

# Les tableaux

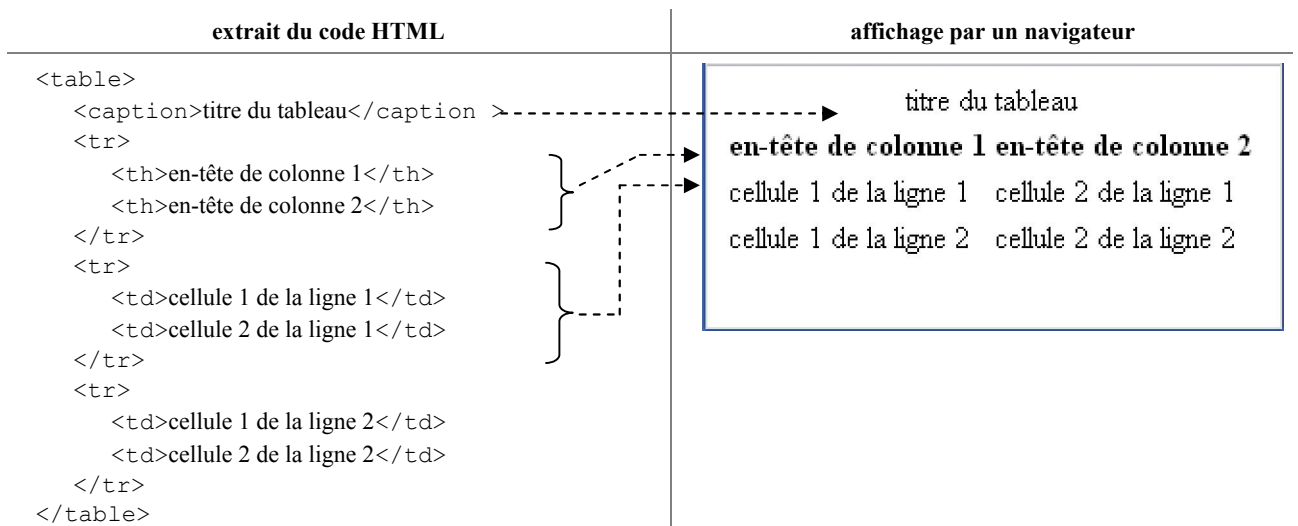
Les balises de tableau sont les suivantes :

balise	explication	attributs
<caption>	titre du tableau	
<table>	tableau	
<tr>	ligne d'un tableau ( <i>table row</i> )	
<td>	cellule d'un tableau ( <i>table data</i> )	colspan ("nb de colonnes fusionnées")
<th>	cellule en-tête ( <i>table header</i> )	rowspan ("nb de lignes fusionnées")

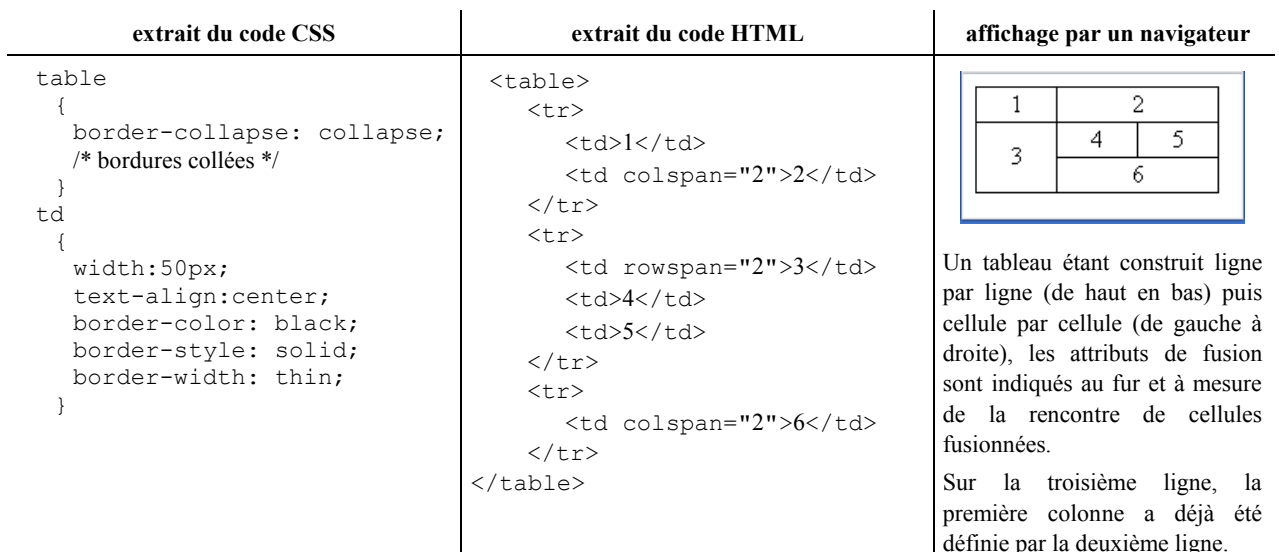
La balise <caption> n'est pas indispensable mais permet de définir une mise en forme particulière (via CSS) pour les titres de tableau. De même, la balise <th>, utilisée pour les en-têtes, applique une mise en forme particulière aux cellules concernées (par défaut, en l'absence d'une redéfinition des styles, elle est en gras).

Les tableaux sont composés :

- ligne par ligne : les balises <tr> sont obligatoirement imbriquées dans une balise <table>,
- puis cellule par cellule : les balises <td> (ou <th>) sont obligatoirement imbriquées dans des balises <tr>.



La mise en forme des tableaux (bordures, alignement horizontal et vertical, marges, ...) se fait à l'aide des styles CSS.



# Les feuilles de style CSS

Une feuille de style CSS associe des mises en forme à des balises HTML. Elle est interprétée par le navigateur (mis à jour, pour reconnaître le plus possible de propriétés CSS).

Les objectifs et les avantages des feuilles de style sont :

- un code HTML uniquement dédié au contenu (sa structure et sa sémantique),
- l'homogénéisation de la mise en forme d'un ensemble d'éléments impliquant la facilité et la rapidité de mise à jour,
- la possibilité de développer facilement plusieurs apparences d'un même contenu (pour ordinateurs, pour mobiles, ...),
- une grande richesse de possibilités.

## L'incorporation des styles

Il existe ensuite 3 modes d'incorporation de styles CSS dans un document XHTML (on privilégiera le premier) :

	dans le fichier HTML	intérêt
<b>Feuille de styles externe</b>	<pre>&lt;head&gt;   &lt;link rel="stylesheet" href="chemin et nom   fichier.css" /&gt; &lt;/head&gt;</pre>	facilite l'application d'un même style à un ensemble de documents et accélère les mises à jour
<b>Styles incorporés au document</b>	<pre>&lt;head&gt;   &lt;style&gt;     définitions de styles   &lt;/style&gt; &lt;/head&gt;</pre>	permet de visualiser les styles sans faire appel à un autre fichier ; à réserver à des styles propres à un seul fichier
<b>Style en ligne</b>	<pre>&lt;balise style="style : valeur ;"&gt;...&lt;/balise&gt;</pre>	adapté à des styles spécifiques ponctuels

Il est possible de faire appel à plusieurs styles externes (plusieurs balises `<link>`); s'ils sont en concurrence (plusieurs portants sur le même élément), le dernier style appelé s'applique.

## La définition d'un style

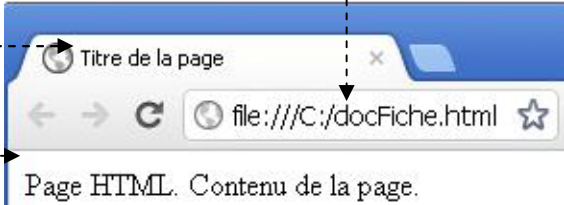
Les styles définis se rattachent aux balises :

```
balise
{
  propriété_style: valeur;
  ...
}
```

extrait du code CSS	extrait du code HTML	affichage par un navigateur
<pre>p {   color: grey;   font-family: arial; } em {   font-weight: bold /* gras */ }</pre>	<pre>&lt;p&gt;Ce mot est &lt;em&gt;important&lt;/em&gt;.&lt;/p&gt;</pre>	<p>Ce mot est <i>important</i>.</p> <p>Le mot <i>important</i> est bien en gras (et en italique par défaut dans ce navigateur)... mais également en gris : la balise <code>&lt;em&gt;</code> <b>hérite en cascade</b> des styles des balises ascendantes (ici <code>&lt;p&gt;</code>) donc de la police et de la couleur. Si une couleur différente avait été spécifiée dans le style de <code>&lt;em&gt;</code> elle aurait été appliquée à la place de la couleur héritée.</p>

# Structure de base d'une page HTML

Un code HTML, au simple format texte brut, doit être sauvegardé dans un fichier au suffixe *.htm* ou *.html* pour pouvoir être lu et interprété par un **navigateur** (l'icône associé à ce fichier sera d'ailleurs celui du navigateur paramétré par défaut).

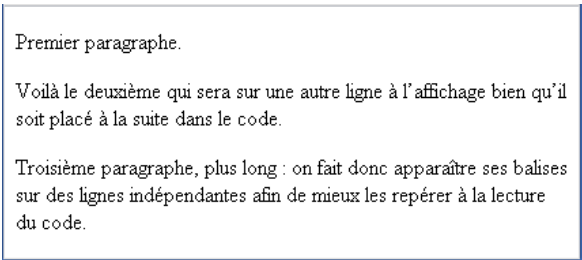
extrait du code HTML	affichage par un navigateur
<pre>&lt;!DOCTYPE html&gt; &lt;html&gt;   &lt;head&gt;     &lt;meta charset="utf-8" /&gt;     &lt;title&gt;Titre de la page&lt;/title&gt;   &lt;/head&gt;   &lt;body&gt;     Page HTML.     Contenu de la page.   &lt;/body&gt; &lt;/html&gt;</pre>	<p>chemin et nom du fichier HTML contenant le code (remarque : le fichier est ici ouvert « en local » et non transmis par le protocole du web HTTP)</p>  <p>Page HTML. Contenu de la page.</p>

Comme le montre l'exemple, un document HTML se compose :

- du **DOCTYPE** qui spécifie qu'il s'agit d'une page web HTML,
- du couple de balises `<html>...</html>` qui englobe tout le code HTML,
- de l'**en-tête** (`<head>...</head>`) qui contient des informations de caractérisation de la page :
  - l'encodage spécifie quel jeu de caractères (dont les caractères spéciaux) sera pris en compte pour l'affichage (le web a une portée mondiale avec différents alphabets), `utf-8` est préconisé pour son universalité (en vérifiant que l'éditeur de texte encode aussi en `utf-8` : sous Notepad++ menu *Encodage* ou *Paramétrage/Préférence/Nouveau document*),
  - le titre (`<title>`) s'affiche dans l'onglet ; c'est aussi lui qui apparaît dans les résultats des moteurs de recherche,
  - d'autres balises seront abordées ultérieurement,
- du **corps** (`<body>...</body>`) qui contient le contenu affiché de la page.

On remarquera que :

- les **balises** peuvent être **imbriquées** c'est-à-dire qu'un couple de balises peut être ouvert à l'intérieur d'un autre (par exemple ici `<head>` et `<body>` sont imbriqués dans `<html>`, `<title>` est imbriqué dans `<head>`) ; une balise imbriquée hérite des caractéristiques de sa balise ascendante,
- les balises doivent être refermées dans l'ordre inverse de leur ouverture pour respecter l'imbrication (`<title>` ouvert dans `<head>` doit être refermé avant `</head>`) ; une mauvaise imbrication conduit à des problèmes d'affichage,
- cette imbrication est marquée visuellement par des retraits de paragraphes ; cette habitude, appelée **indentation** facilite la lecture du code par l'humain mais n'a aucune utilité « technique » pour le navigateur,
- enfin le saut de ligne qui apparaît dans le code du corps n'est pas reproduit à l'affichage ; créons pour cela des paragraphes (couple de balises `<p>...</p>`) distincts dans l'exemple suivant.

extrait du code HTML	affichage par un navigateur
<pre>&lt;body&gt;   &lt;p&gt;Premier paragraphe.&lt;/p&gt;&lt;p&gt;Voilà le deuxième   qui sera sur une autre ligne à l'affichage bien qu'il soit   placé à la suite dans le code.&lt;/p&gt;   &lt;p&gt;     Troisième paragraphe, plus long : on fait donc     apparaître ses balises sur des lignes indépendantes     afin de mieux les repérer à la lecture du code.   &lt;/p&gt; &lt;/body&gt;</pre>	 <p>Premier paragraphe.</p> <p>Voilà le deuxième qui sera sur une autre ligne à l'affichage bien qu'il soit placé à la suite dans le code.</p> <p>Troisième paragraphe, plus long : on fait donc apparaître ses balises sur des lignes indépendantes afin de mieux les repérer à la lecture du code.</p>

# Les contenus médias

## Les images

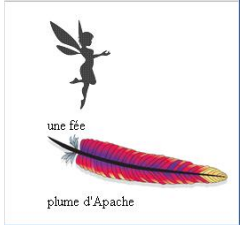
Tout d'abord, on choisira un format d'image présentant un bon rapport poids/qualité :

- JPG pour les photographies,
- PNG 8 bits pour les schémas et graphiques noir et blanc, 24 bits lorsqu'ils sont en couleur,
- GIF pour les GIF animés.

La balise `<img />` (orpheline) présente trois attributs :

- `src` afin de spécifier le chemin et le fichier de l'image (chemin relatif, comme on l'a vu pour les liens, ou URL absolue),
- `alt` pour indiquer un message si l'image ne peut être affichée (cet attribut est également utilisé en tant que facteur d'accessibilité pour les non voyants)
- `title`, facultatif, afin d'ajouter une infobulle.

Plutôt que dans une balise `<p>` on insérera une image dans une balise `<figure>` (de type *block*) en spécifiant éventuellement une légende à l'aide de `<figcaption>` :

extrait du code HTML	affichage par un navigateur
<pre>&lt;figure&gt;   &lt;img src="fee.png" /&gt;   &lt;figcaption/&gt;une fée&lt;/figcaption&gt;   &lt;img src="http://www.apache.org/images/feather-small.gif" class="image" /&gt;   &lt;figcaption/&gt;plume d'Apache&lt;/figcaption&gt; &lt;/figure&gt;</pre>	

## L'audio

Comme pour les images, il faut choisir un format et cela se complique car il n'existe pas un format reconnu par tous les navigateurs.

Le mieux est donc de proposer le fichier audio en différents formats (l'attribut `controls` affiche le lecteur par défaut du navigateur ; d'autres attributs ne sont pas présentés) :

extrait du code HTML	affichage par un navigateur
<pre>&lt;audio controls&gt;   &lt;source src="son.mp3"&gt;&lt;/source&gt;   &lt;source src="son.ogg"&gt;&lt;/source&gt; &lt;/audio&gt;</pre>	

## La vidéo

Il en est de même pour les vidéos que pour l'audio concernant les formats (le format conteneur rassemble un codec audio et un codec vidéo dont les principaux sont présentés ci-contre).

Comme pour l'audio, on proposera donc plusieurs formats.

navigateur	H.264	Ogg Theora	WebM
Firefox		✓	✓
Chrome		✓	✓
Opera	✓	✓	✓
Internet explorer	✓		✓
Safari	✓		

La syntaxe est la même avec la balise `<video controls poster="ecran.jpg">`, `poster` est un attribut permettant d'afficher une image (au lieu d'un écran noir) tant que la vidéo n'est pas diffusée. D'autres attributs ne sont pas présentés.



## Les classes de style

Dans l'exemple ci-dessus, le style défini s'appliquera à tous les paragraphes, or on peut vouloir différentes mises en forme de paragraphes... Les classes permettent de créer plusieurs styles pour une même balise.

définition dans le code CSS	appel dans le code HTML
<pre>balise.nom_classe {   propriété_style : valeur;   ... }</pre>	<pre>&lt;balise class="nom_classe"&gt;...&lt;/balise&gt;</pre>

Une classe universelle pourra être appelée par toute balise (sous réserve que les propriétés soient appropriées) puisque dans sa définition elle n'est rattachée à aucune balise :

```
.nom_classe_universelle
{
  propriété_style: valeur;
  ...
}
```

## class ou id ?

Alternativement aux classes, on peut utiliser les sélecteurs d'id (identifiant), repérés par un # dans la feuille de styles :

définition dans le code CSS	appel dans le code HTML
<pre>#nom_id {   propriété_style: valeur;   ... }</pre>	<pre>&lt;balise id="nom_id"&gt;...&lt;/balise&gt;</pre>

Cependant, dans une même page, une valeur d'id ne peut être portée que par un seul élément alors qu'une même classe peut être appelée par plusieurs balises. L'identification unique d'une balise par un identifiant sert essentiellement au JavaScript (langage de programmation exécuté par le navigateur) donc on utilisera prioritairement les classes. Une balise peut à la fois porter un id et appeler une classe (qui peuvent porter le même nom).

Il est impossible d'aborder brièvement toutes les propriétés CSS (positionnements absolus ou flottants, ...). Présentons juste deux éléments importants.

## Les unités de style

De nombreuses propriétés de style font appel à des valeurs numériques (avec un point '.' pour les décimales) :

- les unités absolues (c'est-à-dire indépendantes du média de sortie) sont *cm* (centimètre), *mm* (millimètre), *in* (inch ou pouce), *pt* (point), *pc* (pica : 12 points) avec  $1 \text{ in} = 2,54 \text{ cm} = 25,4 \text{ mm} = 72 \text{ pt} = 6 \text{ pc}$ ,
- les unités relatives (liées au média de sortie) sont *px* (pixels), *em* (1 em = hauteur de la police défaut du navigateur), *ex* (1 ex = hauteur du « x » dans cette police), % de la taille de l'élément ou de son contenant (de type *block*).

## Les marges

Un élément de type *block* possède deux types de marge :

