

---

## N.S.I. – HTML et CSS

---

### I) Introduction

Ce cours prend place dans la partie « Interaction homme-machine » du programme de NSI de première. Il s'agit ici de l'interaction entre l'utilisateur et le serveur web hébergeant le site web proposant la page. L'interaction se fait sous le mode suivant :

- l'utilisateur envoie une requête au serveur (typiquement, l'utilisateur saisit une adresse (url) dans la barre de navigation de son navigateur afin de se connecter au serveur et à la page indiqués dans l'url (voir cours de SNT de seconde sur la construction d'une « url »)
- le serveur répond en renvoyant vers la machine cliente (le pc de l'utilisateur) la page web demandée.

Le protocole utilisé pour ces requêtes (demandes et réponses) et les échanges d'informations et de données entre le client (utilisateur) et le serveur (machine sur laquelle se trouve la page web recherchée), est le protocole « http » (*HyperText Transfer Protocole*) ou « https » (si la connexion est sécurisée ; les données échangées sont alors cryptées lors de leur transit sur le réseau local ou internet).

Dans ce chapitre, nous examinerons de quoi se compose une page web que l'on consulte, c'est-à-dire comment est écrit le fichier texte renvoyé par le serveur et lu et traduit par le navigateur afin d'afficher la page web sous sa forme désirée.

Il s'agit tout d'abord de comprendre qu'une page web consiste, physiquement, en un fichier texte, dont le contenu est écrit dans divers [langages informatiques](#), et peut-être lu par tout navigateur compatible, quel que soit le système d'exploitation de la machine.

Nous citerons ici les langages informatiques les plus couramment utilisés pour l'écriture d'une page web, en citant leur rôle spécifique :

- le langage HTML (*Hyper Text Mark up Langage*) est un langage de description, qui décrit la structure de la page web, en délimitant les zones de textes, en permettant d'insérer des images ou des liens vers d'autres pages web ou vers des fichiers ;
- le langage CSS (*Cascading Style Sheets*) est un langage de description, qui décrit le style (mise en forme) à appliquer pour les divers éléments de la page web ;
- le langage JavaScript est un langage de programmation permettant de faire tourner de petits programmes sur la machine de l'utilisateur, en interaction avec le contenu de la page web et les actions de l'utilisateur ;
- le langage PHP (*Hypertext Processor*) est un langage de programmation, exécuté celui-là côté serveur et servant à produire de nouvelles pages web, en fonction des requêtes et des informations envoyées au serveur par l'utilisateur, c'est également le PHP qui s'occupe d'interroger les éventuelles bases de données permettant, par exemple de procéder à l'identification d'un utilisateur, ou de rechercher les ressources (fichiers, données) disponibles sur le serveur pour répondre aux demandes de l'utilisateur.

Seuls les trois premiers langages seront étudiés, dans leurs aspects les plus élémentaires, cette année, le PHP étant réservé pour l'année prochaine.

Il est écrit ci-dessus que les langages HTML et CSS sont des langages de description, tandis que JavaScript est un langage de programmation, ceci signifie que les langages HTML et CSS permettent de décrire le contenu à afficher dans la page web mais ne permettent pas de manipuler ou traiter des données, ni de faire des calculs ou mettre en œuvre des algorithmes.

Nous verrons que ces deux langages utilisent un système de balises pour décrire les traitements à effectuer par le navigateur pour produire la page attendue à partir du code textuel (contenu de type texte du fichier .html).

# 1 Le langage HTML

Le langage HTML est incontournable dans l'écriture d'une page web. Les autres langages cités ci-dessus interviennent en complément, avec leurs fonctionnalités propres.

## 1.1 Éléments de base d'une page HTML

Les éléments de base d'une page web, sont des éléments de HTML, que l'on doit trouver nécessairement dans un fichier texte d'extension **.html** qui code une page web correctement selon le format défini par la norme du langage HTML 5.

```
<!DOCTYPE html>
<html lang="fr">
  <head>
    <title> ... </title>
    <meta charset="utf-8"/>
  </head>
  <body>
    ...
  </body>
</html>
```

On observe que le contenu du fichier est structuré par la présence de **balises** (*tags*) (éléments délimités par des paires de crochets « <...> »).

Dans chacune des balises, figure le **nom de la balise**, qui détermine la fonction de la balise.

**La plupart des balises forment des paires de même nom**, chaque paire délimitant une partie du fichier, (comme des paires de parenthèses délimitent une partie d'une expression en mathématiques) :

- `<html>...</html>` : délimite la partie du fichier contenant le code HTML, la première est située immédiatement après la balise `< !DOCTYPE html >` et la seconde est toujours à la fin du fichier ;
- `<head>...</ head>` : délimite la zone des en-têtes du fichier HTML, qui donnent des informations générales pour le traitement de la page web, comme l'encodage des caractères ;
- `<title>...</title>` : entre ces deux balises, le titre de la page tel qu'il s'affichera dans l'onglet du navigateur ;
- `<body>...</body>` : ces deux balises délimitent le « corps » de la page web. Entre ces deux balises se trouve le contenu de la page, qui sera affiché dans la fenêtre du navigateur.

Lorsqu'une balise fonctionne par paire, la première des balises est dite « ouvrante », la seconde « fermante ». La balise fermante porte le même nom que l'ouvrante, mais se distingue par un « / » (*slash*) précédent le nom.

Les balises ne fonctionnant pas par paire, sont dites « **vides** » ou « orphelines », par exemple, ci-dessus, la balise « `meta` » qui indique l'encodage utilisé pour les caractères du fichier .html :

```
<meta charset="utf-8" />
```

Ces balises sont à la fois ouvrantes et fermantes, ce que l'on peut indiquer en ajoutant un « / » (facultatif) avant le crochet fermant « > ».

Dans balise ouvrante d'une paire, ou une balise vide (orpheline), on peut trouver, à la suite du nom de la balise, un attribut et sa valeur, séparés par un signe « = », la valeur de l'attribut est alors indiquée entre guillemets (doubles) « " »<sup>1</sup>.

<sup>1</sup> des guillemets simples (apostrophes « ' ») peuvent aussi être utilisés.

## 1.2 Éléments du corps d'une page HTML

Nous allons voir ici comment structurer le contenu principal d'une page web, qui trouve sa place dans le fichier, entre les balises « `body` », ouvrante et fermante.

Dans ce qui suit, on ne montrera, sauf dans le premier exemple, que la partie de code html située entre `<body>` et `</body>`, le contenu à l'extérieur de ces balises, non reproduit dans les exemples, étant toujours le suivant (en rouge ci-dessous) :

```

<!DOCTYPE html>
<html lang= "fr" >
  <head>
    <title> Titre de la page dans l'onglet </title>
    <meta charset= "utf-8" />
  </head>
  <body>
    <!-- ici le contenu à afficher dans la fenêtre du navigateur -->
  </body>
</html>

```

### 1.2.a. Les différents niveaux de titres

Il existe six niveaux de titres possibles, associés, du plus gros au plus petit, aux noms de balises `h1`, `h2`, ... `h6`.

Code HTML	Rendu dans un navigateur
<pre> &lt;!DOCTYPE html&gt; &lt;html lang= "fr" &gt;   &lt;head&gt;     &lt;title&gt; Cours HTML Exemple   &lt;/title&gt;     &lt;meta charset= "utf-8" /&gt;   &lt;/head&gt;   &lt;body&gt;     &lt;h1&gt; Titre de niveau 1 &lt;/h1&gt;     &lt;h2&gt; Titre de niveau 2 &lt;/h2&gt;     &lt;h3&gt; Titre de niveau 3 &lt;/h3&gt;     &lt;h4&gt; Titre de niveau 4 &lt;/h4&gt;     &lt;h5&gt; Titre de niveau 5 &lt;/h5&gt;     &lt;h6&gt; Titre de niveau 6 &lt;/h6&gt;     Texte qui n'est pas un titre   &lt;/body&gt; &lt;/html&gt; </pre>	<p><b>Titre de niveau 1</b></p> <p><b>Titre de niveau 2</b></p> <p><b>Titre de niveau 3</b></p> <p><b>Titre de niveau 4</b></p> <p><b>Titre de niveau 5</b></p> <p><b>Titre de niveau 6</b></p> <p>Texte qui n'est pas un titre</p>
<b>Cours_HTML_exmp01.html</b>	

Le texte qui n'est pas entre une paire de balise de titre est affiché au format par défaut.

### 1.2.b. Les balises de paragraphe et de saut de ligne

La délimitation des paragraphes se fait à l'aide d'une paire de balises `<p>...</p>`.

On remarquera que les sauts de lignes dans le texte tel qu'il est saisi dans le fichier **.html** ne sont pas respectés au moment de l'affichage dans le navigateur, aussi existe-t-il une balise pour indiquer explicitement les sauts de lignes à imposer.

La balise de saut de ligne est la balise orpheline `<br>` (ou `<br/>`).

Notons que d'une façon générale, toute mise en forme dans le fichier **.html** (passages à la ligne, espaces entre les mots au sein d'une ligne, lignes vides pour espacer les paragraphes) disparaît au moment de l'affichage par le navigateur.

Seuls sont prises en compte les mises en forme indiquées par le système de balises, comme on le constate dans l'exemple ci-après.

Code HTML	Rendu dans un navigateur
<pre> &lt;body&gt;   &lt;p&gt; Les balises fonctionnant   par paires sont les plus   nombreuses en HTML.&lt;br/&gt; Les autres   sont des balises orphelines&lt;/p&gt;   &lt;p&gt; Les balises ouvrantes et les        balises orphelines peuvent   comporter des attributs.   &lt;/p&gt; &lt;/body&gt; </pre>	<p>Les balises fonctionnant par paires sont les plus nombreuses en HTML. Les autres sont des balises orphelines</p> <p>Les balises ouvrantes et les balises orphelines peuvent comporter des attributs.</p>
Cours_HTML_exmp02.html	

### 1.2.c. Élément de mise en forme (**formatage**) du texte

On dispose, pour une mise en forme rapide du texte des paires de balises suivantes :

- `<b>...</b>` : pour mettre en gras ;
- `<i>...</i>` : pour mettre en italique ;
- `<u>...</u>` : pour souligner.

Il en existe d'autres : [ici](#).

Code HTML	Rendu dans un navigateur
<pre> &lt;body&gt;   &lt;i&gt; texte en italique &lt;/i&gt;   &lt;b&gt; texte en gras &lt;/b&gt;   &lt;br/&gt;   &lt;u&gt; texte souligné &lt;/u&gt;   &lt;br/&gt;   &lt;i&gt;&lt;u&gt; mélanges de mise en forme&lt;/u&gt;&lt;/i&gt;   &lt;br/&gt;   &lt;i&gt;&lt;u&gt;&lt;b&gt; mélanges de mise en forme&lt;/b&gt;&lt;/u&gt;&lt;/i&gt;   &lt;br/&gt;   &lt;b&gt;&lt;u&gt;dernier exemple&lt;/u&gt;&lt;/b&gt; &lt;/body&gt; </pre>	<p><i>texte en italique</i> <b>texte en gras</b>  <b>texte souligné</b>  <i><u>mélanges de mise en forme</u></i>  <i><u>mélanges de mise en forme</u></i>  <b><u>dernier exemple</u></b></p>
Cours_HTML_exmp03.html	

### 1.2.d. Couples de balises ouvrante et fermante : bon parenthésage

Des balises allant par paires peuvent être écrites à la suite les unes des autres, ou être imbriquées, comme des « *poupées russes* », mais dans ce cas, il faut veiller à respecter une imbrication correcte (« bon parenthésage ») des balises, c'est-à-dire qu'une balise intérieure doit être fermée avant que la paire de balises qui la contient soit elle-même fermée<sup>2</sup>.

Ainsi, pour mettre du texte en gras et en italique, la séquence

```
<b><i> texte en gras <u>et</u> en italique</i></b>;
```

est valide,

mais la séquence suivante ne l'est pas :

```
<b><i> texte en gras <u>et</u> en italique</b></i>.
```

<sup>2</sup> On peut remarquer que les règles pour un « bon parenthésage » sont les mêmes que celles des mathématiques.

**1.2.e. Balises vides ou « orphelines »**

Les balises vides ou « orphelines », qui ne fonctionnent pas par paires sont les suivantes :

area	br	embed	img	meta	source	wbr
base	col	hr	input	param	track	

Nous n'utiliserons que celles de ces balises qui sont surlignées en jaune.

**1.2.f. Listes**

Le langage HTML propose deux types de listes, ordonnées (listes numérotées), et non ordonnées (listes « à puces »).

Les paires de balises correspondantes sont, respectivement

- `<ol>...</ol>` : listes numérotées (*ordered lists*) ;
- `<i>...</i>` : listes à puces (*unordered lists*).

Dans les deux cas, les différents éléments de la liste (items) doivent être placés dans une paire de balises `<li> ... </li>`.

Code HTML	Rendu dans un navigateur
<pre> &lt;body&gt;   &lt;ul&gt;     &lt;li&gt; Lait &lt;/li&gt;     &lt;li&gt; Oeufs &lt;/li&gt;     &lt;li&gt; Farine &lt;/li&gt;   &lt;/ul&gt;   &lt;ol&gt;     &lt;li&gt;Matin&lt;/li&gt;     &lt;li&gt;Midi&lt;/li&gt;     &lt;li&gt;Soir&lt;/li&gt;   &lt;/ol&gt; &lt;/body&gt; </pre>	<ul style="list-style-type: none"> <li>• Lait</li> <li>• Oeufs</li> <li>• Farine</li> </ul> <ol style="list-style-type: none"> <li>1. Matin</li> <li>2. Midi</li> <li>3. Soir</li> </ol>
<b>Cours_HTML_exmp04.html</b>	

**1.2.g. Tableaux**

La dernière mise en forme pour structurer un contenu textuel que nous verrons est la structure de tableaux.

L'arsenal des balises pour définir un tableau est assez conséquent car il faut délimiter le tableau (balises `<table>...</table>`), délimiter chaque ligne (balises `<tr>...</tr>` - « *table row* »), et dans chaque ligne délimiter chaque cellule (balises `<td>...</td>` - « *table datum* », donnée de table).

Code HTML	Rendu dans un navigateur								
<pre>&lt;body&gt;   &lt;table&gt;     &lt;tr&gt;       &lt;td&gt;Ingrédient&lt;/td&gt;       &lt;td&gt; Lait &lt;/td&gt;       &lt;td&gt; Oeufs &lt;/td&gt;       &lt;td&gt; Farine &lt;/td&gt;     &lt;/tr&gt;     &lt;tr&gt;       &lt;td&gt;Quantité&lt;/td&gt;       &lt;td&gt; 60 cl &lt;/td&gt;       &lt;td&gt; 3 &lt;/td&gt;       &lt;td&gt; 300 g &lt;/td&gt;     &lt;/tr&gt;   &lt;/table&gt; &lt;/body&gt;</pre>	<table><tr><td>Ingrédient</td><td>Lait</td><td>Oeufs</td><td>Farine</td></tr><tr><td>Quantité</td><td>60 cl</td><td>3</td><td>300 g</td></tr></table>	Ingrédient	Lait	Oeufs	Farine	Quantité	60 cl	3	300 g
Ingrédient	Lait	Oeufs	Farine						
Quantité	60 cl	3	300 g						
Cours HTML exmp05.html									

On remarque que l'affichage par défaut est un peu rudimentaire. Seuls les alignements sont perceptibles.

L'ajout minimal de l'attribut « border » (bordure) avec la valeur « 1px » (1 pixel) à la balise ouvrante « table » permet une meilleure lecture de la structure :

Code HTML	Rendu dans un navigateur								
<pre>&lt;body&gt;   &lt;table border="1px"&gt;     &lt;tr&gt;       &lt;td&gt;Ingrédient&lt;/td&gt;       &lt;td&gt;Lait &lt;/td&gt;       &lt;td&gt;Oeufs &lt;/td&gt;       &lt;td&gt;Farine &lt;/td&gt;     &lt;/tr&gt;     &lt;tr&gt;       &lt;td&gt;Quantité&lt;/td&gt;       &lt;td&gt;60 cl &lt;/td&gt;       &lt;td&gt;3 &lt;/td&gt;       &lt;td&gt;300 g &lt;/td&gt;     &lt;/tr&gt;   &lt;/table&gt; &lt;/body&gt;</pre>	<table><tr><td>Ingrédient</td><td>Lait</td><td>Oeufs</td><td>Farine</td></tr><tr><td>Quantité</td><td>60 cl</td><td>3</td><td>300 g</td></tr></table>	Ingrédient	Lait	Oeufs	Farine	Quantité	60 cl	3	300 g
Ingrédient	Lait	Oeufs	Farine						
Quantité	60 cl	3	300 g						
CoursHTML_exmp05-bis.html									

### 1.2.h. Insertion d'une image

L'insertion d'une image se fait à l'aide de la balise vide `<img />`, en utilisant l'attribut `src` (« source ») pour indiquer l'adresse de l'image (il s'agit de l'URL de l'image, qui pointe vers le serveur sur lequel se trouve l'image et l'emplacement de l'image sur le serveur).

- Exemple :

```
<img src ="https://www.ac-paris.fr/serail/upload/docs/image/jpeg/2017-06/cbernard07.jpg"
      alt = " entrée du lycée Claude Bernard" />
```

affiche l'une des photos stockées sur le site du lycée.



La valeur du second attribut, `alt`, donne une description de l'image, qui s'affiche, comme « alternative » à l'image, au cas où celle-ci ne s'affiche pas dans le navigateur en raison d'un problème technique (connexion lente, adresse erronée, visualisation de la page sur un appareil ne disposant pas de capacités d'affichage adaptées).

Les formats supportés pour les images sont :

- .jpg** (JPEG) format compressé avec perte ;
- .png** (Portable Network Graphics), compressé sans perte ;
- .gif**, format plus léger (seulement 256 couleurs) mais permettant des animations ;
- .svg**, format d'image vectoriel, permettant un redimensionnement sans perte de qualité.

### 1.2.i. Insertion de liens

On insère un lien vers une page web ou un fichier sur un serveur avec la balise `<a>`<sup>3</sup>. La valeur de l'attribut `href` est l'adresse de la ressource sur le serveur, et le contenu de la balise est une chaîne de texte décrivant le contenu de la ressource.

```
<a href="https://www.w3schools.com/default.asp"> Page d'accueil du site
w3schools.com</a>
```

Voir **CoursHTML\_exmp06.html** pour les images et les liens.

<sup>3</sup> Si l'on indique pas que une balise est vide (ou orpheline), il faudra comprendre qu'elle fonctionne par paire.

### 1.3 Balises neutres

Il existe deux balises dites « neutres » en ceci qu'elles n'induisent pas de base une mise en forme particulière, mais permettent de délimiter une partie de la page pour lui appliquer un traitement particulier. Il s'agit des balises :

- `div` qui permet de délimiter une section<sup>4</sup> dans le document, dont le comportement par défaut est identique à celui de la balise `p`. Il s'agit d'une balise de structure délimitant un bloc de texte ;
- `span` qui laisse le texte inchangé par défaut et n'interrompt pas le « flot » du texte, *i.e.* ne crée pas un bloc avec un passage à la ligne<sup>5</sup>.

Ces balises montreront leur utilité dans l'application de styles CSS.

Voici, un premier exemple utilisant l'exemple « historique » de texte, [Lorem ipsum](#).

<pre> &lt;body&gt;   &lt;div style="color:red;font-weight:bold;"&gt;     Lorem ipsum &lt;span style="color:blue"&gt;dolor sit amet&lt;/span&gt;,     consectetur       &lt;span style="color:blue;font-weight:normal;"&gt;         adipiscing       &lt;/span&gt;     elit.&lt;br/&gt;     Suspendisse lectus tortor, dignissim sit amet, adipiscing nec, ultricies sed,     dolor.   &lt;/div&gt;   &lt;div style="color:orange"&gt;Sed non risus.&lt;/div&gt; &lt;/body&gt; </pre>
<p><b> Lorem ipsum dolor sit amet, consectetur adipiscing elit.</b>  <b> Suspendisse lectus tortor, dignissim sit amet, adipiscing nec, ultricies sed, dolor.</b>  Sed non risus.</p>
<p align="center"><b>Cours_HTML_exmp07.html</b></p>

### 1.4 Imbrications d'éléments

Dans une page HTML, chaque balise vide, chaque partie de la page délimitée par une paire de balises de même nom, constitue un **élément** de la page.

Différents éléments peuvent être imbriqués, à condition de vérifier un bon parenthésage, comme on l'a vu pour les balises de texte (`i`, `b`, `u`). Les éléments à l'intérieur d'un élément sont appelés « sous-éléments ».

L'exemple précédent est un exemple où les deux balises `span` sont des sous-éléments de la première balise `div`.

On y observe que les attributs de la première balise `div` sont respecté avant et après les deux sous-éléments marqués par les balises `span`, et aussi appliqués au contenu de ces sous-éléments, sauf si les valeurs des attributs sont en conflit (couleur ou graisse (*weight*) de la police ici), auquel cas c'est la valeur de l'attribut spécifiée dans le sous-élément qui prime.

Voici enfin un exemple d'élément avec un sous-élément sous la forme d'une liste contenant des sous-listes :

<pre> &lt;ul&gt;   &lt;li&gt;élément 1     &lt;ul&gt;       &lt;li&gt;item 1&lt;/li&gt;       &lt;li&gt;item 2&lt;/li&gt;     &lt;/ul&gt;&lt;/li&gt;   &lt;li&gt;élément 2&lt;/li&gt; &lt;/ul&gt; </pre>	<ul style="list-style-type: none"> <li>élément 1 <ul style="list-style-type: none"> <li>item 1</li> <li>item 2</li> </ul> </li> <li>élément 2</li> </ul>
<p><b>Cours_HTML_exmp08.html</b></p>	

<sup>4</sup> La notion de section est plus générale que la notion de paragraphe. Une section est destinée à contenir plusieurs éléments de diverse nature (texte, images, ...) tandis que l'on gardera au paragraphe son sens usuel de sous-partie au sein d'un texte.

<sup>5</sup> La nuance entre `div` et `span` est très bien explicitée [ici](#).



## 1.5 Attributs de balises

Nous avons vu jusque-là quelques balises comportant des [attributs](#) avec leur valeur :

meta, table, img, a, style.

De façon générale (cf. [site w3schools](#)) :

- Tous les éléments HTML peuvent avoir des **attributs**
- Les attributs fournissent **des informations supplémentaires** sur un élément
- Les attributs sont toujours spécifiés dans **la balise de début**
- Les attributs se présentent généralement sous forme de paires nom / valeur comme: **nom = "valeur"**

Vous trouverez sur le site **w3schools.com**, la liste de toutes les [balises existantes](#), et, en allant sur la page dédiée à chaque balise, tous les attributs admissibles pour cette balise.

Vous trouverez aussi, réciproquement, la liste exhaustive de tous les [attributs reconnus](#) par le HTML, avec pour chacun les balises auxquelles ils peuvent s'appliquer.

Pour une balise donnée, les attributs peuvent être :

- obligatoires (comme `src` et `alt` pour la balise `img`)
- optionnels (comme `height` ou `width` pour la balise `img` qui permettent de préciser la taille de l'image, par sa hauteur ou sa largeur).

Lorsque plusieurs attributs sont à spécifier, on les indique, avec leurs valeurs respectives, les uns à la suite des autres, séparés par un espace :

```
<body>
   <!-- l'image s'affichera avec une hauteur de 50 pixels-->
  <br/>
   <!-- l'image s'affichera en occupant 30% de la largeur de la
fenêtre-->
</body>
```



Cours\_HTML\_exmp09.html

Parmi les attributs possibles pour une balise, on peut distinguer les [attributs dits « globaux »](#), qui peuvent s'appliquer à toutes les balises, parmi eux, on distinguera spécialement l'attribut `style`, permettant de mettre en forme un élément HTML, c'est-à-dire une partie de la page HTML délimitée par une paire de balises de même nom.

La valeur de l'attribut `style` se décompose en une liste de propriétés « CSS », sous la forme de couples « nom\_de\_la\_propriété : valeur\_de\_la\_propriété », séparées par des « ; ».

```
<h1 style="color:blue;text-align:center">This is a header</h1>
<p style="color:green">This is a paragraph.</p>
```

Ceci étant, ces propriétés de styles sont destinées à être spécifiés et appliquées à travers l'utilisation de sélecteurs CSS, ce qui fait l'objet de la partie suivante.



## 2 CSS (Cascading Style Sheets)

### 2.1 Éléments de style dans une balise

Il est possible de placer des éléments de style dans les balises du HTML (voir [ici](#)), pour spécifier, une couleur de fond, une police de caractères, une couleur de caractères, la taille du texte, ou l'alignement (droit, gauche, centré, justifié), comme dans les exemples ci-dessous.

On dit que l'on applique le style « *in line* ».

```
<body style="background-color:powderblue;">
  <h1 style="color:blue; text-align:center;">This is a heading</h1>
  <h2 style="text-align:center;">This is a heading</h2>
  <p style="font-family:courier; font-size:300%;">
    This is a paragraph.</p>
  ...
</body>
```

La syntaxe est alors de la forme :

`style=" [nom de la propriété de style] : [valeur] ; "`

à insérer dans la balise ouvrante de l'élément à styliser.

Le style s'applique alors à tous les sous-éléments au sein de la zone couverte par la balise.

Il y a des éléments de style applicables à tous éléments quels qu'ils soient, comme les propriétés ci-dessus et aussi des éléments de style spécifiques à certaines balises (listes, tableaux, ...), par exemple, pour les [listes non ordonnées](#) :

```
<ul style="list-style-type:square;">
```

On comprend bien que placer ces éléments de style dans chaque balise représente une lourdeur difficilement gérable et cela ne répond ni aux besoins ni aux standards actuels.

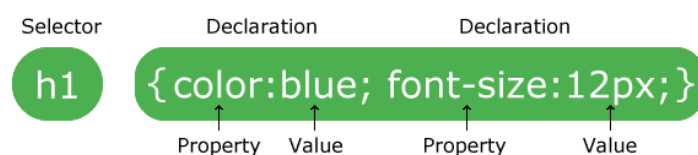
Nous allons voir ci-après plusieurs moyens et stratégies pour ajouter un style à des éléments d'une page web.

### 2.2 Éléments de style par type de balise

Il est possible de spécifier des éléments de style, soit au sein d'une page HTML, entre deux balises `<style> ... </style>` insérées dans les balises d'en-tête de fichier `<head> ... </head>`, ou dans un fichier d'extension **.css**.

La déclaration doit avoir la forme suivante, pour chaque type de balise à styliser (voir [ici](#)) :

Un ensemble de règles CSS se compose d'un sélecteur et d'un bloc de déclaration :



Le sélecteur pointe vers l'élément HTML que vous souhaitez styliser.

Le bloc de déclaration contient une ou plusieurs déclarations séparées par des points-virgules.

Chaque déclaration comprend un nom de propriété CSS et une valeur, séparés par deux points.

Une déclaration CSS se termine toujours par un point-virgule et les blocs de déclaration sont entourés d'accolades.

Si on déclare les styles directement dans la page HTML, les déclarations auront l'allure suivante :

```
...
<head>
  ...
  <style type="text/css">
    h1 {
      color: red;
      text-align: center;
    }
  </style>
</head>
```

```

    p {
      text-align: center;
    }
  </style>
</head>
...

```

Si les styles sont spécifiés dans un fichier spécifique, d'extension **.css**, la déclaration de style se fera par l'insertion d'un lien vers ce fichier, nommé sur l'exemple **monstyle.css** :

```

...
<head>
  ...
  <link rel="stylesheet" type="text/css" href="monstyle.css"/>
  ...
</head>
...

```

et le fichier **.css** peut comporter l'entête suivante :

```

@charset "utf-8";
/* CSS Document */

```

Dans cette entête :

- `@charset "utf-8"` : indique<sup>6</sup> [l'encodage des caractères](#) pour le fichier **.css** ;
- `/* CSS Document */` : est un commentaire indiquant que le fichier contient des éléments de style CSS (les commentaires en CCS sont à délimiter par « `/* ... */` »).

C'est cette dernière option, où l'on spécifie les styles dans un fichier séparé d'extension **.css**, appelé « **feuille de style CSS** » qui offre le plus de possibilités.

## 2.3 Quelques propriétés CSS

Tous les éléments de style (propriétés CSS) existants sont référencés sur la page correspondante du site **w3school.com** :

<https://www.w3schools.com/cssref/default.asp>

Nous n'en citerons que quelques-unes, qui suffiront pour comprendre les exemples et traiter les exercices.

### 2.3.a. Propriétés de style pour les « boîtes »

Les balises HTML sont des « conteneurs » de texte, d'images, de liens ou de sous-éléments, que l'on peut envisager comme des « [boîtes](#) » pour lesquelles les propriétés suivantes sont pertinentes.

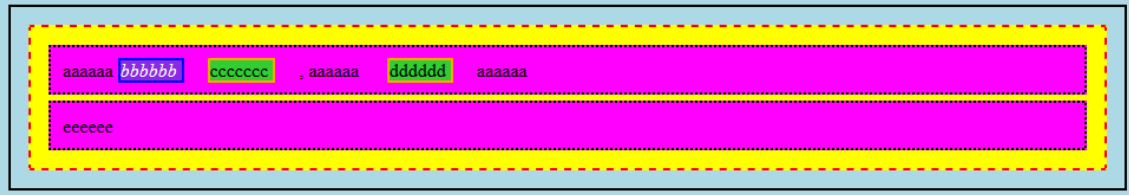
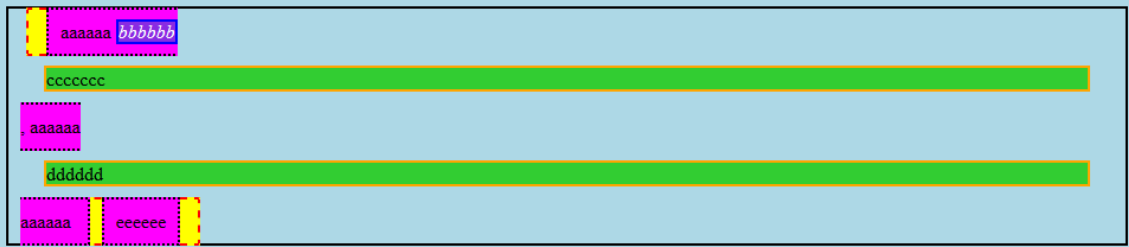
Propriété	Valeurs possibles	Description
display	none, block, inline	mode d'affichage de la boîte (si none la boîte ne s'affiche pas)
background-color	couleur	couleur de l'arrière-plan de la boîte
border	taille en pixels (px) ou en « point » (pt) motif (solid, dotted ou dashed) couleur ou none	si la valeur est none, la bordure est masquée, sinon on spécifie les valeurs de taille, de motif et la couleur de la bordure en les séparant par des espaces
margin	longueur en pixels (ou <a href="#">autre unité CSS</a> )	taille des marges autour de l'élément (marges <i>intérieures</i> )
padding	longueur en pixels (ou <a href="#">autre unité CSS</a> )	taille des ajustements ou « rembourrage » (marges <i>intérieures</i> à l'élément, autour de son contenu)

**Note** : la propriété [display](#) est à la valeur par défaut `block` pour les balises dites « de structure » (`div`, `p`, ...), et à la valeur par défaut `inline` pour les balises de texte (`span`, `img`, `a`, `i`, `b`, ...).

<sup>6</sup> Obsolète selon la [norme 2020](#).


Dans les deux exemples suivants, on utilise les propriétés CSS listées dans le tableau précédent pour visualiser les différentes « boîtes » définies par quelques balises usuelles.

Dans le second exemple, on a changé, pour chaque élément, la valeur de la propriété `display`, en la mettant à l'opposé de sa valeur par défaut.

Cours_HTML_exmp10.html	Cours_HTML_exmp10-bis.html
<pre> &lt;head&gt; ... &lt;style type="text/css"&gt;   body {     border: 2px solid black;     background-color: lightblue;     margin: 10 px;     padding: 10px;   }   div {     background-color: yellow;     border: 2px dashed red;     margin: 5px;     padding: 10px}   p {     border: 2px dotted black;     background-color: magenta;     margin: 5px;     padding: 10px}   span {     border: 2px solid orange;     background-color: limegreen;     margin: 20px;     padding: 30 px;}   i {     border: 2px solid blue;     background-color: blueviolet;     color: white;} &lt;/style&gt; &lt;/head&gt; </pre>	<pre> &lt;head&gt; ... &lt;style type="text/css"&gt;   body {     border: 2px solid black;     background-color: lightblue;     margin: 10 px;     padding: 10px;   }   div {display: inline;     background-color: yellow;     border: 2px dashed red;     margin: 5px;     padding: 10px}   p {display: inline;     border: 2px dotted black;     background-color: magenta;     margin: 5px;     padding: 10px}   span {display: block;     border: 2px solid orange;     background-color: limegreen;     margin: 20px;     padding: 30 px;}   i {display: block;     border: 2px solid blue;     background-color: blueviolet;     color: white;} &lt;/style&gt; &lt;/head&gt; </pre>
Cours_HTML_exmp10.html	
	
Cours_HTML_exmp10-bis.html	
	

Ces paramètres sont spécialement utiles pour la mise en forme des tableaux, qui s'affichent par défaut sans bordures et qui sont par construction des imbrications de boîtes, le tableau (`table`) contenant des lignes (`tr`) qui contiennent des cellules (`td`).

L'exemple ci-dessous utilise la possibilité d'appliquer un style à plusieurs types de balises à la fois (ici les balises `table` et `td`), en les séparant par des virgules.

<pre> &lt;head&gt; ... &lt;style type="text/css"&gt;   table, td {border: 1px solid black;} &lt;/style&gt; &lt;/head&gt; </pre>	
Cours_HTML_exmp11.html	

Pour les [tableaux spécialement](#), on signale en complément la propriété `border-collapse` de valeur `separate` ou `collapse` (les bordures des boîtes adjacentes sont fondues en une seule).

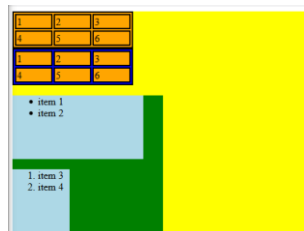
```
<head>
...
<style type="text/css">
  table {border-collapse: collapse;}
  table, td {border: 1px solid black;}
</style>
</head>
```

1	2	3
4	5	6

### Cours\_HTML\_exmp11-bis.html

Signalons enfin la possibilité de fixer la taille de la « boîte » correspondant à chaque élément, par sa hauteur (`height`) et/ou sa largeur (`width`), exprimées en pixels<sup>7</sup> (`px`) ou en pourcentage (%) de la taille correspondante de son conteneur (élément contenant directement le sous-élément dont on veut fixer la taille).

```
<!DOCTYPE html>
<html lang="fr" >
<head>
  <title>Cours CSS Exemple 12</title>
  <meta charset="utf-8" />
  <style type="text/css">
    div {width: 50%;
          height: 50%;}
    table {width: 40%;
            height: 30%;}
    table, td {
      background-color: orange;
      border: 2px solid black;}
    ul {width: 70%;
         height: 100px;}
    ol {width: 50px;
         height: 100px;}
  </style>
</head>
<body>
  <div style="background-color: yellow">
    <table style="background-color: orange">
      <tr>
        <td>1</td><td>2</td><td>3</td>
      </tr>
      <tr>
        <td>4</td><td>5</td><td>6</td>
      </tr>
    </table>
    <table style="background-color: blue">
      <tr>
        <td>1</td><td>2</td><td>3</td>
      </tr>
      <tr>
        <td>4</td><td>5</td><td>6</td>
      </tr>
    </table>
    <div style="background-color: green">
      <ul style="background-color: lightblue">
        <li>item 1</li>
        <li>item 2</li>
      </ul>
      <ol style="background-color: lightblue">
        <li>item 3</li>
        <li>item 4</li>
      </ol>
    </div>
  </div>
</body>
</html>
```



### Cours\_HTML\_exmp12.html

On voit ici les limites de la mise en page (*layout*) avec ces outils, par exemple, on constate que les différents éléments, qui correspondent tous à des balises de structure, viennent tous se positionner prennent les uns au-dessous des autres.

Pour mettre en page, un site web, on utilisera des éléments de mise en page qui dépassent largement le cadre de ce cours, et dont vous pourrez trouver la liste [ici](#).

<sup>7</sup> Sur les unités valides en CSS, on peut voir [ici](#).

### 2.3.b. Propriétés de style pour le texte

Le formatage des éléments de texte peut être fixé par des [propriétés de formatage de texte CCS](#), qui s'ajoutent aux mises en forme ponctuelles que l'on a appris à appliquer avec les balises de [formatage de texte](#) (i, b, u, ...).


De même, la police des caractères, avec ses caractéristiques (police, taille, style, grasse, ...), est prise en charge par des [propriétés de police CSS](#).

Voici une liste non exhaustive de ces propriétés et de leurs valeurs dans le tableau ci-dessous.

Propriété	Valeurs possibles	Description
text-decoration	none, underline, overline, ...	Décoration du texte
text-align	left, right, center, ou justify	Justification du texte
font-family	"Times New Roman", helvetica, ... serif, ...	Nom de la police <sup>8</sup> (ou famille générique)
font-weight	normal, light, bold ou bolder	Graisse de la police
font-style	normal ou italic	Style de la police
font-size	taille de la police en pixels (px) ou points (pt) ou xx-small, x-small, small, normal, large, x-large, xx-large	Taille de la police

Nous utiliserons ces propriétés dans les exemples des paragraphes suivants.

Notons ici qu'il existe une [propriété font](#) qui permet de fixer plusieurs paramètres de police en une seule fois :

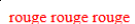
<pre>&lt;body&gt;   &lt;span style="font: bold 20px 'Courier New';"&gt;Courier New&lt;/span&gt;   &lt;span style="font: 30px arial ;"&gt;Arial&lt;/span&gt;   &lt;span style="font: italic 15px 'Times New Roman' ;"&gt;Times New Roman&lt;/span&gt; &lt;/body&gt;</pre>

<b>Cours_HTML_exmp13.html</b>

### 2.3.c. Propriétés de couleur

Les couleurs (de l'arrière-plan, du texte, des bordures), au travers des propriétés `background-color` et `color`, peuvent être [spécifiées de diverses façons en CSS](#)<sup>9</sup> :

- par leur nom (il existe [140 noms de couleurs en HTML et CCS](#)) ;
- par des niveaux de couleur RGB entre 0 et 255 (codage de chaque niveau sur 8 bits,  $2^{24} \approx 16$  millions de couleurs), ou exprimés par un triplet de valeurs en base 10 ou par une chaîne de six chiffres en hexadécimal (codage de chaque niveau par une séquence à deux chiffres en base 16).

Voici ainsi trois façons de définir la couleur primaire rouge d'intensité maximale :

<pre>&lt;body&gt;   &lt;span style="color:red"&gt;rouge&lt;/span&gt;   &lt;span style="color:#ff0000"&gt;rouge&lt;/span&gt;   &lt;span style="color:rgb(255,0,0)"&gt;rouge&lt;/span&gt; &lt;/body&gt;</pre>

<b>Cours_HTML_exmp14.html</b>

<sup>8</sup> Il est conseillé d'indiquer [plusieurs polices](#) au cas où le navigateur n'est pas en mesure de toutes les afficher.

<sup>9</sup> Il est possible de définir aussi les couleurs avec le système HSL et qu'il y a aussi possibilité d'ajouter un quatrième canal de transparence.

## 2.4 Sélecteurs CSS

Lorsque l'on applique les styles à l'aide de la balise `<style type="text/css">` située dans l'entête, head, du fichier **.html** ou dans un fichier **.css** distinct, à l'aide d'une balise

```
<link rel="stylesheet" type="text/css" href="monstyle.css"/>
```

il faut indiquer à quels éléments de la page HTML ils s'appliquent : c'est l'utilité des [sélecteurs CSS](#).

### 2.4.a. Sélection d'éléments par le nom de balise

Ce mode de sélection permet d'appliquer un style à tous les éléments définis par un même nom de balise.

On a déjà vu ce mode de sélection à l'œuvre dans les exemples du paragraphe 2.3.

En voici un autre, utilisant une feuille de style séparée.

Cours_HTML_exmp15.html	Cours_HTML_exmp15.css
<pre>&lt;!DOCTYPE html&gt; &lt;html lang= "fr" &gt;   &lt;head&gt;     &lt;title&gt;Cours CSS Exemple 15&lt;/title&gt;     &lt;meta charset= "utf-8" /&gt;     &lt;link rel="stylesheet"           type="text/css"           href="Cours_HTML_exmp15.css"/&gt;   &lt;/head&gt;   &lt;body&gt;     &lt;span&gt;Exemple&lt;/span&gt; 15 &lt;br/&gt;     On applique un style aux balises     &lt;ol&gt;       &lt;li&gt;body&lt;/li&gt;       &lt;li&gt;ol&lt;/li&gt;       &lt;li&gt;span&lt;/li&gt;     &lt;/ol&gt;   &lt;/body&gt; &lt;/html&gt;</pre>	<pre>/* CSS Exemple 15 */  body {   background-color: lightblue; }  ol, span {   color: red; }  li {   font: 20px Consolas; }</pre>
<p><b>Exemple 15</b> On applique un style aux balises</p> <ol style="list-style-type: none"> <li>1. body</li> <li>2. ol</li> <li>3. span</li> </ol>	

On constate que la propriété de couleur pour la balise `ol` est « héritée » (s'applique aussi) à ses sous-éléments, `li`.

### 2.4.b. Sélection d'éléments par la valeur de l'attribut class

L'attribut `class` permet d'étiqueter tout type de balise avec un nom de classe arbitraire, afin d'appliquer un même style à toutes les balises appartenant à la même classe.

On définit le style appliqué à une classe en précédant le nom de la classe par un point.

Cours_HTML_exmp16.html	Cours_HTML_exmp16.css
<pre>&lt;!DOCTYPE html&gt; &lt;html lang= "fr" &gt;   &lt;head&gt;     &lt;title&gt;Cours CSS Exemple 16&lt;/title&gt;     &lt;meta charset= "utf-8" /&gt;     &lt;link rel="stylesheet"           type="text/css"           href="Cours_HTML_exmp16.css"/&gt;   &lt;/head&gt;   &lt;body&gt;     &lt;span class="c1"&gt;Exemple&lt;/span&gt;16     &lt;br/&gt;     On définit ici deux classes CSS     &lt;ul&gt;       &lt;li class="c1"&gt;une classe c1&lt;/li&gt;       &lt;li class="c2"&gt;une classe         &lt;span class="c2"&gt;c2&lt;/span&gt;&lt;/li&gt;     &lt;/ul&gt;   &lt;/body&gt; &lt;/html&gt;</pre>	<pre>/* CSS Exemple 16 */  .c1 {   color: blue; }  .c2 {color: red; }  span.c1 {   font-weight: bold; }  span.c2 {   font: 30px Consolas; }  span {   text-decoration: underline }</pre>
<p><b>Exemple 16</b> On définit ici deux classes CSS</p> <ul style="list-style-type: none"> <li>• une classe <code>c1</code></li> <li>• une classe <b><code>c2</code></b></li> </ul>	

### 2.4.c. Sélection d'un élément par un attribut id

Ce même que l'on peut étiqueter toute balise par un attribut `class`, on peut étiqueter (« tagger ») toute balise par un attribut `id`.

La différence essentielle entre l'attribut `class` et `id`, est **qu'il n'est possible d'attribuer un même identifiant qu'à un seul élément d'une page .html**.

On définit le style appliqué à un élément identifié en précédant l'identifiant par un « # » (*hashtag*).

Il est possible de précéder l'identifiant par le type de la balise, mais comme un identifiant ne peut être attribué qu'à un seul élément de la page, ceci n'a d'intérêt que pour juste rappeler le type de la balise qui a été étiquetée.

Cours_HTML_exmp17.html	Cours_HTML_exmp17.css
<pre>&lt;!DOCTYPE html&gt; &lt;html lang= "fr" &gt;   &lt;head&gt;     &lt;title&gt;Cours CSS Exemple 17&lt;/title&gt;     &lt;meta charset= "utf-8" /&gt;     &lt;link rel="stylesheet" type="text/css"       href="Cours_HTML_exmp17.css"/&gt;   &lt;/head&gt;   &lt;body&gt;     &lt;span id="id1"&gt;Exemple&lt;/span&gt; 17   &lt;br/&gt;   On étiquète ici deux éléments par un   identifiant   &lt;ul id="id2"&gt;     &lt;li&gt;l'identifiant ne peut être     utilisé       que pour &lt;span&gt;un seul       élément&lt;/span&gt; de la page&lt;/li&gt;     &lt;li&gt;et ne doit pas commencer par     un chiffre&lt;/li&gt;   &lt;/ul&gt;   &lt;/body&gt; &lt;/html&gt;</pre>	<pre>/* CSS Exemple 17 */  #id1 {color: blue; }  ul#id2 {color: red; }  span {   text-decoration: underline }</pre>
<p><u>Exemple 17</u> On étiquète ici deux éléments par un identifiant</p> <ul style="list-style-type: none"> <li>• l'identifiant ne peut être utilisé que pour <u>un seul élément</u> de la page</li> <li>• et ne doit pas commencer par un chiffre</li> </ul>	

### 2.4.d. Sélecteur universel

Le symbole « \* » sélectionne tous les éléments d'une page HTML.

## 2.5 Combinaison de sélecteurs (vs regroupement de sélecteurs)

On faut distinguer la combinaison de sélecteurs du regroupement de sélecteurs.

Le regroupement de sélecteurs (déjà utilisé ci-avant) permet d'appliquer un même ensemble de propriétés de style à un ensemble d'éléments désignés par des sélecteurs séparés par des virgules.

La **combinaison** de sélecteurs, dans laquelle les sélecteurs sont séparés par des espaces, procède par pas, chaque pas consiste en la donnée d'un sélecteur désignant un ensemble d'éléments, et le sélecteur spécifié au pas suivant un sous-ensemble des éléments déjà sélectionnés.

Une combinaison `div h1` sélectionne les éléments qui sont des titres de niveau 1 qui sont à l'intérieur d'une balise `div`. C'est une règle à deux pas.

Une combinaison `div *` sélectionne les éléments qui sont à l'intérieur d'une balise `div`. C'est aussi une règle à deux pas.



La règle `div p span` comporte trois pas :

- au premier pas, le sélecteur `div` désigne tous les éléments du corps de la page qui sont déterminent des sections du document ;
- au second pas on ne considère plus que les paragraphes qui sont dans une balise `div`, et enfin, parmi ces éléments, on garde tous les sous-éléments de type `span`.

Cours_HTML_exmp18.html	
<pre>&lt;!DOCTYPE html&gt; &lt;html&gt; &lt;head&gt; &lt;style&gt; div h1 {color: red; } div * {background-color: lightblue; } div span { font-size: 20px ; } div p span { text-decoration: underline; } &lt;/style&gt; &lt;/head&gt; &lt;body&gt; &lt;h1&gt;Titre 1&lt;/h1&gt; &lt;h2&gt;Titre 2&lt;/h2&gt; Texte dans le &lt;span&gt;corps de page&lt;/span&gt; &lt;div&gt; &lt;h1&gt;Titre 1 dans div&lt;/h1&gt; texte dans un div&lt;br/&gt; &lt;span&gt;span dans div&lt;/span&gt; &lt;p&gt; Paragraphe.&lt;br/&gt; &lt;span&gt;span dans p qui est dans div&lt;/span&gt; &lt;/p&gt; &lt;/div&gt; &lt;/body&gt; &lt;/html&gt;</pre>	<p><b>Titre 1</b></p> <p><b>Titre 2</b></p> <p>Texte dans le corps de page</p> <p><b>Titre 1 dans div</b></p> <p>texte dans un div span dans div</p> <p>Paragraphe. <u>span dans p qui est dans div</u></p>
<p><b>Exemple 17</b> On étiquète ici deux éléments par un identifiant</p> <ul style="list-style-type: none"> <li>• l'identifiant ne peut être utilisé que pour <u>un seul élément de la page</u></li> <li>• et ne doit pas commencer par un chiffre</li> </ul>	

D'après la règle `div *`, tous les éléments (titres, paragraphes, parties de texte délimitées par `span`, ...) situés dans une balise à l'intérieur d'une balise `div` sont sur fond bleu.

Tous les éléments de type `span` qui sont dans une balise `div` sont avec une police de 20 pixels, tandis que seuls les éléments de type `span` qui inclus dans un paragraphe inclus lui-même dans une balise `div` voient leur texte souligné.

- Un exemple pour la mise en forme d'un tableau :

Cours_HTML_exmp19.html	Cours_HTML_exmp19.css	résultat												
<pre>&lt;table&gt; &lt;tr&gt; &lt;th id="id1"&gt;&lt;/th&gt; &lt;th&gt;valeur &lt;span&gt;1&lt;/span&gt;&lt;/th&gt; &lt;th&gt;valeur 2&lt;/th&gt; &lt;/tr&gt; &lt;tr class="impair"&gt; &lt;td class="item"&gt;item &lt;span&gt;1&lt;/span&gt;&lt;/td&gt; &lt;td class="c"&gt;1&lt;/td&gt; &lt;td&gt;2&lt;/td&gt; &lt;/tr&gt; &lt;tr class="pair"&gt; &lt;td class="item"&gt;item &lt;span&gt;2&lt;/span&gt;&lt;/td&gt; &lt;td&gt;3&lt;/td&gt; &lt;td class="c"&gt;4&lt;/td&gt; &lt;/tr&gt; &lt;tr class="impair"&gt; &lt;td class="item"&gt;item &lt;span&gt;3&lt;/span&gt;&lt;/td&gt; &lt;td&gt;5&lt;/td&gt; &lt;td class="c"&gt;6&lt;/td&gt; &lt;/tr&gt;&lt;/table&gt;</pre>	<pre>/* CSS Exemple 19 */ table, th, td { border : 2px solid black; border-collapse: collapse;} #id1 {background-color: #444444;} th {background-color: lightblue;} td {height: 40 px; width: 100px; text-align: center;}  .impair { background-color: rgb(240, 240, 240);} .pair { background-color: rgb(200, 200, 200);} .item {background-color: orange; text-align: left;}  .impair .c { font-weight:bold; color:green;} .pair .c { font-weight:bold;color: red;} .impair .item span { font-size: 40px;}</pre>	<table> <tr> <th></th><th>valeur 1</th><th>valeur 2</th></tr> <tr> <td>item 1</td><td>1</td><td>2</td></tr> <tr> <td>item 2</td><td>3</td><td>4</td></tr> <tr> <td>item 3</td><td>5</td><td>6</td></tr> </table>		valeur 1	valeur 2	item 1	1	2	item 2	3	4	item 3	5	6
	valeur 1	valeur 2												
item 1	1	2												
item 2	3	4												
item 3	5	6												

## 2.6 Cascades et règles de priorité

Avec les sélecteurs et les règles de sélection précédente, il peut arriver que deux valeurs différentes aient été spécifiées pour une même propriété d'un même élément, ces deux règles sont alors en conflit.

C'est le cas par exemple dans l'exemple, **20**, suivant, dans lequel la couleur de fond et la couleur des caractères pour les balises, `h1` et `span`, sont fixées, soit dans la balise elle-même avec l'attribut `style`, soit dans l'en-tête, avec la balise `style`, ou par appel à un fichier **.css**, avec ici deux appels à deux fichiers distincts dans des balises `link`.


### 2.6.a. Priorité aux dernières règles appliquées

Lorsqu'une même propriété CSS pour une balise est fixée par plusieurs règles distinctes, la règle retenue est, par ordre de priorité décroissante :

1. La règle donnée par l'attribut `style` dans la balise elle-même ;
2. La règle écrite dans la balise `style` ;
3. La règle écrite dans le dernier fichier **.css** appelé.

Cependant, si le fichier **.css** est appelé *après* la balise `style`, c'est la règle écrite dans le fichier **.css** qui prend le pas sur la règle de la balise `style`.

L'usage veut que les appels aux fichiers **.css** soient écrits avant la balise `style` de sorte que l'on ait l'assurance qu'elle ait la priorité sur le style dans la page<sup>10</sup>.

Cours_HTML_exmp20.html	Cours_HTML_exmp20-1.css
<pre> &lt;!DOCTYPE html&gt; &lt;html lang= "fr" &gt; &lt;head&gt;   &lt;title&gt;Cours CSS Exemple 20&lt;/title&gt;   &lt;meta charset= "utf-8" /&gt;   &lt;link rel="stylesheet" type="text/css"         href="Cours_HTML_exmp20-1.css"/&gt;   &lt;link rel="stylesheet" type="text/css"         href="Cours_HTML_exmp20-2.css"/&gt;   &lt;style&gt;     h1.c2 {color:red; background-color:yellow;}     span {color: red;}   &lt;/style&gt;   &lt;link rel="stylesheet" type="text/css"         href="Cours_HTML_exmp20-3.css"/&gt; &lt;/head&gt; &lt;body&gt;   &lt;div id="id1" class="c1"&gt;     &lt;h1 class="c2"         style="color:orange;"&gt;       Titre 1&lt;/h1&gt;     &lt;h2&gt;&lt;span id="id2"&gt;texte&lt;/span&gt;&lt;/h2&gt;   &lt;/div&gt; &lt;/body&gt; &lt;/html&gt; </pre>	Cours_HTML_exmp20-2.css
	<pre> /* CSS Exemple 20-2 */ #id1 #id2 {   background-color: lightblue;} </pre>
	Cours_HTML_exmp20-3.css
	<pre> /* CSS Exemple 20-3 */ #id2 {background-color: green;} </pre>
	Résultat
	

En vertu de ces règles, sur l'exemple :

- la propriété `color` de la balise `h1` est fixée par l'attribut `style` dans la balise `h1` ;
- la propriété `background-color` de la balise `h1` est fixée par la balise `style` dans l'en-tête ;
- la propriété `color` de la balise `span` est fixée par le fichier **Cours\_HTML\_exmp20-3.css** ;
- la propriété `background-color` de la balise `span` est fixée par le fichier **Cours\_HTML\_exmp20-2.css**.

<sup>10</sup> Plus précisément, les fichiers **.css** étant destinés à appliquer les mêmes règles à plusieurs pages d'un même site web, on veillera à placer en premier le fichier décrivant les règles générales, communes à toutes les pages du site, puis, dans l'ordre, les fichiers **.css** concernant des sous-ensembles de pages de plus en plus restreints, jusqu'à celui qui ne concerne que le style de la page actuelle, que l'on écrira dans un fichier si les règles à appliquer sont nombreuses et directement dans la balise `style` si l'ensemble de règles est restreint.

### 2.6.b. Priorité pour les règles concurrentes en un même emplacement

En addition aux règles précédentes, il y a une priorité pour les règles en concurrence, soit au sein de la même balise style, soit au sein d'un même fichier .css.

Dans ce cas, ce qui complète les règles du 2.6.a., la priorité n'est plus à la dernière règle lue, mais est attribuée selon un algorithme de calcul de la [spécificité d'une règle de style](#).

On donne une version simplifiée de cet algorithme ici.


Pour une combinaison de sélecteurs donnée, on calcule :

- le nombre  $i$  d'identifiants dans le sélecteur ;
- le nombre  $c$  de classes ;
- le nombre  $b$  de balises (*note* : le sélecteur « \* » comporte pour zéro).

La priorité est donnée au sélecteur ou à la combinaison de sélecteurs qui a, selon un ordre qui est donc lexicographique sur les valeurs du triplet  $(i, c, b)$  :

1. le plus grand  $i$  ;
2. si les  $i$  sont égaux, le plus grand  $b$  ;
3. si les  $i$  et  $b$  sont égaux, le plus grand  $c$  ;
4. si les triplets  $(i, c, b)$  sont égaux, la priorité est donnée au dernier sélecteur écrit dans l'ordre de lecture du fichier.

Voici un exemple :

Cours_HTML_exmp21.html	Cours_HTML_exmp21.css
<pre>&lt;!DOCTYPE html&gt; &lt;html lang= "fr" &gt; &lt;head&gt;   &lt;title&gt;Cours CSS Exemple 21&lt;/title&gt;   &lt;meta charset= "utf-8" /&gt;   &lt;link rel="stylesheet"         type="text/css"         href="Cours_HTML_exmp21.css"/&gt; &lt;/head&gt; &lt;body&gt;   &lt;div id="id1" class="c1"&gt;     &lt;span&gt;début&lt;/span&gt;     &lt;h1&gt;Titre 1&lt;/h1&gt;     &lt;h2 class="c2"&gt;       &lt;span id="id2"&gt;text&lt;/span&gt;     &lt;/h2&gt;   &lt;/div&gt; &lt;/body&gt; &lt;/html&gt;</pre>	<pre>/* CSS Exemple 21 */  /* règles fixant la couleur pour h1 */ div h1 {color: blue;} /* (i, c, b) = (0, 0, 2) */ h1 {color: magenta;} /* (i, c, b) = (0, 0, 1) */  /* règles fixant la couleur de fond pour h1 */ .c1 h1 {background-color: lightblue;} /* (i, c, b) = (0, 1, 1) */ .c1 * {background-color: orange;} /* (i, c, b) = (0, 1, 0) */ #id1 h1 {background-color: violet;} /* (i, c, b) = (1, 0, 1) */  /* règles fixant la couleur pour h2 */ .c1 h2 span {color: grey;} /* (i, c, b) = (0, 1, 2) */ .c1 .c2 span {color: yellow;} /* (i, c, b) = (0, 2, 1) */  /* règles fixant la couleur de fond pour h2 */ #id1 .c2 {background-color: cyan;} /* (i, c, b) = (2, 0, 0) */ .c1 #id2 {background-color: tomato;} /* (i, c, b) = (2, 0, 0) */ #id1 h2 span {background-color: aquamarine;} /* (i, c, b) = (1, 0, 2) */</pre>
Résultat	
	

En vertu des règles et des calculs de la spécificité de chaque sélecteur (*cf.* encadré), sur l'exemple :

- la propriété `color` de `h1` est fixée par le sélecteur `div h1` car  $(0, 0, 2) > (0, 0, 1)$  ;
- la propriété `background-color` de `h1` est fixée par le sélecteur `#id1 h1` car  $(1, 0, 1) > (0, 1, 1) > (0, 1, 0)$  ;
- la propriété `color` de `h2` est fixée par le sélecteur `.c1 .c2 span` car  $(0, 2, 1) > (0, 1, 2)$  ;
- la propriété `background-color` de `h2` est fixée par le sélecteur `.c1 #id2` car il est situé après le sélecteur de même valeur `#id1 .c2`  
 $(1, 1, 0) = (1, 1, 0) > (1, 0, 2)$ .