Elément de liste 2</li>
    </ul>
  </body>
</html>
```

Fichier test.css

```
/*Les éléments h1 et p afficheront en rouge*/

h1, p {
  color: blue;
}
```

Un titre de niveau 1

Un premier paragraphe

Un autre paragraphe

- Elément de liste 1
- Elément de liste 2

CSS

❖ Utiliser plusieurs sélecteurs CSS à la suite

- En mentionnant plusieurs sélecteurs à la suite en CSS, nous allons pouvoir appliquer des styles à certains éléments contenus dans d'autres éléments.
- Par exemple, utiliser le sélecteur **p a** en CSS va nous permettre d'appliquer des styles à **tous les éléments a** contenus dans des éléments **p**.

Fichier test.html

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8" />
    <title>Page Web stylée</title>
    <link href="test.css" rel="stylesheet">
  </head>
  <body>
    <h1>Un titre de niveau 1</h1>
    <p>Un premier paragraphe</p>
    <p>Un autre paragraphe avec un
      <a href="#">lien</a>
    </p>
    <ul>
      <li><a href="#">Élément de liste (lien) 1</a></li>
      <li>Élément de liste 2</li>
    </ul>
  </body>
</html>
```

Fichier test.css

```
p a
{
  background-color: yellow;
}
```

Un titre de niveau 1

Un premier paragraphe

Un autre paragraphe avec un **lien**

- [Élément de liste \(lien\) 1](#)
- Élément de liste 2

CSS

❖ Appliquer des styles aux enfants directs

- Pour cibler uniquement un élément ou un groupe d'éléments enfants directs d'un autre élément, on utilise le signe de supériorité stricte ou le caractère chevron fermant >.
- Un élément est un enfant direct ou « descendant direct » d'un autre élément s'il est directement contenu dans celui-ci.

1

Fichier test.html

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8" />
    <title>Page Web stylée</title>
    <link href="test.css" rel="stylesheet">
  </head>
  <body>
    <h1>Un titre de niveau 1</h1>
    <a href="#">Un lien</a>
    <p>Un premier paragraphe</p>
    <p>Un autre paragraphe avec un
      <a href="#">lien</a>
    </p>
    <ul>
      <li><a href="#">Elément de liste (lien) 1</a></li>
      <li>Elément de liste 2</li>
    </ul>
  </body>
</html>
```

Fichier test.css

```
body > a
{
  background-color: yellow;
}
```

Un titre de niveau 1

Un lien

Un premier paragraphe

Un autre paragraphe avec un lien

- [Elément de liste \(lien\) 1](#)
- Elément de liste 2

CSS

❖ Sélectionner l'élément suivant directement un autre élément

- Le CSS permet de cibler encore plus précisément un élément en ciblant l'élément suivant directement un autre élément grâce au caractère **+**.
- Par exemple, on va pouvoir cibler les éléments **a** (*liens*) suivant directement un élément **p**:

Fichier test.html

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8" />
    <title>Page Web stylée</title>
    <link href="test.css" rel="stylesheet">
  </head>
  <body>
    <h1>Un titre de niveau 1</h1>
    <a href="#">Un lien</a>
    <p>Un premier paragraphe</p>
    <a href="#">Un autre lien</a>
    <p>Un autre paragraphe avec un
      <a href="#">lien</a>
    </p>
    <ul>
      <li><a href="#">Élément de liste (lien) 1</a></li>
      <li>Élément de liste 2</li>
    </ul>
  </body>
</html>
```

a suit p →

a est à l'intérieur de p →

Fichier test.css

```
p + a
{
  background-color: yellow;
}
```

Un titre de niveau 1

[Un lien](#)

Un premier paragraphe

[Un autre lien](#)

Un autre paragraphe avec un [lien](#)

- [Élément de liste \(lien\) 1](#)
- Élément de liste 2

CSS

❖ Sélectionner tous les éléments suivant un autre élément

- Le CSS permet de cibler tous les éléments suivant un autre élément grâce au caractère `~`.
- Par exemple, on va pouvoir cibler tous les éléments **a** placés après un élément **p** et qui sont de même niveau

Fichier test.html

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8" />
    <title>Page Web stylée</title>
    <link href="test.css" rel="stylesheet">
  </head>
  <body>
    <h1>Un titre de niveau 1</h1>
    <a href="#">Un lien</a>
    <p>Un premier paragraphe</p>
    <a href="#">Un autre lien</a>
    <p>Un autre paragraphe avec un
      <a href="#">lien</a>
    </p>
    <ul>
      <li><a href="#">Élément de liste (lien) 1</a></li>
      <li>Élément de liste 2</li>
    </ul>
    <a href="#">Un dernier lien</a>
  </body>
</html>
```

Fichier test.css

```
p ~ a
{
  background-color: yellow;
}
```

Un titre de niveau 1

[Un lien](#)

Un premier paragraphe

[Un autre lien](#)

Un autre paragraphe avec un [lien](#)

- [Élément de liste \(lien\) 1](#)
- Élément de liste 2

[Un dernier lien](#)

CSS

❖ Sélectionner un élément avec attribut

E[attribut] : un élément E qui possède un attribut

<code>a[title]</code> { }	Sélectionne tous les liens <a> qui possèdent un attribut <code>title</code> . Exemple : <code></code>
---------------------------------	--

1

E[attribut="Valeur"] : un élément E avec un attribut et une valeur exacte

<code>a[title="Cliquez ici"]</code> { }	Sélectionne tous les liens <a> qui possèdent un attribut <code>title</code> qui doit en plus avoir exactement pour valeur « Cliquez ici ». Exemple : <code></code>
---	---

E[attribut*="Valeur"] : Un élément E avec un attribut et une valeur

<code>a[title*="ici"]</code> { }	Sélectionne tous les liens <a> qui possèdent un attribut <code>title</code> qui doit cette fois contenir dans sa valeur le mot « ici » (peu importe sa position). Exemple : <code></code>
--	---

CSS

❖ Les attributs **class** et **id**

Les attributs HTML class et id sont des attributs dits globaux car on peut les ajouter dans la balise ouvrante de n'importe quel élément HTML pour le cibler.

- Les valeurs de ces attributs ne doivent contenir ni caractères spéciaux ni espaces et commencer par une lettre.
- Chaque attribut id doit avoir une valeur propre ou unique dans une page. En revanche, plusieurs attribut class peuvent avoir les mêmes valeurs.
- Pour la feuille de style CSS, les sélecteurs seront écrits avec la syntaxe **#nom_id** pour les id et **.nom_de_classe** pour les class.

Fichier test.html

```
<!DOCTYPE html>
<html>
  <head>
    <link href="test.css" rel="stylesheet">
  </head>
  <body>
    <h1 id="orange">Un titre de niveau 1</h1>
    <p class="bleu">Un premier paragraphe</p>
    <p class="bleu">Un autre paragraphe avec un</p>
    <p id="gros">Un troisième paragraphe</p>
  </body>
</html>
```

Fichier test.css

```
#orange{
  color: orange;
}

#green{
  background: green;
}

.bleu{
  color: blue;
}
```



Un titre de niveau 1

Un premier paragraphe

Un autre paragraphe avec un

Un troisième paragraphe

CSS

❖ Affecter plusieurs attributs class à un élément HTML

- Une bonne pratique en CSS et pour la création d'un site est de ne pas surcharger un sélecteur **.class** avec de nombreux styles CSS mais au contraire d'utiliser de multiples sélecteurs .class qui se contenteront de définir chacun un comportement ou plusieurs propriétés d'un même type.
- Il suffit d'indiquer les différentes valeurs séparées par un espace.

Fichier test.html

```
<!DOCTYPE html>
<html>
  <head>
    <link href="test.css" rel="stylesheet">
  </head>
  <body>
    <h1 class="gros souligne orange">Titre 1</h1>
    <p class="bleu">Un premier paragraphe</p>
    <p>Un autre paragraphe</p>
    <h2 class="gros italique">Titre 2</h2>
  </body>
</html>
```

Fichier test.css

```
.gros{
  background: green;
}
.italique{
  font-style: italic;
}
.souligne{
  text-decoration: underline;
}
.orange{
  color: orange;
}
.bleu{
  color: blue;
}
```



Titre 1

Un premier paragraphe

Un autre paragraphe

Titre 2

CSS

❖ Les éléments HTML **div**

- On utilise l'élément **div** comme conteneur pour différents éléments afin de pouvoir ensuite facilement appliquer les mêmes styles CSS à tous les éléments contenus dans **div** par héritage ou pour les mettre en forme en appliquant un style spécifique au div.

Fichier test.html	Fichier test.css
<pre><!DOCTYPE html> <html> <head> <link href="test.css" rel="stylesheet"> </head> <body> <h1>Un titre de niveau 1</h1> <div> <p>Un premier paragraphe</p> <p>Un autre paragraphe</p> </div> <p>Un troisième paragraphe</p> <div> <p>Un dernier paragraphe</p> </div> </body> </html></pre>	<pre>body{ background-color: lightBlue; } div{ color: white; background-color: purple; width: 80%; }</pre>

Un titre de niveau 1

Un premier paragraphe

Un autre paragraphe

Un troisième paragraphe

Un dernier paragraphe

❖ Les éléments HTML **span**

- L'élément **span** sert de conteneur à un autre niveau: il sert de conteneur interne à un élément plutôt que de conteneur pour plusieurs éléments.

Fichier test.html

```
<!DOCTYPE html>
<html>
  <head>
    <link href="test.css" rel="stylesheet">
  </head>
  <body>
    <h1>Un titre de niveau 1</h1>
    <div>
      <p>Un premier paragraphe</p>
      <p>Un <span>autre</span> paragraphe</p>
    </div>
    <p>Un troisième paragraphe</p>
    <div>
      <p>Un dernier paragraphe</p>
    </div>
  </body>
</html>
```

Fichier test.css

```
div{
  color: white;
  background-color: lightBlue;
  width: 80%;
}

Span{
  color: purple;
  background-color: red;
}
```

Un titre de niveau 1

Un premier paragraphe

Un **autre** paragraphe

Un troisième paragraphe

Un dernier paragraphe

CSS

❖ Les couleurs

- En CSS, vous pouvez spécifier les couleurs de différentes manières pour styliser les éléments de vos pages web
- Les couleurs sont spécifiées à l'aide de noms de couleurs prédéfinis : **red, blue, black**

- Hexadécimal (Hex) :

```
color: #FF5733; /* Rouge orangé */  
color: #FFFFFF; /* Blanc */
```

- RGB (Red, Green, Blue) :

```
color: rgb(255, 87, 51); /* Équivalent à #FF5733 */  
color: rgb(0, 0, 0); /* Noir */
```

Les valeurs de rouge, vert et bleu sont définies sur une échelle de 0 à 255

- RGBA (avec transparence)

```
rgba(255, 99, 71, 0.6)
```

Alpha entre 0,0 et 1,0:
transparence

❖ Les couleurs

Exemple

Fichier test.html

```
<!DOCTYPE html>
<html>
  <head>
    <link href="test.css" rel="stylesheet">
  </head>
  <body>
    <h1>Un titre de niveau 1</h1>
    <div>
      <p>Un premier paragraphe</p>
      <p>Un <span>autre</span> paragraphe</p>
    </div>
    <p>Un troisième paragraphe</p>
    <div>
      <p>Un dernier paragraphe</p>
    </div>
  </body>
</html>
```

Fichier test.css

```
body{
  background-color: BurlyWood;
}
div{
  color: white;
  background-color: #FF1493 ; / DeepPink
  width: 80%;
}
Span{
  background-color: rgb(127,255,0);//Chartreuse
}
```

Un titre de niveau 1

Un premier paragraphe

Un autre paragraphe

Un troisième paragraphe

Un dernier paragraphe