

HTML5



Charlotte Perriand Genech

Option NSI

Contenu

HTML et CSS : deux langages pour créer un site web	2
Les rôles de HTML et CSS.....	2
Le langage HTML	3
1. Les balises	3
2. Les attributs	4
3. Structure de base d'une page HTML5	5
4. Organiser son site.....	8
5. Les listes.....	13
6. Les liens	15
7. Les images	22
8. Les tableaux.....	24
9. Mise en forme avancée d'une page web	30

HTML et CSS : deux langages pour créer un site web

Pour créer un site web, on doit donner des instructions à l'ordinateur. Il ne suffit pas simplement de taper le texte qui devra figurer dans le site (comme on le ferait dans un traitement de texte Word, par exemple), il faut aussi indiquer où placer ce texte, insérer des images, faire des liens entre les pages, etc.

Les rôles de HTML et CSS

Pour expliquer à l'ordinateur ce que vous voulez faire, il va falloir utiliser un langage qu'il comprend. Et c'est là que les choses se corsent, parce qu'*il va falloir apprendre deux langages* !

HTML (*HyperText Markup Language*) : il a fait son apparition dès 1991 lors du lancement du Web. Son rôle est de gérer et organiser le contenu. C'est donc en HTML que vous écrirez ce qui doit être affiché sur la page : du texte, des liens, des images... Vous direz par exemple : « Ceci est mon titre, ceci est mon menu, voici le texte principal de la page, voici une image à afficher, etc. ».

CSS (*Cascading Style Sheets*, aussi appelées *Feuilles de style*) : le rôle du CSS est de gérer l'apparence de la page web (agencement, positionnement, décoration, couleurs, taille du texte...). Ce langage est venu compléter le HTML en 1996.

Vous pouvez très bien créer un site web uniquement en HTML, mais celui-ci ne sera pas très beau : l'information apparaîtra « brute ». C'est pour cela que le langage CSS vient toujours le compléter.

HTML 5 est LA dernière version. Encore assez peu répandue, elle fait beaucoup parler d'elle car elle apporte de nombreuses améliorations comme la possibilité d'inclure facilement des vidéos, un meilleur agencement du contenu, de nouvelles fonctionnalités pour les formulaires, etc.

C'est cette version que nous allons découvrir ensemble.

Le langage HTML

1. Les balises

Les pages HTML sont remplies de ce qu'on appelle des balises. Celles-ci sont invisibles à l'écran pour vos visiteurs, mais elles permettent à l'ordinateur de comprendre ce qu'il doit afficher.

Les balises se repèrent facilement. Elles sont entourées de « chevrons », c'est-à-dire des symboles < et >, comme ceci : <balise>

À quoi est-ce qu'elles servent ? Elles indiquent la nature du texte qu'elles encadrent. Elles veulent dire par exemple : « Ceci est le titre de la page », « Ceci est une image », « Ceci est un paragraphe de texte », etc.

On distingue deux types de balises : les balises en paires et les balises orphelines.

- **Les balises en paires**

Elles s'ouvrent, contiennent du texte, et se ferment plus loin. Voici à quoi elles ressemblent :

<titre> Ceci est un titre </titre>

On distingue une balise ouvrante (<titre>) et une balise fermante (</titre>) qui indique que le titre se termine. Cela signifie pour l'ordinateur que tout ce qui n'est *pas* entre ces deux balises... n'est pas un titre.

Ceci n'est pas un titre <titre> Ceci est un titre </titre> Ceci n'est pas un titre

- **Les balises orphelines**

Ce sont des balises qui servent le plus souvent à insérer un élément à un endroit précis (par exemple une image). Il n'est pas nécessaire de délimiter le début et la fin de l'image, on veut juste dire à l'ordinateur « Insère une image ici ».

Une balise orpheline s'écrit comme ceci :

<image/>

Notez que le / de fin n'est pas obligatoire. On pourrait écrire seulement <image>. Néanmoins, pour ne pas les confondre avec le premier type de balise, les webmasters recommandent de rajouter ce / (*slash*) à la fin des balises orphelines. Vous me verrez donc mettre un / aux balises orphelines et je vous recommande de faire de même, c'est une bonne pratique.

2. Les attributs

Les attributs sont un peu les options des balises. Ils viennent les compléter pour donner des informations supplémentaires. L'attribut se place après le nom de la balise ouvrante et a le plus souvent une valeur, comme ceci :

```
<balise attribut="valeur">
```

À quoi cela sert-il ? Prenons la balise <image /> que nous venons de voir. Seule, elle ne sert pas à grand-chose. On pourrait rajouter un attribut qui indique le nom de l'image à afficher :

```
<image nom="photo.jpg" />
```

L'ordinateur comprend alors qu'il doit afficher l'image contenue dans le fichier photo.jpg.

Dans le cas d'une balise fonctionnant « par paire », on ne met les attributs que dans la balise ouvrante et pas dans la balise fermante. Par exemple, ce code indique que la citation est de Neil Armstrong et qu'elle date du 21 Juillet 1969 :

```
<citation auteur = "Neil Armstrong" date="21/07/1969">  
C'est un petit pas pour l'homme, mais un bond de géant pour  
l'humanité.  
</citation>
```

Toutes les balises que nous venons de voir sont fictives. Les vraies balises ont des noms en anglais (eh oui !), nous allons les découvrir dans la suite de ce cours.

3. Structure de base d'une page HTML5

Prenons un éditeur de texte (dans mon cas Notepad++) et créer le fichier page.html

Ce code correspond à la base d'une page web en HTML5 :

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

Vous noterez que les balises s'ouvrent et se ferment dans un ordre précis. Par exemple, la balise `<html>` est la première que l'on ouvre et c'est aussi la dernière que l'on ferme (tout à la fin du code, avec `</html>`).

Les balises doivent être fermées dans le sens inverse de leur ouverture. Un exemple :

`<html><body></body></html>` : **correct**. Une balise qui est ouverte à l'intérieur d'une autre doit aussi être fermée à l'intérieur.

`<html><body></html></body>` : **incorrect**, les balises s'entremêlent.

- Le doctype

```
<!DOCTYPE html>
```

La toute première ligne s'appelle le doctype. Elle est indispensable car c'est elle qui indique qu'il s'agit bien d'une page web HTML. Ce n'est pas vraiment une balise comme les autres (elle commence par un point d'exclamation), vous pouvez considérer que c'est un peu l'exception qui confirme la règle.

- La balise `</html>`

```
<html> ... </html>
```

C'est la balise principale du code. Elle englobe tout le contenu de votre page. Comme vous pouvez le voir, la balise fermante `</html>` se trouve tout à la fin du code !

- L'en-tête `<head>` et le corps `<body>`

Une page web est constituée de deux parties :

L'en-tête `<head>` : cette section donne quelques informations générales sur la page comme son titre, l'encodage (pour la gestion des caractères spéciaux), etc. Cette section est généralement assez courte. Les informations que contient l'en-tête ne sont pas affichées sur la page, ce sont simplement des informations générales à destination de l'ordinateur. Elles sont cependant très importantes !

Le corps `<body>` : c'est là que se trouve la partie principale de la page. Tout ce que nous écrirons ici sera affiché à l'écran. C'est à l'intérieur du corps que nous écrirons la majeure partie de notre code.

Pour le moment, le corps est vide (nous y reviendrons plus loin). Intéressons-nous par contre aux deux balises contenues dans l'en-tête...

- L'encodage (charset)

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

Cette balise indique l'encodage utilisé dans votre fichier .html.

Sans rentrer dans les détails, car cela pourrait vite devenir compliqué, l'encodage indique la façon dont le fichier est enregistré. C'est lui qui détermine comment les caractères spéciaux vont s'afficher (accents, idéogrammes chinois et japonais, caractères arabes, etc.).

Il y a plusieurs techniques d'encodage portant des noms bizarres et utilisées en fonction des langues : ISO-8859-1, OEM 775, Windows-1253... Une seule cependant devrait être utilisée aujourd'hui autant que possible : **UTF-8**. Cette méthode d'encodage permet d'afficher sans aucun problème pratiquement tous les symboles de toutes les langues de notre planète ! C'est pour cela que j'ai indiqué utf-8 dans cette balise.

Il faut aussi que votre fichier soit bien enregistré en UTF-8. C'est le cas le plus souvent sous Linux par défaut mais, sous Windows, il faut généralement le dire au logiciel.

Sous Notepad++, allez dans le menu Encodage > Encoder en UTF-8 (sans BOM) pour que votre fichier soit enregistré en UTF-8 dès le début. Cela ne s'applique qu'au fichier actuellement ouvert.

Pour ne pas avoir à le faire pour chaque nouveau fichier, je vous conseille d'aller dans le menu Paramétrage > Préférences, onglet Nouveau document/Dossier. Sélectionnez UTF-8 sans BOM dans la liste.

Si vous avez un problème d'affichage des accents plus tard sur votre page web, c'est qu'il y a un problème avec l'encodage. Vérifiez que la balise indique bien UTF-8 et que votre fichier est enregistré en UTF-8 (votre éditeur de texte est capable de vous le dire, Notepad++ le fait dans le menu Encodage).

- **Le titre principal de la page**

`<title>`

C'est le titre de votre page, probablement l'élément le plus important ! Toute page doit avoir un titre qui décrit ce qu'elle contient.

Il est conseillé de garder le titre assez court (moins de 100 caractères en général).

Le titre ne s'affiche pas dans votre page mais en haut de celle-ci (souvent dans l'onglet du navigateur). Enregistrez votre page web et ouvrez-la dans votre navigateur. Vous verrez que le titre s'affiche dans l'onglet.

- **Les commentaires**

Un **commentaire** en HTML est un texte qui sert simplement de mémo. Il n'est pas affiché, il n'est pas lu par l'ordinateur, cela ne change rien à l'affichage de la page.

Un commentaire est une balise HTML avec une forme bien spéciale :

`<!-- Ceci est un commentaire -->`

Vous pouvez le mettre où vous voulez au sein de votre code source : il n'a aucun impact sur votre page, mais vous pouvez vous en servir pour vous aider à vous repérer dans votre code source (surtout s'il est long).

Exemple:

```
<!DOCTYPE html>
<html>
  <head>
    <!-- En-tête de la page -->
    <meta charset="utf-8" />
    <title>Titre</title>
  </head>
  <body>
    <!-- Corps de la page -->
```



```
</body>  
</html>
```

Terminons par une remarque importante : *tout le monde peut voir le code HTML de votre page* une fois celle-ci mise en ligne sur le Web. Il suffit de faire un clic droit sur la page et de sélectionner « Afficher le code source de la page » (l'intitulé peut changer selon votre navigateur), comme le montre la figure suivante.

En résumé

- On utilise l'éditeur de texte (Notepad++, jEdit, vim...) pour créer un fichier ayant l'extension .html (par exemple : test.html). Ce sera notre page web.
- Ce fichier peut être ouvert dans le navigateur web simplement en faisant un double-clic dessus.
- À l'intérieur du fichier, nous écrirons le contenu de notre page, accompagné de balises HTML.
- Les balises peuvent avoir plusieurs formes :
 - <balise> </balise> : elles s'ouvrent et se ferment pour délimiter le contenu (début et fin d'un titre, par exemple).
 - <balise /> : balises orphelines (on ne les insère qu'en un seul exemplaire), elles permettent d'insérer un élément à un endroit précis (par exemple une image).
 - Les balises sont parfois accompagnées d'attributs pour donner des indications supplémentaires (exemple : <image nom="photo.jpg" />).
- Une page web est constituée de deux sections principales : un en-tête (<head>) et un corps (<body>).
- On peut afficher le code source de n'importe quelle page web en faisant un clic droit puis en sélectionnant Afficher le code source de la page.

4. Organiser son site

• Les paragraphes

La plupart du temps, lorsqu'on écrit du texte dans une page web, on le fait à l'intérieur de paragraphes. Le langage HTML propose justement la balise <p> pour délimiter les paragraphes.

```
<p>Bonjour et bienvenue sur mon site !</p>
```

- `<p>` signifie « Début du paragraphe » ;
- `</p>` signifie « Fin du paragraphe ».

Comme je vous l'ai dit au chapitre précédent, on écrit le contenu du site web entre les balises `<body>` `</body>`. Il nous suffit donc de mettre notre paragraphe entre ces deux balises et nous aurons enfin notre première vraie page web avec du texte !

Je reprends donc exactement le même code qu'au chapitre précédent et j'y ajoute mon paragraphe :

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8" />
    <title>Paragaphes</title>
  </head>

  <body>
    <p>Bonjour et bienvenue sur mon site !</p>
  </body>
</html>
```

- **Sautez une ligne**

Il existe une balise « Aller à la ligne » C'est une balise **orpheline** qui sert juste à indiquer qu'on doit aller à la ligne : `<br />`.

Vous devez obligatoirement la mettre à l'intérieur d'un paragraphe

En résumé:

- `<p>` `</p>` : pour organiser son texte en paragraphes ;
- `<br />` : pour aller à la ligne.

- Les titres

Lorsque le contenu de votre page va s'étoffer avec de nombreux paragraphes, il va devenir difficile pour vos visiteurs de se repérer. C'est là que les titres deviennent utiles.

En HTML, on est verni, on a le droit d'utiliser six niveaux de titres différents. Je veux dire par là qu'on peut dire « Ceci est un titre très important », « Ceci est un titre un peu moins important », « Ceci est un titre encore moins important », etc. On a donc six balises de titres différentes :

`<h1> </h1>` : signifie « titre très important ». En général, on s'en sert pour afficher le titre de la page au début de celle-ci.

`<h2> </h2>` : signifie « titre important ».

`<h3> </h3>` : pareil, c'est un titre un peu moins important (on peut dire un « sous-titre » si vous voulez).

`<h4> </h4>` : titre encore moins important.

`<h5> </h5>` : titre pas important.

`<h6> </h6>` : titre vraiment, mais alors là vraiment pas important du tout.

Attention : ne confondez pas avec la balise `<title>` ! La balise `<title>` affiche le titre de la page dans la barre de titre du navigateur comme nous l'avons vu. Les titres `<h1>` et compagnie, eux, servent à créer des titres qui seront affichés *dans* la page web.

Essayez de créer une page web avec des titres pour voir ce que cela donne (titre1.html):

```
<html>
  <head>
    <meta charset="utf-8" />
    <title>Niveaux de titres</title>
  </head>

  <body>
    <h1>Titre super important</h1>
    <h2>Titre important</h2>
    <h3>Titre un peu moins important (sous-titre)</h3>
```

```
<h4>Titre pas trop important</h4>
<h5>Titre pas important</h5>
<h6>Titre vraiment pas important du tout</h6>
</body>
</html>
```

Autre exemple (titre2.html):

```
<html>
  <head>
    <meta charset="utf-8" />
    <title>Présentation du Site Spécialité NSI </
title>  </head>
  <body>
<h1>Bienvenue sur le site de la spé NSI en première !</h1>
    <p>Bonjour et bienvenue sur mon site.<br />
      Le Site de la spécialité NSI, qu'est-ce que c'est ?</p>
    <h2>Des travaux pratiques pour découvrir la programmation et
l'informatique</h2>
    <p>Le Site de la spécialité NSI en seconde vous présente nos principales
réalisations. !</p>
  </body>
</html>
```

Info : Oui, mais moi je veux centrer mon titre, l'écrire en rouge et le souligner !

Grâce au langage CSS, vous pourrez dire « Je veux que mes titres importants soient centrés, rouges et soulignés ». Pour le moment, en HTML, nous ne faisons que structurer notre page. *Nous rédigeons le contenu avant de nous amuser à le mettre en forme.*

- **La mise en valeur**

Au sein de vos paragraphes, certains mots sont parfois plus importants que d'autres et vous aimeriez les faire ressortir. HTML vous propose différents moyens de mettre en valeur le texte de votre page.

Pour mettre un peu en valeur votre texte, vous devez utiliser la balise `<em>` `</em>`. Son utilisation est très simple : encadrez les mots à mettre en valeur avec ces balises et

c'est bon ! Je reprends un peu l'exemple de tout à l'heure et j'y mets quelques mots en évidence :

```
<html>
<head>
  <meta charset="utf-8" />
  <title>Emphase</title>
</head>
<body>
  <p>Bonjour et bienvenue sur mon site !<br />
  Ceci est mon premier test alors <em>soyez indulgents</em>
  s'il vous plaît, j'apprends petit à petit comment cela
  marche.</p>
</body>
</html>
```

Comme vous pouvez le voir, utiliser la balise `<em>` a pour conséquence de mettre le texte en *italique*. En fait, c'est le navigateur qui choisit comment afficher les mots. On lui dit que les mots sont assez importants et, pour faire ressortir cette information, il change l'apparence du texte en utilisant l'italique.

Mettre bien en valeur

Pour mettre un texte bien en valeur, on utilise la balise `<strong>` qui signifie « **fort** », ou « **important** » si vous préférez. Elle s'utilise exactement de la même manière que `<em>` :

```
<html>
<head>
  <meta charset="utf-8" />
  <title>Forte emphase</title>
</head>
<body>
  <p>Bonjour et bienvenue sur mon site !<br />
  Ceci est mon premier test alors <strong>soyez indulgents</strong> s'il
  vous plaît, j'apprends petit à petit comment cela marche.</p>
</body>
</html>
```

Vous voyez sûrement le texte s'afficher en gras. Là encore, le gras n'est qu'une *conséquence*. Le navigateur a choisi d'afficher en gras les mots importants pour les faire ressortir davantage.

La balise `<strong>` ne signifie pas « mettre en gras » mais « important ». On pourra décider plus tard, en CSS, d'afficher les mots « importants » d'une autre façon que le gras si on le souhaite.

Marquer le texte

La balise `<mark>` permet de faire ressortir visuellement une portion de texte. L'extrait n'est pas forcément considéré comme important mais on veut qu'il se distingue bien du reste du texte. Cela peut être utile pour faire ressortir un texte pertinent après une recherche sur votre site par exemple.

```
<html>
<head>
  <meta charset="utf-8" />
  <title>Marquage du texte</title>
</head>

<body>
  <p>Bonjour et bienvenue sur mon site !<br />
  Ceci est mon premier test alors <mark>soyez indulgents</mark> s'il vous
  plaît, j'apprends petit à petit comment cela marche.</p>
</body>
</html>
```

Par défaut, `<mark>` a pour effet de surligner le texte. On pourra changer l'affichage en CSS (décider de surligner dans une autre couleur, d'encadrer le texte, etc.). C'est le même principe que ce que je vous disais pour les balises précédentes : elles indiquent le *sens* des mots et non pas comment ceux-ci doivent s'afficher.

N'oubliez pas : HTML pour le fond, CSS pour la forme

5. Les listes

Les listes nous permettent souvent de mieux structurer notre texte et d'ordonner nos informations.

Nous allons découvrir ici deux types de listes :

- les listes non ordonnées ou listes à puces ;
- les listes ordonnées ou listes numérotées ou encore énumérations.

Liste non ordonnée

Une liste non ordonnée ressemble à ceci :

- Fraises
- Framboises
- Cerises

C'est un système qui nous permet de créer une liste d'éléments sans notion d'ordre (il n'y a pas de « premier » ni de « dernier »). Créer une liste non ordonnée est très simple. Il suffit d'utiliser la balise `<ul>` que l'on referme un peu plus loin avec `</ul>`.

Commencez donc à taper ceci :

```
<ul> </ul>
```

Et maintenant, voilà ce qu'on va faire : on va écrire chacun des éléments de la liste entre deux balises `<li></li>`. Chacune de ces balises doit se trouver entre `<ul>` et `</ul>`. Vous allez comprendre de suite avec cet exemple :

```
<ul>  
  <li>Fraises</li>  
  <li>Framboises</li>  
  <li>Cerises</li>  
</ul>
```

Notez que la liste doit être placée à l'intérieur de `<body></body>`. À partir de maintenant, je ne mets pas tout le code de la page pour rester lisible.

Retenez donc ces deux balises :

- `<ul></ul>` délimite toute la liste ;
- `<li></li>` délimite un élément de la liste (une puce).

Vous pouvez mettre autant d'éléments que vous voulez dans la liste à puces, vous n'êtes pas limités à trois éléments.

Et voilà, vous savez créer une liste à puces ! Pas si dur une fois qu'on a compris comment imbriquer les balises.

Liste ordonnée

Une liste ordonnée fonctionne de la même façon, seule une balise change : il faut remplacer `<ul></ul>` par `<ol></ol>`.

À l'intérieur de la liste, on ne change rien : on utilise toujours des balises `<li></li>` pour délimiter les éléments.

L'ordre dans lequel vous placez les éléments de la liste est important. Le premier `<li></li>` sera l'élément n° 1, le second sera le n°2 etc...

Comme c'est particulièrement intuitif, je vous laisse admirer la simplicité de cet exemple:

```
<h1>Ma journée</h1>
```

```
<ol>
  <li>Je me lève.</li>
  <li>Je mange et je bois.</li>
  <li>Je retourne me coucher.</li>
</ol>
```

En résumé

- Le HTML comporte de nombreuses balises qui nous permettent d'organiser le texte de notre page. Ces balises donnent des indications comme « Ceci est un paragraphe », « Ceci est un titre », etc.
- Les paragraphes sont définis par la balise `<p> </p>` et les sauts de ligne par la balise `<br />`.
- Il existe six niveaux de titre, de `<h1> </h1>` à `<h6> </h6>`, à utiliser selon l'importance du titre.
- On peut mettre en valeur certains mots avec les balises `<strong>`, `<em>` et `<mark>`.
- Pour créer des listes, on doit utiliser la balise `<ul>` (liste à puces, non ordonnée) ou `<ol>` (liste ordonnée). À l'intérieur, on insère les éléments avec une balise `<li>` pour chaque item.

6. Les liens

Comme vous le savez, un site web est composé de plusieurs pages. Comment faire pour aller d'une page vers une autre ? À l'aide de liens pardi ! Dans ce chapitre, nous allons justement apprendre à créer des liens entre nos pages.

Je suppose que chacun d'entre vous sait ce qu'est un lien : il s'agit d'un texte sur lequel on peut cliquer pour se rendre sur une autre page.

On peut faire un lien d'une page a.html vers une page b.html, mais on peut aussi faire un lien vers un autre site (par exemple, <http://www.sncf.com>). Dans les deux cas, nous allons voir que le fonctionnement est le même.

- Un lien vers un autre site

Il est facile de reconnaître les liens sur une page : ils sont écrits d'une façon différente (par défaut, en bleu et soulignés) et un curseur en forme de main apparaît lorsqu'on pointe dessus.

Pour faire un lien, la balise que nous allons utiliser est très simple à retenir : `<a>`. Il faut cependant lui ajouter un attribut, href, pour indiquer vers quelle page le lien doit conduire.

Par exemple, le code ci-dessous est un lien qui amène vers le site de la sncf, situé à l'adresse <http://www.sncf.com> :

```
<a href="http://www.sncf.com">Site de la sncf</a>
```

Nous allons placer ce lien au sein d'un paragraphe. Voici donc comment reproduire l'exemple de la figure précédente :

```
<p>Bonjour. Souhaitez-vous visiter le <a href="http://www.sncf.com">Site de la Sncf</a> ?<br />C'est un bon site pour voyager ! ;-)</p>
```

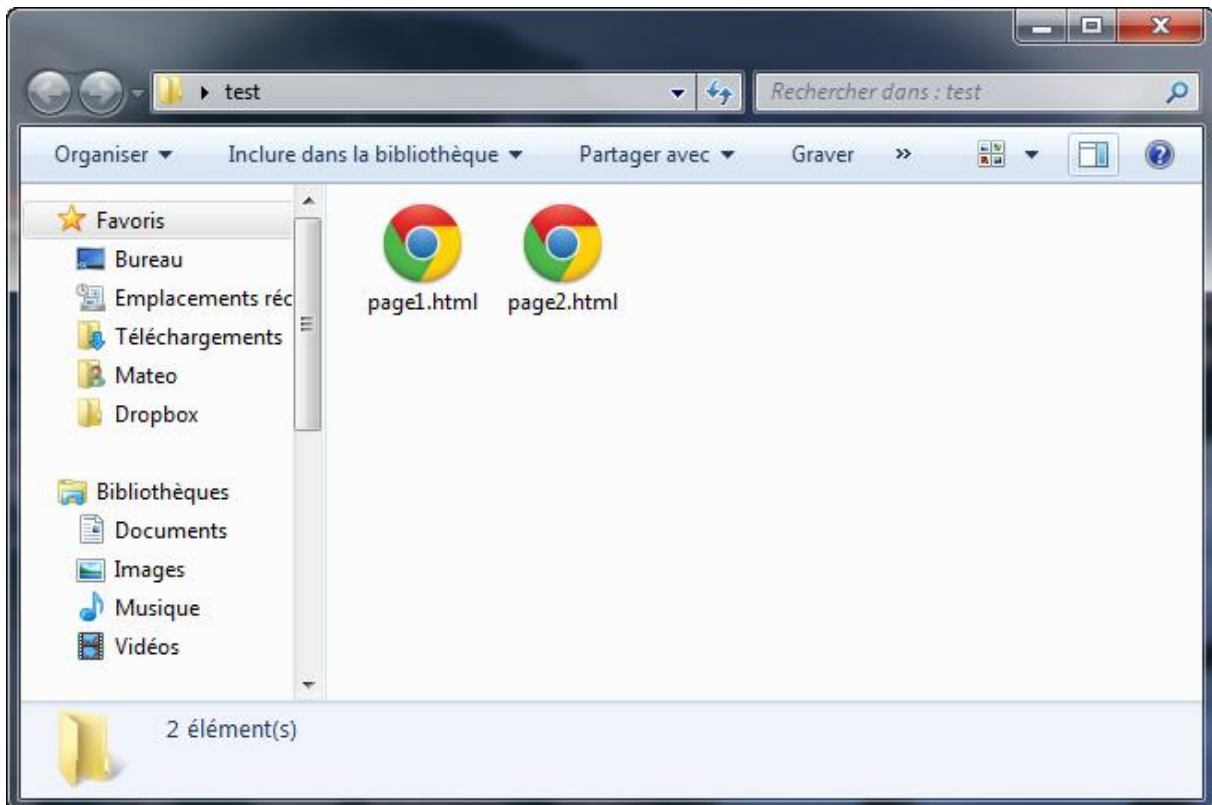
Les liens que nous venons de voir sont appelés **liens absolus** car on indique l'adresse complète. Nous allons maintenant voir que l'on peut écrire les liens d'une façon un peu différente, ce qui va nous être utile pour faire des liens entre les pages de notre site.

- Un lien vers une autre page de son site

Deux pages situées dans un même dossier

Pour commencer, nous allons créer deux fichiers correspondant à deux pages HTML différentes. Comme je suis très inspiré, je vous propose de les appeler page1.html et page2.html.

Nous aurons donc ces deux fichiers sur notre disque *dans le même dossier* (figure suivante).



Comment faire un lien de la page 1 vers la page 2, sans avoir d'adresse en `http://` ? En fait, c'est facile : si les deux fichiers sont situés dans le même dossier, il suffit d'écrire comme cible du lien le nom du fichier vers lequel on veut amener.

Par exemple : `<a href="page2.html">`. On dit que c'est un **lien relatif**.

Voici le code que nous allons utiliser dans nos fichiers `page1.html` et `page2.html`.

`page1.html`

```
<p>Bonjour. Souhaitez-vous consulter <a href="page2.html">la page 2</a>
?</p>
```

`page2.html`

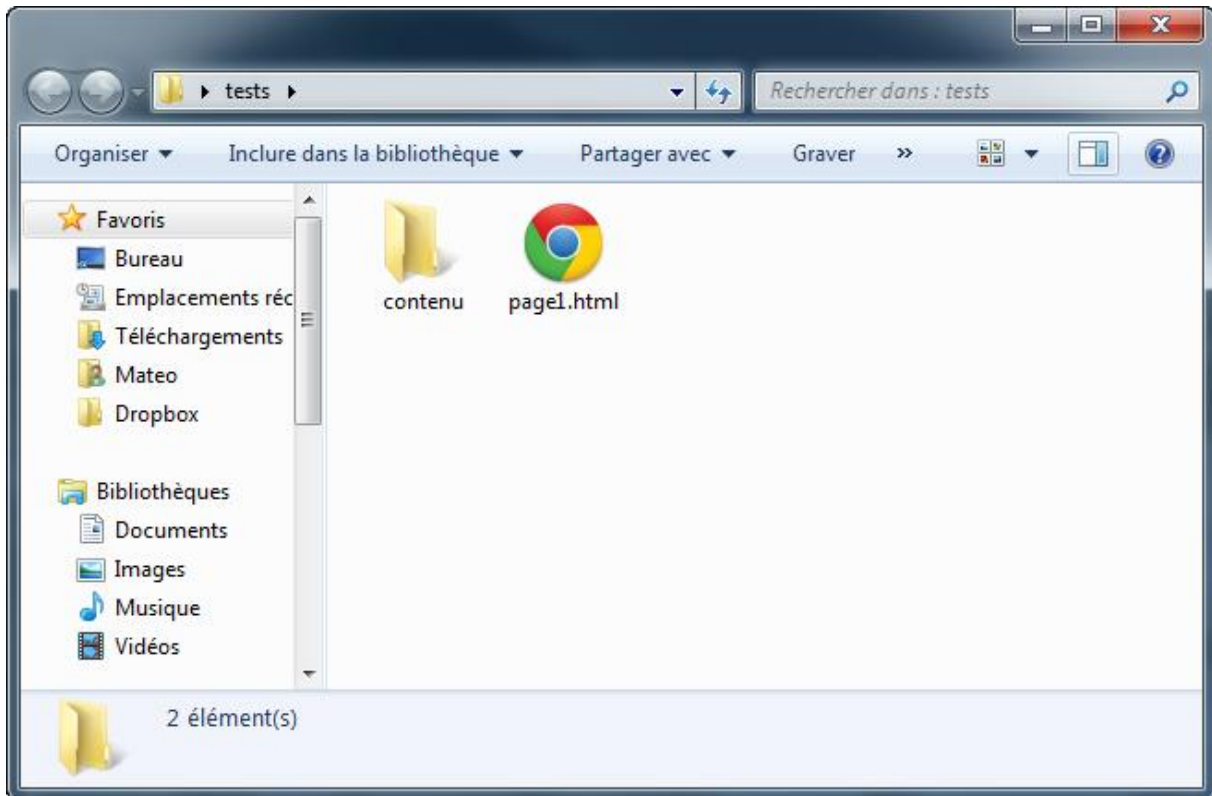
La page 2 (page d'arrivée) affichera simplement un message pour indiquer que l'on est bien arrivé sur la page 2 :

```
<h1>Bienvenue sur la page 2 !</h1>
```

Deux pages situées dans des dossiers différents

Les choses se corsent un petit peu si les pages sont situées dans des dossiers différents. Idéalement, elles ne devraient pas être trop loin l'une de l'autre (dans un sous-dossier par exemple).

Imaginons que page2.html se trouve dans un sous-dossier appelé contenu, comme à la figure suivant



Dans ce cas de figure, le lien doit être rédigé comme ceci :

```
<a href="contenu/page2.html">
```

S'il y avait plusieurs sous-dossiers, on écrirait ceci :

```
<a href="contenu/autredossier/page2.html">
```

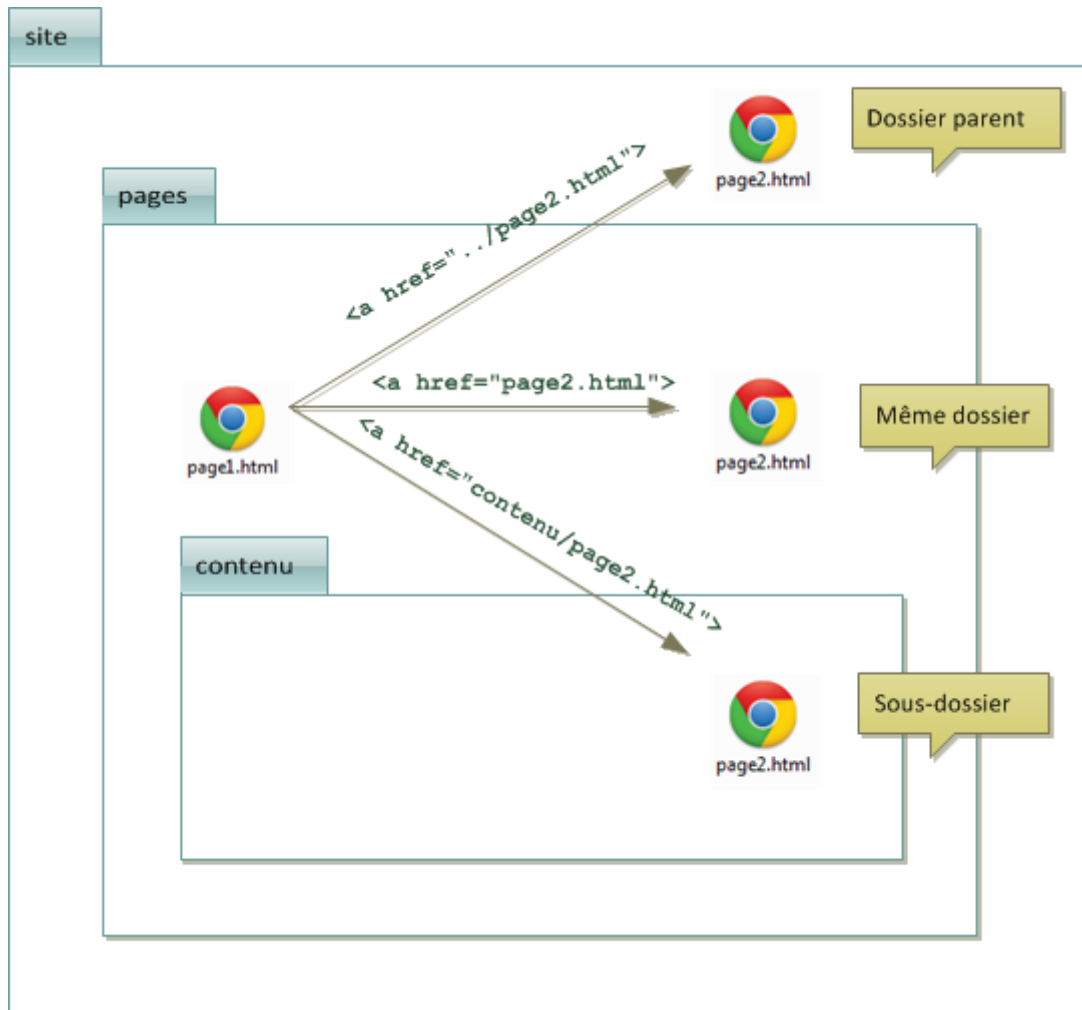
Si votre fichier cible est placé dans un dossier qui se trouve « plus haut » dans l'arborescence, il faut écrire deux points comme ceci :

```
<a href="../page2.html">
```

Résumé en images

Les liens relatifs ne sont pas bien compliqués à utiliser une fois qu'on a compris le principe. Il suffit de regarder dans quel « niveau de dossier » se trouve votre fichier

cible pour savoir comment écrire votre lien. La figure suivante fait la synthèse des différents liens relatifs possibles.



- Un lien vers une ancre

Une **ancre** est une sorte de point de repère que vous pouvez mettre dans vos pages HTML lorsqu'elles sont très longues.

En effet, il peut alors être utile de faire un lien amenant plus bas dans la même page pour que le visiteur puisse sauter directement à la partie qui l'intéresse.

Pour créer une ancre, il suffit de rajouter l'attribut **id** à une balise qui va alors servir de repère. Ce peut être n'importe quelle balise, un titre par exemple.

Utilisez l'attribut **id** pour donner un nom à l'ancre. Cela nous servira ensuite pour faire un lien vers cette ancre. Par exemple :

```
<h2 id="mon_ancre">Titre</h2>
```

Ensuite, il suffit de créer un lien comme d'habitude, mais cette fois l'attribut href contiendra un dièse (#) suivi du nom de l'ancre. Exemple :

```
<a href="#mon_ancre">Aller vers l'ancre</a>
```

Normalement, si vous cliquez sur le lien, cela vous amènera plus bas dans la même page (à condition que la page comporte suffisamment de texte pour que les barres de défilement se déplacent automatiquement).

Voici un exemple de page comportant beaucoup de texte et utilisant les ancres (j'ai mis n'importe quoi dans le texte pour remplir) :

```
<h1>Ma grande page</h1>
```

```
<p>
```

```
  Aller directement à la partie traitant de :<br />
```

```
  <a href="#cuisine">La cuisine</a><br />
```

```
  <a href="#rollers">Les rollers</a><br />
```

```
  <a href="#arc">Le tir à l'arc</a><br />
```

```
</p>
```

```
<h2 id="cuisine">La cuisine</h2>
```

```
<p>... (beaucoup de texte) ...</p>
```

```
<h2 id="rollers">Les rollers</h2>
```

```
<p>... (beaucoup de texte) ...</p>
```

```
<h2 id="arc">Le tir à l'arc</h2>
```

```
<p>... (beaucoup de texte) ...</p>
```

S'il ne se passe rien quand vous cliquez sur les liens, c'est qu'il n'y a pas assez de texte. Dans ce cas, vous pouvez soit rajouter du blabla dans la page pour qu'il y ait (encore) plus de texte, soit réduire la taille de la fenêtre de votre navigateur pour faire apparaître les barres de défilement sur le côté.

L'attribut id sert à donner un nom « unique » à une balise, pour s'en servir de repère. Et, croyez-moi, vous n'avez pas fini d'entendre parler de cet attribut. Ici, on s'en sert

pour faire un lien vers une ancre mais, en CSS, il nous sera très utile pour « repérer » une balise précise, vous verrez.

Évitez cependant de créer des id avec des espaces ou des caractères spéciaux, utilisez simplement, dans la mesure du possible, des lettres et chiffres pour que la valeur soit reconnue par tous les navigateurs.

- **Lien vers une ancre située dans une autre page**

Alors là je vous préviens, on va faire le Mégamix !

L'idée, c'est de faire un lien qui ouvre une autre page ET qui amène directement à une ancre située plus bas sur cette page.

En pratique c'est assez simple à faire : il suffit de taper le nom de la page, suivi d'un dièse (#), suivi du nom de l'ancre.

Par exemple : `<a href="ancres.html#rollers">`

... vous amènera sur la page ancres.html, directement au niveau de l'ancre appelée rollers.

Voici une page qui contient trois liens, chacun amenant vers une des ancres de la page de l'exemple précédent :

```
<h1>Le Mégamix</h1>
<p>
  Rendez-vous quelque part sur une autre page :<br />
  <a href="ancres.html#cuisine">La cuisine</a><br />
  <a href="ancres.html#rollers">Les rollers</a><br />
  <a href="ancres.html#arc">Le tir à l'arc</a><br />
</p>
```

- **Un lien pour envoyer un e-mail**

Si vous voulez que vos visiteurs puissent vous envoyer un e-mail, vous pouvez utiliser des liens de type mailto. Rien ne change au niveau de la balise, vous devez simplement modifier la valeur de l'attribut href comme ceci :

```
<p><a href="mailto:votrenom@bidule.com">Envoyez-moi un e-mail !</a></p>
```

Il suffit donc de faire commencer le lien par mailto: et d'écrire l'adresse e-mail où on peut vous contacter. Si vous cliquez sur le lien, un nouveau message vide s'ouvre, prêt à être envoyé à votre adresse e-mail.

En résumé

- Les liens permettent de changer de page et sont, par défaut, écrits en bleu et soulignés.
- Pour insérer un lien, on utilise la balise `<a>` avec l'attribut `href` pour indiquer l'adresse de la page cible. Exemple : `<a href="http://www.sncf.com">`.
- On peut faire un lien vers une autre page de son site simplement en écrivant le nom du fichier : `<a href="page2.html">`.
- Les liens permettent aussi d'amener vers d'autres endroits sur la même page. Il faut créer une ancre avec l'attribut `id` pour « marquer » un endroit dans la page, puis faire un lien vers l'ancre comme ceci : `<a href="#ancre">`.

7. Les images

Insérer une image dans une page web ? Vous allez voir, c'est d'une facilité déconcertante... Enfin presque. Il existe différents **formats** d'image que l'on peut utiliser sur des sites web, et on ne doit pas les choisir au hasard. En effet, les images sont parfois volumineuses à télécharger, ce qui ralentit le temps de chargement de la page (beaucoup plus que le texte !).

- **Insertion d'une image**

Quelle est la fameuse balise qui va nous permettre d'insérer une image ? Il s'agit de... `<img />` !

C'est une balise de type **orpheline** (comme `<br />`). Cela veut dire qu'on n'a pas besoin de l'écrire en deux exemplaires comme la plupart des autres balises que nous avons vues jusqu'ici. En effet, nous n'avons pas besoin de délimiter une portion de texte, nous voulons juste insérer une image à un endroit précis.

La balise doit être accompagnée de deux attributs obligatoires :

src : il permet d'indiquer où se trouve l'image que l'on veut insérer. Vous pouvez soit mettre un chemin absolu (ex. : `http://www.site.com/fleur.png`), soit mettre le chemin en relatif (ce qu'on fait le plus souvent). Ainsi, si votre image est dans un sous-dossier `images`, vous devrez taper : `src="images/fleur.png"`

alt : cela signifie « texte alternatif ». On doit *toujours* indiquer un texte alternatif à l'image, c'est-à-dire un court texte qui décrit ce que contient l'image. Ce texte sera affiché à la place de l'image si celle-ci ne peut pas être téléchargée (cela arrive), ou

dans les navigateurs de personnes handicapées (non-voyants) qui ne peuvent malheureusement pas « voir » l'image. Cela aide aussi les robots des moteurs de recherche pour les recherches d'images. Pour la fleur, on mettrait par exemple : `alt="Une fleur"`.

Les images doivent se trouver obligatoirement à l'intérieur d'un paragraphe (`<p></p>`). Voici un exemple d'insertion d'image :

```
<p>  
    Voici une photo que j'ai prise lors de mes dernières vacances à la  
montagne :<br />  
      
</p>
```

Bref, l'insertion d'image est quelque chose de très facile pour peu qu'on sache indiquer où se trouve l'image, comme on avait appris à le faire avec les liens.

La plus grosse « difficulté » (si on peut appeler cela une difficulté) consiste à choisir le bon format d'image. Ici, c'est une photo donc c'est évidemment le format JPEG qu'on utilise.

Je le répète : évitez à tout prix les accents, majuscules et espaces dans vos noms de fichiers et de dossiers. Voici un chemin qui va poser problème :

`"Images du site/Image toute bête.jpg"`.

Il faudrait supprimer les espaces (ou les remplacer par le symbole « `_` »), supprimer les accents et tout mettre en minuscules comme ceci :

`"images_du_site/image_toute_bete.jpg"`.

Sachez donc que, si votre image ne s'affiche pas, c'est très certainement parce que le chemin est incorrect ! Simplifiez au maximum vos noms de fichiers et de dossiers, et tout ira bien

Miniature cliquable

Si votre image est très grosse, il est conseillé d'en afficher la miniature sur votre site. Ajoutez ensuite un lien sur cette miniature pour que vos visiteurs puissent afficher l'image en taille originale.

Il existe des millions de logiciels permettant de créer des miniatures d'images. J'utilise personnellement [Easy Thumbnails](#). Je vais ainsi disposer de deux versions de ma photo, comme à la figure suivante) : la miniature et l'image d'origine

Je les place toutes les deux dans un dossier appelé par exemple img. J'affiche la version montagne_mini.jpg sur ma page et je fais un lien vers montagne.jpg pour que l'image agrandie s'affiche lorsqu'on clique sur la miniature.

Voici le code HTML que je vais utiliser pour cela :

```
<p>
```

```
    Vous souhaitez voir l'image dans sa taille d'origine ? Cliquez dessus  
!<br />
```

```
    <a href="img/montagne.jpg"></a>
```

```
</p>
```

En résumé

- Il existe plusieurs formats d'images adaptées au Web : JPEG : pour les photos ; PNG : pour toutes les autres illustrations ; GIF : similaire au PNG, plus limité en nombre de couleurs mais qui peut être animé.
- On insère une image avec la balise . Elle doit obligatoirement comporter au moins ces deux attributs : src (nom de l'image) et alt (courte description de l'image).

8. Les tableaux

La première balise à connaître est <table> </table>. C'est cette balise qui permet d'indiquer le début et la fin d'un tableau.

Cette balise est de type bloc, il faut donc la placer en dehors d'un paragraphe.

Exemple :

```
<p>Ceci est un paragraphe avant le tableau.</p>
```

```
<table>
```

```
    <!-- Ici, on écrira le contenu du tableau -->  
</table>
```

```
<p>Ceci est un paragraphe après le tableau.</p>
```

Voici deux nouvelles balises très importantes :

- `<tr>` `</tr>` : indique le début et la fin d'une ligne du tableau ;
- `<td>` `</td>` : indique le début et la fin du contenu d'une cellule.

En HTML, un tableau se construit ligne par ligne. Dans chaque ligne (`<tr>`), on indique le contenu des différentes cellules (`<td>`).

Exemple :

```
<table>
  <tr>
    <td>Carmen</td>
    <td>33 ans</td>
    <td>Espagne</td>
  </tr>
  <tr>
    <td>Michelle</td>
    <td>26 ans</td>
    <td>États-Unis</td>
  </tr>
</table>
```

Carmen	33 ans	Espagne
Michelle	26 ans	Etats-Unis

Un tableau sans CSS paraît bien vide.

Heureusement, il existe une **propriété CSS** spécifique aux tableaux, `border-collapse`, qui signifie « coller les bordures entre elles ».

Cette propriété peut prendre deux valeurs :

- `collapse` : les bordures seront collées entre elles, c'est l'effet qu'on recherche ici ;
- `separate` : les bordures seront dissociées (valeur par défaut)

extrait fichier.css

```
table
{
  border-collapse: collapse;
}

td
{
  border: 1px solid black;
}
```

Carmen	33 ans	Espagne
Michelle	26 ans	Etats-Unis

La ligne d'en-tête

Maintenant que l'on a ce qu'on voulait, on va rajouter la ligne d'en-tête du tableau. Dans l'exemple ci-dessous, les en-têtes sont « Nom », « Âge » et « Pays ».

La ligne d'en-tête se crée avec un `<tr>` comme on l'a fait jusqu'ici, mais les cellules qu'elle contient sont, cette fois, encadrées par des balises `<th>` et non pas `<td>` !

```
<table>
  <tr>
    <th>Nom</th>
    <th>Âge</th>
    <th>Pays</th>
  </tr>

  <tr>
    <td>Carmen</td>
    <td>33 ans</td>
    <td>Espagne</td>
  </tr>
  <tr>
    <td>Michelle</td>
    <td>26 ans</td>
    <td>États-Unis</td>
  </tr>
</table>
```

La ligne d'en-tête est très facile à reconnaître pour deux raisons :

- les cellules sont des `<th>` au lieu des `<td>` habituels ;
- c'est la première ligne du tableau (c'est idiot, mais encore faut-il le préciser).

Comme le nom des cellules est un peu différent pour l'en-tête, il faut penser à mettre à jour le CSS pour lui dire d'appliquer une bordure sur les cellules normales *et* sur l'en-tête (figure suivante).

extrait fichier.css

```
table
{
  border-collapse: collapse;
}
td, th /* Mettre une bordure sur les td ET les th */
{
  border: 1px solid black;
}
```

Comme vous pouvez le constater, votre navigateur a mis en gras le texte des cellules d'en-tête. C'est ce que font en général les navigateurs mais, si vous le désirez, vous pouvez changer cela à coups de CSS : modifier la couleur de fond des cellules d'en-tête, leur police, leur bordure, etc.

Nom	Age	Pays
Carmen	33 ans	Espagne
Michelle	26 ans	Etats-Unis

Titre du tableau

Normalement, tout tableau doit avoir un titre. Le titre permet de renseigner rapidement le visiteur sur le contenu du tableau.

Le titre d'un tableau se précise grâce à la balise `<caption>`. Cette balise se place tout au début du tableau, juste avant l'en-tête. C'est elle qui contient le titre du tableau (figure suivante) :

```
<table>
  <caption>Passagers du vol 377</caption>

  <tr>
    <th>Nom</th>
    <th>Âge</th>
    <th>Pays</th>
  </tr>
  <tr>
    <td>Carmen</td>
    <td>33 ans</td>
    <td>Espagne</td>
  </tr>
  <tr>
    <td>Michelle</td>
    <td>26 ans</td>
    <td>États-Unis</td>
  </tr>
</table>
```

Sachez que vous pouvez changer la position du titre avec la **propriété CSS `caption-side`** qui peut prendre deux valeurs :

- `top` : le titre sera placé au-dessus du tableau (par défaut) ;
- `bottom` : le titre sera placé en dessous du tableau.

Passagers du vol 377

Nom	Age	Pays
Carmen	33 ans	Espagne
Michelle	26 ans	Etats-Unis

Un tableau structuré

Nous avons appris à construire des petits tableaux simples. Ces petits tableaux suffisent dans la plupart des cas, mais il arrivera que vous ayez besoin de réaliser des tableaux plus... complexes.

Nous allons découvrir deux techniques particulières :

- Pour les gros tableaux, il est possible de les **diviser** en trois parties :
 - En-tête ;
 - Corps du tableau ;
 - Pied de tableau.
- Pour certains tableaux, il se peut que vous ayez besoin de **fusionner** des cellules entre elles.

Diviser un gros tableau

Si votre tableau est assez gros, vous aurez tout intérêt à le découper en plusieurs parties. Pour cela, il existe des balises HTML qui permettent de définir les trois « zones » du tableau :

- **l'en-tête (en haut)** : il se définit avec les balises <thead></thead> ;
- **le corps (au centre)** : il se définit avec les balises <tbody></tbody> ;
- **le pied du tableau (en bas)** : il se définit avec les balises <tfoot></tfoot>.

Exemple

```
<table>
  <caption>Passagers du vol 377</caption>

  <thead> <!-- En-tête du tableau -->
    <tr>
      <th>Nom</th>
      <th>Âge</th>
      <th>Pays</th>
    </tr>
  </thead>

  <tbody> <!-- Corps du tableau -->
    <tr>
      <td>Carmen</td>
      <td>33 ans</td>
```

```
        <td>Espagne</td>
    </tr>
    <tr>
        <td>Michelle</td>
        <td>26 ans</td>
        <td>États-Unis</td>
    </tr>
    <tr>
        <td>François</td>
        <td>43 ans</td>
        <td>France</td>
    </tr>
    <tr>
        <td>Martine</td>
        <td>34 ans</td>
        <td>France</td>
    </tr>
    <tr>
        <td>Jonathan</td>
        <td>13 ans</td>
        <td>Australie</td>
    </tr>
    <tr>
        <td>Xu</td>
        <td>19 ans</td>
        <td>Chine</td>
    </tr>
</tbody>

<tfoot> <!-- Pied de tableau -->
    <tr>
        <th>Nom</th>
        <th>Âge</th>
        <th>Pays</th>
    </tr>
</tfoot>

</table>
```

Fusionner des cellules

Dans certains tableaux complexes, vous aurez besoin de « fusionner » des cellules entre elles.

Un exemple de fusion ? Regardez le tableau à la figure suivante, qui dresse une liste de films et indique à qui ils s'adressent.

Pour effectuer une fusion, on rajoute un attribut à la balise <td>. Il faut savoir qu'il existe deux types de fusion :

- **La fusion de colonnes** : c'est ce que je viens de faire dans cet exemple. La fusion s'effectue horizontalement.

On utilisera l'attribut `colspan`.

- **La fusion de lignes** : là, deux lignes seront groupées entre elles. La fusion s'effectuera verticalement.

On utilisera l'attribut `rowspan`.

Comme vous le savez, vous devez donner une valeur à l'attribut (que ce soit `colspan` ou `rowspan`). Il faut indiquer le nombre de cellules à fusionner entre elles.

Si nous devons fusionner deux cellules, nous écrirons : `<td colspan="2">`

En résumé

- Un tableau s'insère avec la balise `<table>` et se définit ligne par ligne avec `<tr>`.
- Chaque ligne comporte des cellules `<td>` (cellules normales) ou `<th>` (cellules d'en-tête).
- Le titre du tableau se définit avec `<caption>`.
- On peut ajouter une bordure aux cellules du tableau avec `border`. Pour fusionner les bordures, on utilise la propriété CSS `border-collapse`.
- Un tableau peut être divisé en trois sections : `<thead>` (en-tête), `<tbody>` (corps) et `<tfoot>` (bas du tableau). L'utilisation de ces balises n'est pas obligatoire.
- On peut fusionner des cellules horizontalement avec l'attribut `colspan` ou verticalement avec `rowspan`. Il faut indiquer combien de cellules doivent être fusionnées.

9. Mise en forme avancée d'une page web

Nous approchons de plus en plus du but. Si nos pages web ne ressemblent pas encore tout à fait aux sites web que nous connaissons, c'est qu'il nous manque les connaissances nécessaires pour faire la mise en page.

En général, une page web est constituée d'un en-tête (tout en haut), de menus de navigation (en haut ou sur les côtés), de différentes sections au centre... et d'un pied de page (tout en bas).

Dans ce chapitre, nous allons nous intéresser aux nouvelles balises HTML dédiées à la structuration du site. Ces balises ont été introduites par HTML5 (elles n'existaient pas avant) et vont nous permettre de dire : « Ceci est mon en-tête », « Ceci est mon menu de navigation », etc.

Les Balises structurantes HTML5

<header> : l'en-tête

La plupart des sites web possèdent en général un en-tête, appelé *header* en anglais. On y trouve le plus souvent un logo, une bannière, le slogan de votre site...

Vous devrez placer ces informations à l'intérieur de la balise <header> :

```
<header>
  <!-- Placez ici le contenu de l'en-tête de votre page -->
</header>
```

L'en-tête peut contenir tout ce que vous voulez : images, liens, textes...

Il peut y avoir plusieurs en-têtes dans votre page. Si celle-ci est découpée en plusieurs sections, chaque section peut en effet avoir son propre <header>.

<footer> : le pied de page

À l'inverse de l'en-tête, le pied de page se trouve en général tout en bas du document. On y trouve des informations comme des liens de contact, le nom de l'auteur, les mentions légales, etc.

```
<footer>
  <!-- Placez ici le contenu du pied de page -->
</footer>
```

<nav> : principaux liens de navigation

La balise <nav> doit regrouper tous les principaux liens de navigation du site. Vous y placerez par exemple le menu principal de votre site.

Généralement, le menu est réalisé sous forme de liste à puces à l'intérieur de la balise `<nav>` :

```
<nav>
  <ul>
    <li><a href="index.html">Accueil</a></li>
    <li><a href="forum.html">Forum</a></li>
    <li><a href="contact.html">Contact</a></li>
  </ul>
</nav>
```

`<section>` : une section de page

La balise `<section>` sert à regrouper des contenus en fonction de leur thématique. Elle englobe généralement une portion du contenu au centre de la page.

```
<section>
  <h1>Ma section de page</h1>
  <p>Bla bla bla bla</p>
</section>
```

`<aside>` : informations complémentaires

La balise `<aside>` est conçue pour contenir des informations complémentaires au document que l'on visualise. Ces informations sont généralement placées sur le côté (bien que ce ne soit pas une obligation).

```
<aside>
  <!-- Placez ici des informations complémentaires -->
</aside>
```

Il peut y avoir plusieurs blocs `<aside>` dans la page

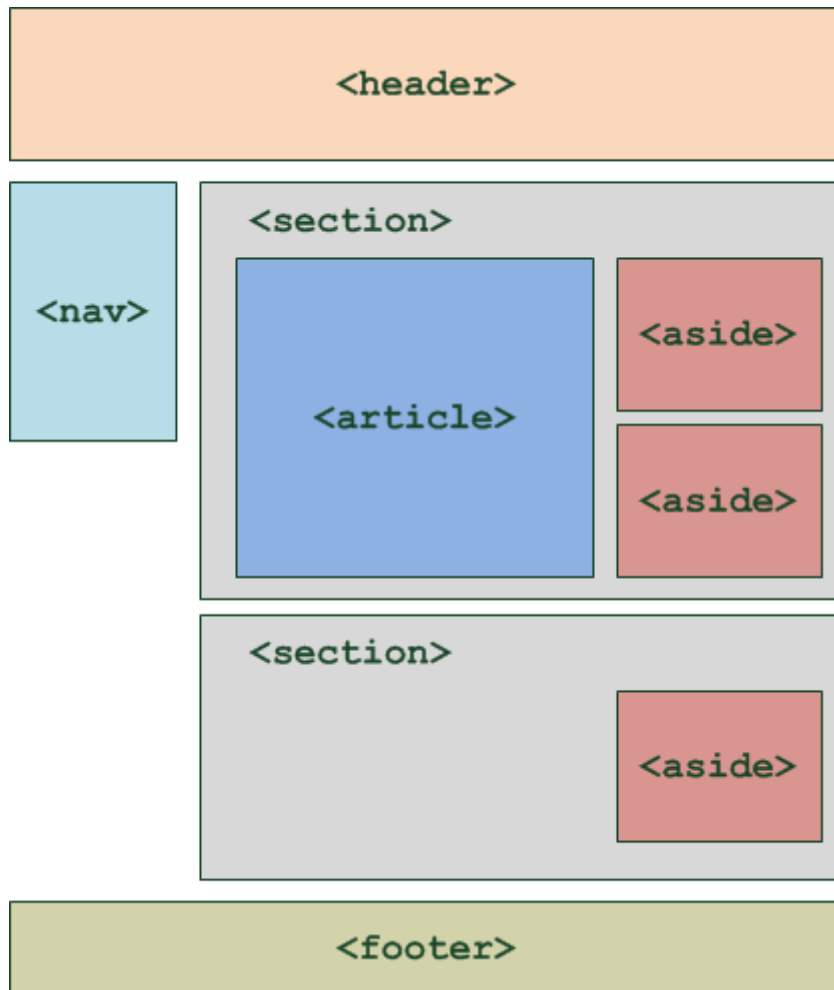
`<article>` : un article indépendant

La balise `<article>` sert à englober une portion généralement autonome de la page. C'est une partie de la page qui pourrait ainsi être reprise sur un autre site. C'est le cas par exemple des actualités (articles de journaux ou de blogs).

```
<article>
  <h1>Mon article</h1>
  <p>Bla bla bla bla</p>
</article>
```

Résumé

Un exemple de présentation :



Autre méthode : utiliser la balise div

La balise `<div>` est un conteneur (à appréhender comme un cadre) qui peut inclure tous les tags html (tels que les paragraphes les tableaux, les listes, les images ..., et également d'autres divisions).

Cet élément est l'un des plus utilisé pour structurer les pages html en plusieurs grandes sections (en tête, partie principale, partie navigation, pied de page ...).

En dehors de cet élément, les listes sont également très utilisées elles aussi pour structurer une page html, sous condition de leur associer des styles CSS adapté.

Exemple :

```
!DOCTYPE html>
```

```
<html>
```

```
  <head>
```

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

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

```
    <title> Ma page personnelle </title>
```

```
  </head>
```

```
  <body>
```

```
    <div class = "entete">  Entête du site </div>
```

```
    <div class = "menu"> contient les liens vers les autres pages  </div>
```

```
    <div class = "corps">
```

```
      ... corps de ma page ...
```

```
    </div>
```

```
    <div class = "footer"> Contient les éléments de bas de pages </div>
```

```
</body>
```

```
</html>
```