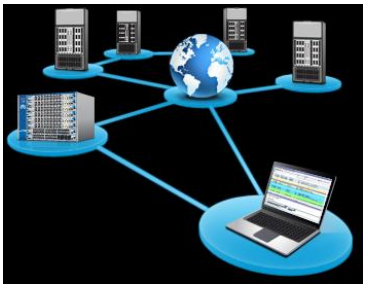
Un premier pas dans les langages HTML et CSS

Sommaire

I. Quelques termes importants

1.1 Internet



Internet désigne le réseau informatique mondial physique accessible au public, composé des réseaux publics, privés, universitaires, commerciaux, ...

L'accès à internet peut être obtenu par des FAI (fournisseurs d'accès internet) via ADSL, fibre, câble, satellite, 3G+, etc.

1.2 www - world wide web

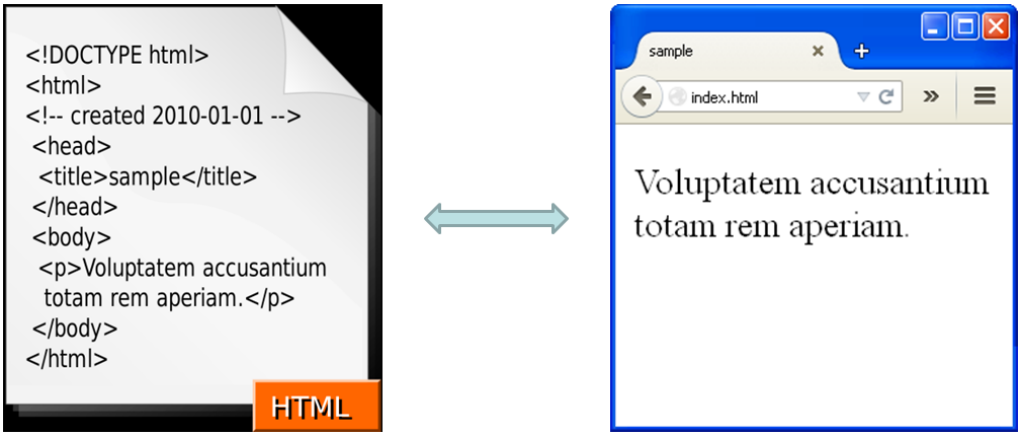
Le WWW ou "le web", qui signifie "la toile d'araignée mondiale", désigne l'ensemble des informations contenues sur les machines du réseau et la manière de communiquer entre elles.

Le web est une application du réseau internet parmi d'autres (mail, messagerie instantanée, partage de fichiers P2P, streaming vidéo, etc.).

1.3 HTML- HyperText Markup Language

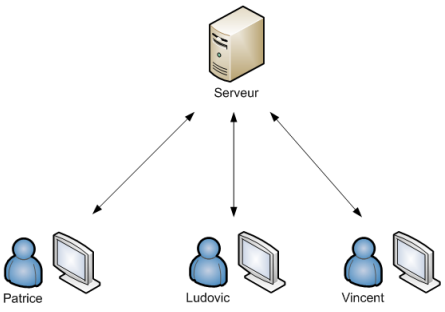
Le HTML (HyperText Markup Language) est un langage de balisage de "fichiers hypertextes", aujourd'hui appelés pages web.

Il s'agit d'un langage utilisé pour **décrire** des pages web, ce n'est donc pas un langage de programmation.



On visualise les fichiers HTML avec des navigateurs web comme :

1.4 Principe de fonctionnement du Web : Client/Serveur



Un site web classique fonctionne sur un système **Cient-Serveur** (voir https://fr.wikipedia.org/wiki/Client%E2%80%93serveur).

Le **client** est un **navigateur** (mozilla, chrome, edge, internet explorer,...).

Le navigateur est un **programme** de votre ordinateur personnel qui permet **d'interroger un serveur par un lien ou URL**.

Un lien identifie de façon unique une ressource distante sur le réseau web (une application ou page web hébergée sur un serveur web).

Lorsque qu'un **lien ou URL** est saisi dans le navigateur ou lorsque qu'un lien est cliqué à partir d'une page, le **navigateur interroge le réseau** par une **requête**.

Le serveur interrogé construit une réponse et l'envoie au client (vers le navigateur de votre ordinateur). Le message échangé entre le client et le serveur transite sur le réseau en utilisant un **protocole HTTP** ou **HTTPs** ('s' pour sécurisé).

La page web renvoyée peut être **interprété par le navigateur** qui reçoit la réponse. Le navigateur affiche la page qui contient:

- des balises HTML pour structurer l'information,
- du CSS pour la mise en page et le style,
- du JavaScript pour exécuter des petits traitements "coté client" (à l'opposé des traitements exécutés "coté serveur" lors de la préparation de la page).

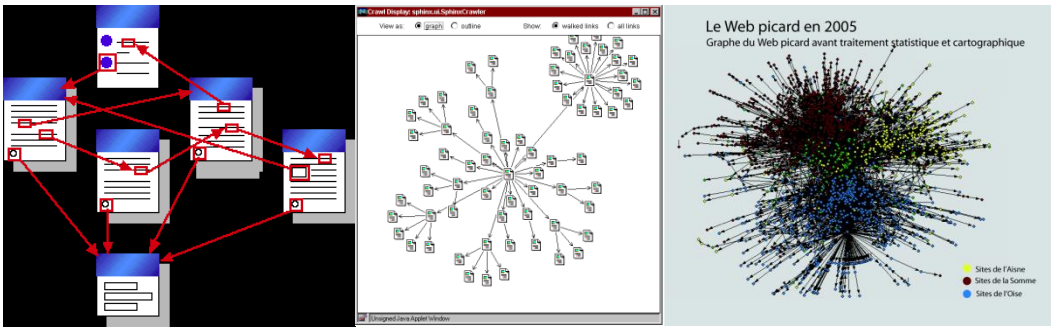
II. Présentation du HTML

2.1 Pourquoi apprendre la HTML ?

- Le HTML est un langage de représentation **universel** et **unique.**
- Le HTML est un langage de représentation **transparent**. Pour vous en convaincre, choisissez une page HTML et tapez « Ctrl + U » sur votre clavier…

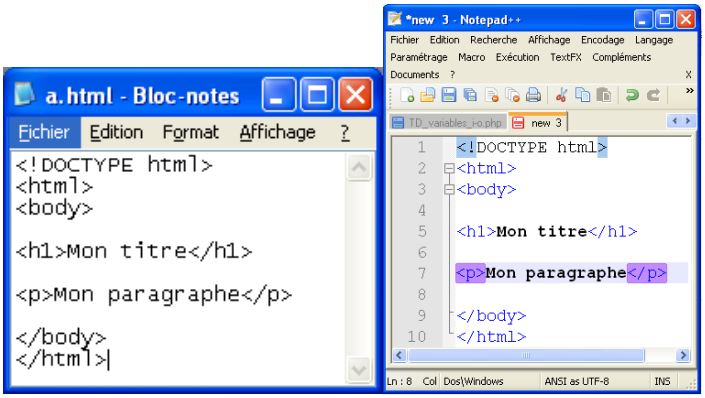
2.2 Documents hypertextes

- Les pages HTML sont du texte "enrichi".
- Les liens hypertexte permettent de passer d'une page à une autre.



2.3 Une page HTML vue par un codeur et un utilisateur

Un codeur



Avec un éditeur de texte (Notepad++, Vim, Emacs, Eclipse, etc.)

Un utilisateur

Avec un navigateur (Firefox, Chrome, etc.)

Certains éléments, comme <html>, <body>, <b> etc... ne sont pas affiché sur la page web. Ces éléments donnent des indications au navigateur sur la façon d'afficher le texte.

III. Programmer en HTML

3.1 Le HTML : un langage à balises

Pour composer une page web, on écrit donc du texte, et on y insère des caractères spéciaux qui permettront au navigateur (par exemple Firefox ou Internet Explorer), de mettre en forme ce texte. Ces caractères spéciaux se nomment des balises.
Un **document HTML** est donc composé de **texte** et de **balises**.

Exemple :

```
<html>
  <head>
    <title>Le titre de la page</title>
  </head>
  <body>
    <h1>Mon premier titre</h1>
    <p>Mon premier <b>paragraphe</b></p>
  </body>
</html>
```

Dans cet exemple, les balises sont :
<html>, <head>, <title>, </title>, </head>, <body>, <h1>, </h1>, , , , , </body>, </html>

Bilan :

- Une **balise HTML** est un élément de texte (un nom) encadré par le caractère inférieur (<) et le caractère supérieur (>).
  - par exemple, <h1> (titre 1)
- Une balise ouverte doit (presque) toujours être refermée.
  - par exemple, <h1> du texte </h1>
- Il existe de nombreuses balises pour mettre en forme et enrichir le texte.
  - par exemple, <h1>, <h2>, <i><u><img><video> , etc.
- Le texte écrit entre la balise ouvrante et la balise fermante est le contenu de la balise.
  - Dans l'exemple suivant : Mon premier paragraphe, le terme « paragraphe » est le contenu de la balise .

Exemple de contenu de balise :

```
<html>
  <head>
    <title>Le titre de la page</title>
  </head>
  <body>
    <h1>Mon premier titre</h1>
    <p>Mon premier <b>paragraphe</b></p>
  </body>
</html>
```

Dans l'exemple ci-contre, le contenu de certaines balises sont les suivants :

- contenu de la balise <head> : <title>Le titre de la page</title>
- contenu de la balise <h1> : Mon premier titre</h1>
- contenu de la balise <body> : <h1>Mon premier titre</h1> Mon premier paragraphe

Les balises agissent sur leur contenu.

Exercice : trouver le contenu de la balise <body>.

3.2 Structure de base d'un document HTML

La structure de base de tout document HTML est la suivante :

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

L'ensemble d'un document HTML est contenu entre les balises <html> et </html>.
Entre les balises <head> et </head> se trouvent des informations supplémentaires, qui n'apparaissent pas directement sur la page, comme par exemple les styles utilisés sur la page, l'encodage des caractères, le nom donné à la page, ou encore le code des fonctions (JavaScript) qui seront utilisées sur la page.
Les informations qui apparaîtront directement sur la page sont écrites entre les balises <body> et </body>.

Les principales balises :

<title>
La balise <title>, inscrite dans la partie <head> correspond en fait au texte qui sera affiché comme titre de l'onglet. Le contenu de cette balise ne correspond pas à un titre qui apparaîtrait directement sur la page.

<h1>
La balise <h1> signifie heading 1. Cette balise permet de faire des titres, à l'intérieur de la page.
Les caractères entre <h1> et </h1> sont donc affichés avec une taille de police plus grande que les autres caractères de la page. Il est également possible de définir des sous-titres, des titres de section, etc... en utilisant les balises suivantes : <h1>, <h2>, <h3>, <h4>, <h5>, <h6>. <h1> sera utilisée pour les titres principaux, <h2> pour les sous-titres, <h3> pour les titres de section, etc...

<p>
La balise <p> permet de définir les paragraphes.

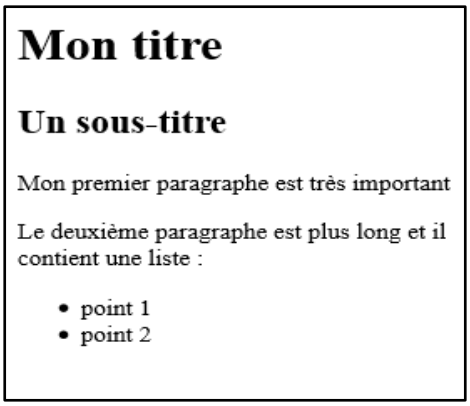
**<b>, <i> et **
La balise <b> (abréviation pour bold) permet de mettre du texte en gras. Pour mettre du texte en italique, on utilise <i> (pour italique) ou la balise (pour "emphasis", mettre en gras).

3.3 Structuration des pages

Le HTML permet de structurer les pages pour mieux faire ressortir les titres, les sous-titres, les paragraphes, les listes, les citations, etc.

Exemple :

```
<h1> Mon titre </h1>
<h2> Un sous-titre </h2>
<p> Mon premier paragraphe est très
important </p>
<p> Le deuxième paragraphe est plus
long et il contient une liste :
<ul>
<li> point 1</li>
<li> point 2</li>
</ul>
</p>
```



Code HTML

Résultat sur le navigateur

3.4 Mise en forme du texte – Imbriquer les balises

Des balises de mise en forme du texte telles que <i>, <b>, <u> permettent de mettre une partie du texte en **gras**, *italique*, souligné,...

Exemple : `Une partie en <b> gras </b> et une en <i> italique </i>`

Résultat : Une partie en **gras** et une en *italique*

Exemple : `Une partie en <u><i> italique </i> et tout est souligné </u>`

Résultat : Une partie en *italique* et tout est souligné

3.5 Attributs des balises

Exemple : Pour expliquer le rôle d'un attribut d'une balise, prenons l'exemple suivant :

```
Magnifique hôtel
<hr width= align='left'/>
Vue sur le lac
```

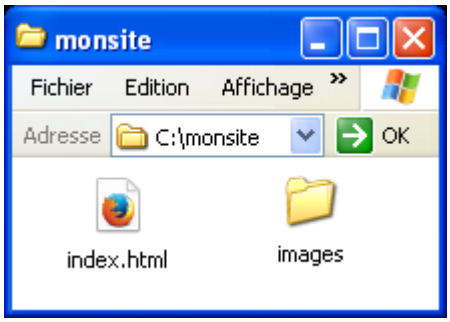
La balise **<hr/>** permet d'afficher des **lignes horizontales**. Par défaut, la ligne sera affichée sur toute la largeur de la page. L'attribut width='150px' permet de définir la largeur de la ligne à 150 pixels.

Caractéristiques d'un attribut :

- Un attribut a un **nom** (*width*) **suivi du signe « = »** et d'une **valeur** (*'150px'*).
- Les attributs d'une balise s'insèrent **après le nom** de la balise et **avant le chevron de fermeture** (>).
- Les attributs ne se placent que dans la **balise d'ouverture**, jamais dans celle de fermeture.
- Quand il y a plusieurs attributs, ils sont placés **les uns à la suite des autres**, dans un **ordre quelconque** et **séparés par des espaces**.
- La **valeur** de l'attribut est entourée par des **guillemets** (width="150px") ou des **apostrophes** (width='150px').

3.6 Structuration des fichiers

Lorsque l'on construit un site, il est important de structurer le stockage des fichiers. Le fichier de lancement du site s'intitule *index.html*. Il convient de le placer dans un dossier portant le nom de votre site (par exemple *C:\monsite*). Pour lancer le site dans le navigateur, il suffit de double-cliquer sur le fichier *index.html*.



3.7 Insertion d'images

Toutes les images associées aux pages de votre site doivent être placées dans un dossier nommé par exemple « *images* ». Il existe alors plusieurs manières d'insérer de l'image dans le texte :

- En spécifiant un **chemin réduit** qui, par défaut, partira du dossier courant (C:\monsite) :

```
<img src='images/pacman.jpg' alt='pacman' />
```

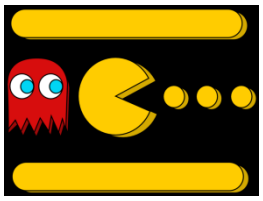
- En spécifiant **l'adresse internet (URL)** de l'image :

```
<img src='http://deviantart.com/art/pacman.jpg' alt='pacman' />
```

- En spécifiant le **chemin complet** à partir de la racine (plus sûr !!):

```
<img src='C:/monsite/images/pacman.jpg' alt='pacman' />
```

Le résultat de ces trois méthodes est identique et conduit à l'affichage de l'image de « pacman » :



Remarque : dans la balise <*img/*>, l'attribut *src* (source) prend comme valeur le chemin d'accès au fichier image (*pacman.jpg*).

3.8 Insertion de vidéos

Depuis la version 5 du HTML, il est possible d'intégrer facilement une ou plusieurs vidéos. Le code est le suivant :

Code HTML

```
<video width="200" height="300" controls="controls"
autoplay="true">
  <source src="minestorm.mp4"type="video/mp4" />
  <source src="minestorm.webm" type="video/webm" />
  <source src="minestorm.ogg" type="video/ogg" />
  Ecrire une alternative à la vidéo
</video>
```

Toutes les vidéos appelées seront lancées les unes à la suite des autres…

3.9 Insertion de liens : sans liens, pas de web !!

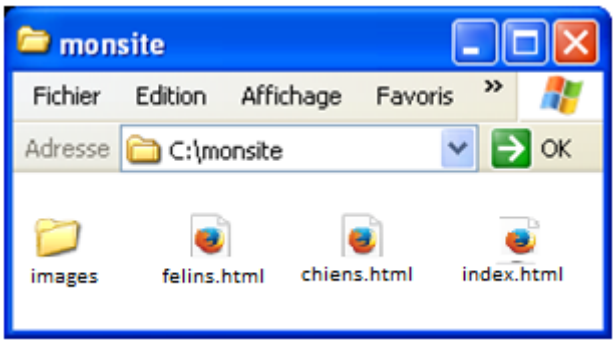
A l'intérieur du texte, vous avez la possibilité d'insérer des « liens hypertextes » vers de nouvelles pages ou des sites internet. A partir de la page principale accessible par le fichier *index.html*, considérons un lien qui amène vers la page web suivante, écrite dans le fichier *felins.html* enregistré dans le dossier local (voir image ci-dessous):

Page locale

Page locale

Page web

Dans cette page, un lien amène vers une autre page sur les chiens en cliquant sur « notre site sur les canidés ». Cette nouvelle page est par exemple écrite dans le fichier *chiens.html* et est enregistré aussi dans le dossier local :



Pour accéder à la page « le site des amis des chiens », taper le code suivant :

```
... visitez <a href='chiens.html'> notre site sur les canidés</a> ...
```

Pour revenir à la page « le site des amis des félins », en cliquant sur l'image «  », le code (tapé dans la page *chiens.html*) est le suivant :

```
... préférer les <a href='felins.html'><imgsrc='images/tete_chat.png' /></a>
```

Enfin, il est aussi possible d'insérer des liens permettant d'accéder à une page web sur internet. Dans la page *felins.html*, le code est le suivant :

```
...<a href='http://www.30millionsdamis.fr'> 30millionsdamis.fr</a> ...
```

3.10 Balises sans contenu

La plus part des balises ont un contenu (par exemple une <*b*>*belle*</*b*> *ville*, le contenu de <*b*> étant *belle*).

D'autres balises n'ont pas de contenu (par exemple <*hr*>, ou encore <*img src='... '*>). Pour ces balises sans contenu, plutôt que d'écrire la balise ouvrante et la balise fermante :

<*hr*></*hr*>),

on fusionne ces deux balises, pour n'écrire qu'une balise :

<*hr*/>, ou ****

dans laquelle le slash se situe à la fin de la balise.

3.11 Balises

La balise n'a aucun effet ! Par exemple, écrire :

Un peu de texte

ou

Un peu de texte

produit exactement le même résultat.

Cependant, une telle balise est utile pour délimiter une portion du texte, à savoir le texte présent entre la balise ouvrante et la balise fermante . Dans l'exemple ci-dessus, il s'agit des termes « *peu de* ».

Nous verrons dans la section suivante (CSS) qu'il est parfois utile de pouvoir délimiter une certaine portion du texte, soit pour la mettre en forme, soit pour la modifier.

3.12 Architecture générale d'une page web

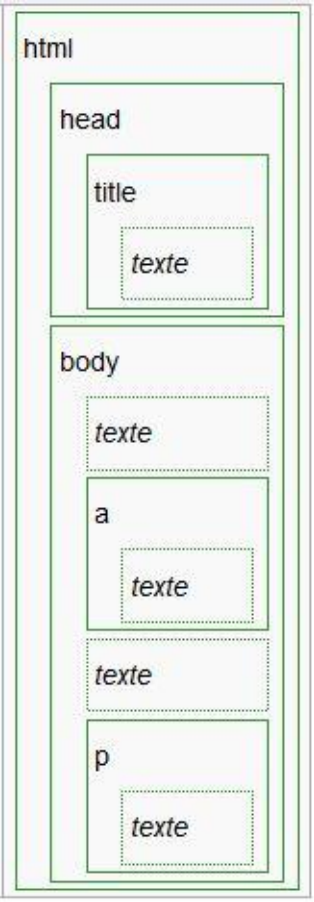
L'architecture d'une page HTLM est la suivante :

Code HTML

```
<!DOCTYPE html>
<html>  <!-- Marque le début du document HTML -->
        <head><!-- Marque le début de la zone d'en-tête -->
        <metacharset="utf-8" />
        <title>Titre de la page</title><!-- Titre du document -->
        </head><!-- Marque la fin de la zone d'en-tête -->

        <body><!-- Marque le début du corps de la page -->
        ...
        </body><!-- Marque la fin du corps de la page -->
</html> <!-- Marque la fin du document HTML -->
```

Résultat sur le navigateur

```
<!DOCTYPE html>
<html>
 <head>
  <title>
   Exemple de HTML
  </title>
 </head>
 <body>
  Ceci est une phrase avec un <a href="cible.html">hyperlien</a>.
  <p>
   Ceci est un paragraphe où il n'y a pas d'hyperlien.
  </p>
 </body>
</html>
```



Remarques importantes :

- L'essentiel du site se trouvera entre les balises **body**.
- Il est important d'**indenter** pour mieux visualiser son code.
- En cas d'oubli, le navigateur affiche quand même ce qu'il peut…

3.13 Règles de codage à respecter

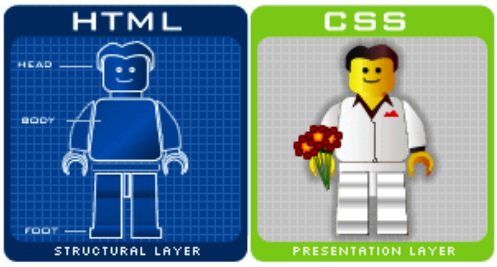
Avec les navigateurs actuels (Internet explorer, Firefox, Chrome, etc...), même lorsque le code HTML contient certaines erreurs, la page s'affichera malgré tout correctement. Cependant, certains supports n'ont pas les ressources nécessaires pour interpréter du code HTML contenant des erreurs. Il s'agit par exemple des pages destinées à être vues sur un téléphone portable.
Voici donc certaines règles qu'il convient de respecter, même si les navigateurs actuels afficheront correctement le résultat :

- **l'imbrication des balises doit être correct** :
  Une belle ville, et pas *Une belle ville*
- **les balises doivent toujours être fermées** : ** et pas *<img src='monImg.jpg'>*
- **les noms des balises et leurs attributs s'écrivent en minuscule** : ** et pas *<IMG SRC='monImg.jpg'/>*
- **les valeurs des attributs sont toujours encadrées par des guillemets ou des apostrophes** : *img src='monImg.jpg'* et pas *img src=monImg.jpg*

IV. La CSS (Cascading Style Sheets) habille le HTML

4.1 A quoi sert la CSS

Les **feuilles de style en cascade**, généralement appelées **CSS** de l'anglais **Cascading Style Sheets**, forment un langage informatique qui décrit la présentation des documents HTML.



L'idée principale de la CSS est de séparer le contenu de la page et sa mise en forme. La CSS va permettre entre autre :

- La mise en forme du texte (police, taille du texte, style, couleur ...)
- Le positionnement, la mise en page des différents éléments de la page.

4.2 La syntaxe de la CSS : un exemple

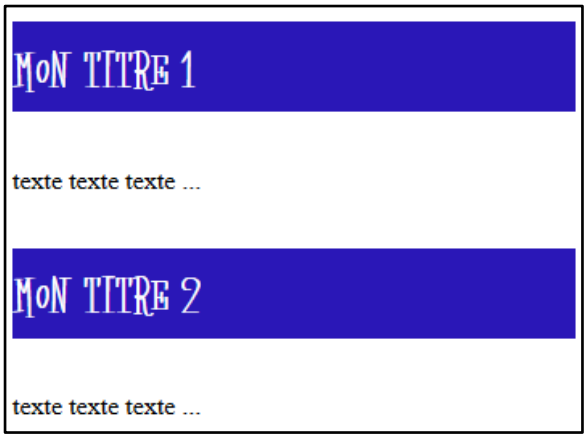
Supposons que vous souhaitiez afficher tous les titres de votre page en blanc, sous un fond bleu, avec une police de caractère de 20 pixels et représenté sous la font « darkmoon ». Le code HTML et la CSS associé seront les suivants :

Code HTML

```
<h1> Mon titre 1</h1>
<p> texte texte texte ... </p>
<h1> Mon titre 2 </h1>
<p> texte texte texte ...</p>
```

Code CSS

```
h1{
  background-color : #2A17B7;
  color : white;
  font-size : 20px;
  font-family : darkmoon;
}
```



Ecriture générale :
La syntaxe de base de CSS est composée de 3 parties :

- un **sélecteur**
- une **propriété**
- une **valeur**

`sélecteur {propriété:valeur}`

Un **sélecteur** correspond à une **balise HTML** (<p>, <h1>, etc...) et la **propriété** est un **attribut** dont on veut changer la valeur.

Exemple : Pour `h1 {font-size : 20px}`, tous les titres principaux (`sélecteur` : <h1>) du document auront une taille de 20 pixels (attribut `font-size` : 20px).

Propriétés :

- Si la valeur d'un attribut contient un espace, alors la valeur de l'attribut s'écrit en guillemets :

```
h2 {font-family : "sans serif"}
```

- Il est possible de définir plusieurs attributs pour un même sélecteur. Dans ce cas, chaque propriété sera séparée par un point-virgule :

```
p {font-family:"sans serif"; font-size: 90%; color: blue}
```

En écrivant une propriété par ligne, la lisibilité est meilleure :

```
p
{
  font-family:"sans serif";
  font-size: 90%;
  color: blue;
}
```

4.3 Grouper les sélecteurs

Si certaines propriétés s'appliquent à plusieurs sélecteurs, il est possible de les grouper. Ainsi, plutôt que d'écrire :

```
h1
{
 font-family:"sans serif";
 color: blue;
}
h2
{
 font-family:"sans serif";
 color: blue;
}
```

Il est possible, et plus rapide, de grouper les sélecteurs h1 et h2 et d'écrire :

```
h1,h2
{
 font-family:"sans serif";
 color: blue;
}
```

4.4 Les différentes manières de placer le code CSS

a) Dans les balises HTML avec l'attribut ***style*** : **une méthode lourde à éviter !!**

Code HTML

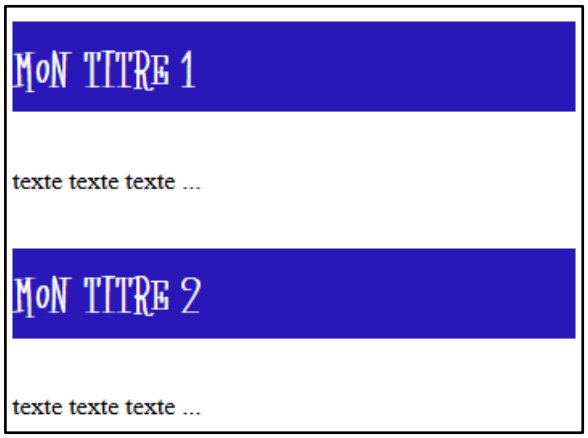
```
<h1
   style = "background-color : #2A17B7;
        color : white;
        font-size : 20px;
        font-family : darkmoon;">
   Mon titre 1
</h1>
<p> texte texte texte ... </p>
<h1> Mon titre 2 </h1>
<p> texte texte texte ... </p>
```

Dans ce cas, seule la première balise est concernée.

b) Dans le « HEAD » en précisant les balises concernées: **une méthode à éviter aussi !!**

Code HTML

```
<html>
 <head>
  <style>
   h1{
         background-color : #2A17B7;
         color : white;
         font-size : 3em;
         font-family : darkmoon;
      }
  </style>
 </head>
<body>
  <h1> Mon titre 1</h1>
  <p> texte texte texte ...</p>
  <h1> Mon titre 2 </h1>
  <p> texte texte texte ...</p>
 </body>
</html>
```

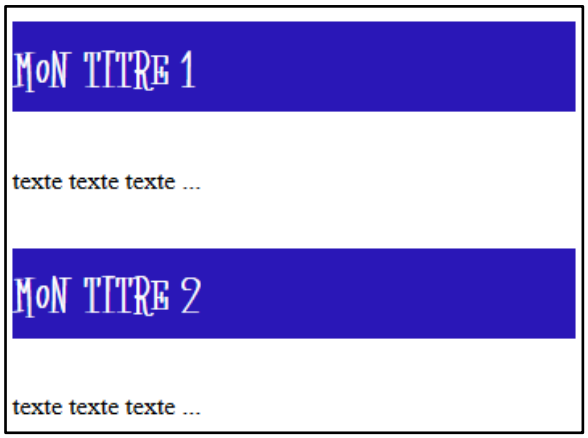


Dans ce cas, toutes les balises <h1> du document sont concernées.

c) Dans un fichier (par exemple : *monstyle.CSS*) séparé : **LA méthode à utiliser autant que possible !!**

Code HTML

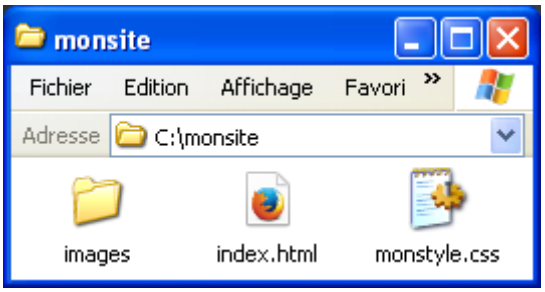
```
<html>
 <head>
  <linkrel="stylesheet" ref="monstyle.css">
 </head>
 <body>
  <h1> Mon titre 1 </h1>
  <p> texte texte texte ... </p>
  <h1> Mon titre 2 </h1>
  <p> texte texte texte ...</p>
 </body>
</html>
```



Code CSS (dans le fichier *monstyle.CSS*)

```
body{
 background : white;
}
h1{
 background-color : #2A17B7;
 color : white;
 font-size : 3em;
 font-family : darkmoon;
}
```

Dans ce cas, vous devez enregistrer le fichier.css dans votre dossier courant :



4.5 Formater certains éléments avec l'attribut « *class* »

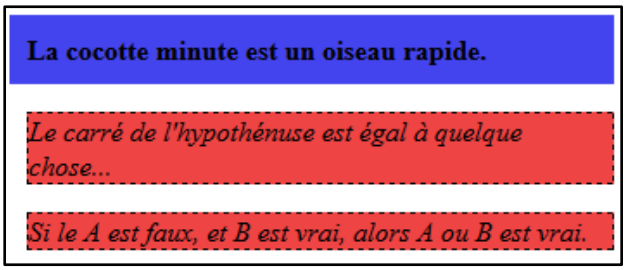
Il est possible de formater plusieurs éléments de même type avec la même mise en forme en utilisant l'attribut « *class* ». Par exemple sur un site de mathématique, vous souhaitez une mise en forme différente pour les définitions et les théorèmes. Voici la manière d'utiliser la mise en forme spécifique en utilisant l'attribut « *class* » :

Code HTML

```
<p class='definition'>
La cocotte minute est un
oiseau rapide.
</p>

<p class='theorem'>
Le carré de l'hypothénuse
est égal à quelque chose...
</p>

<p class='theorem'>
Si le A est faux, et B est
vrai,
alors A ou B est vrai.
</p>
```

Code CSS

```
p.theorem{
background-color : #EE4444;
font-style: italic;
border-style:dashed;
border-width:1px;
margin-left : 10px;
}

p.definition{
background-color : #4444EE;
font-weight: bold;
padding : 10px;
}
```



4.6 Formater un élément précis avec l'attribut « *id* »

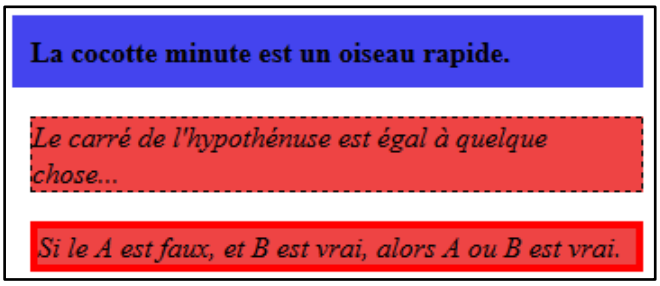
Il est possible de formater un élément précis avec une mise singulière en utilisant l'attribut « *id* ». Par exemple sur le site de mathématique précédent, vous souhaitez une mise en forme différente pour les définitions et les théorèmes, et vous souhaitez qu'un des deux théorèmes soit mis en valeur par un encadrement rouge. Voici la manière d'utiliser la mise en forme spécifique en utilisant l'attribut « *id* » :

Code HTML

```
<p class='def'>
La cocotte minute est un
oiseau rapide.
</p>

<p class='theorem'>
Le carré de l'hypothénuse
est égal à quelque chose...
</p>

<p id='th_logique_1' 'class='theorem'>
Si le A est faux, et B est vrai,
alors A ou B est vrai.
</p>
```

Code CSS

```
p.theorem{
background-color : #EE4444;
font-style: italic;
border-style:dashed;
border-width:1px;
margin-left : 10px;
}

p#th_logique_1{
border-style:solid;
border-width:4px;
border-color:red;
}
```



4.7 Feuille de style et visualisation en fonction du type d'appareil utilisé

Comme nous l'avons vu précédemment, les styles sont définis dans un ou plusieurs fichiers séparés, stockés dans un répertoire nomme par exemple « css ». Ces fichiers se terminent avec l'extension .css : *style.css*.

Ce fichier doit être inclus dans le fichier html de la manière suivante entre les balises `<head></head>` :

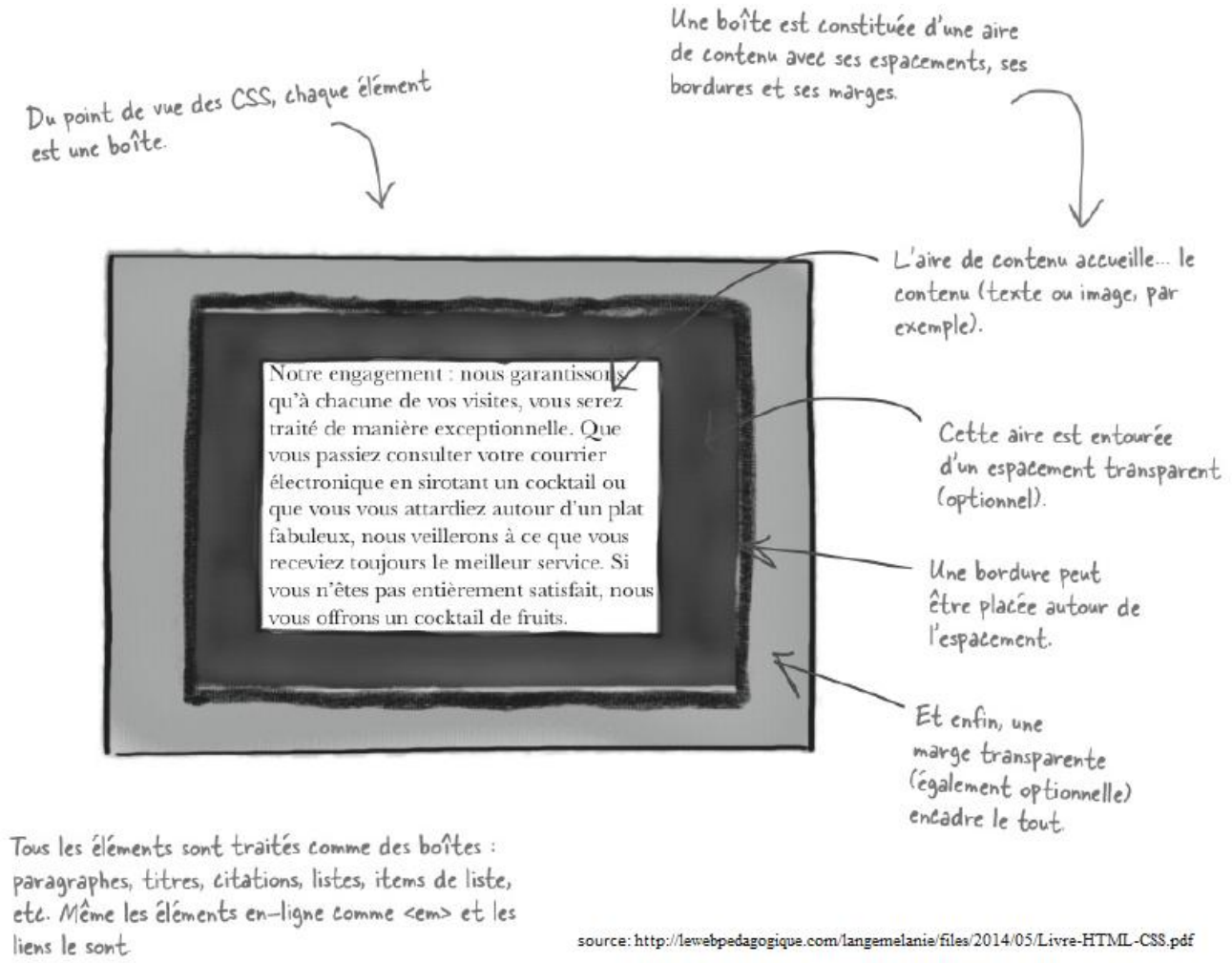
```
<link href="css/style.css" rel="stylesheet" type="text/css" media="screen" />
```

Dans cet exemple, l'attribut media de définir différentes feuilles de style suivant le type d'appareil utilisé :

- `screen : pour un écran`
- `print : pour une imprimante`
- `hanheld : pour un téléphone portable`
- etc.

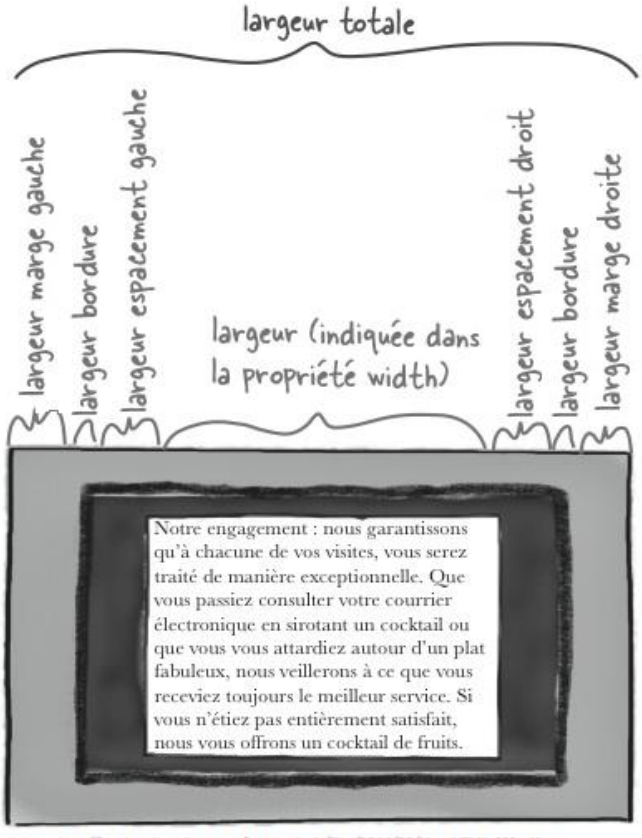
4.8 Balise div, marges et espacement

Une balise `<div>`, qui signifie « division du document », est un conteneur dans lequel des éléments sont placés pour les maintenir ensemble. Cela permet, par exemple, d'appliquer un même style à un groupe d'éléments ou encore de placer ces différents éléments ensembles sur une page, en utilisant les attributs `class` ou `id`.



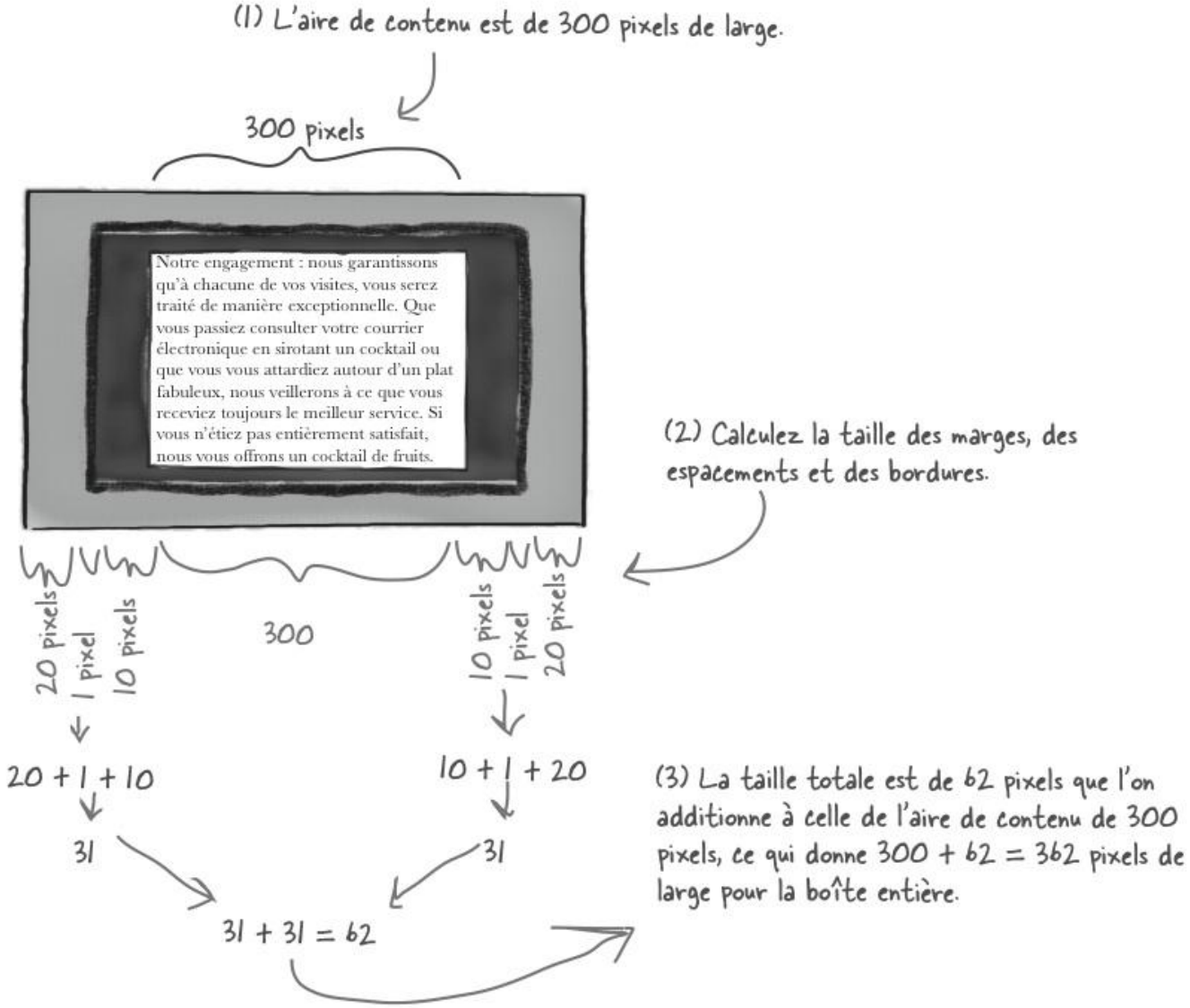
source: http://lewebpedagogique.com/langemelanie/files/2014/05/Livre-HTML-CSS.pdf

Pour ajouter de **l'espacement**, on utilise la propriété `padding`. Pour ajouter une marge, on utilise la propriété `margin`. Pour avoir une idée de la largeur de toute la boîte, il faut additionner la largeur de l'aire de contenu à celles des marges de gauche et de droite plus celle de la bordure qu'il faudra compter deux fois puisqu'il y en a une à gauche et une autre à droite.



source : http://lewebpedagogique.com/langemelanie/files/2014/05/Livre-HTML-CSS.pdf

Voici un exemple :



source : http://lewebpedagogique.com/langemelanie/files/2014/05/Livre-HTML-CSS.pdf

4.9 Interligne

La propriété `line-height` permet de déterminer la taille de l'espace vertical situé entre deux lignes de texte. Comme pour les autres propriétés qui s'appliquent aux polices, il est possible de l'exprimer en pixels, en `em` (taille de police de caractère) ou en pourcentage, ces deux derniers étant relatifs à la taille de la police.

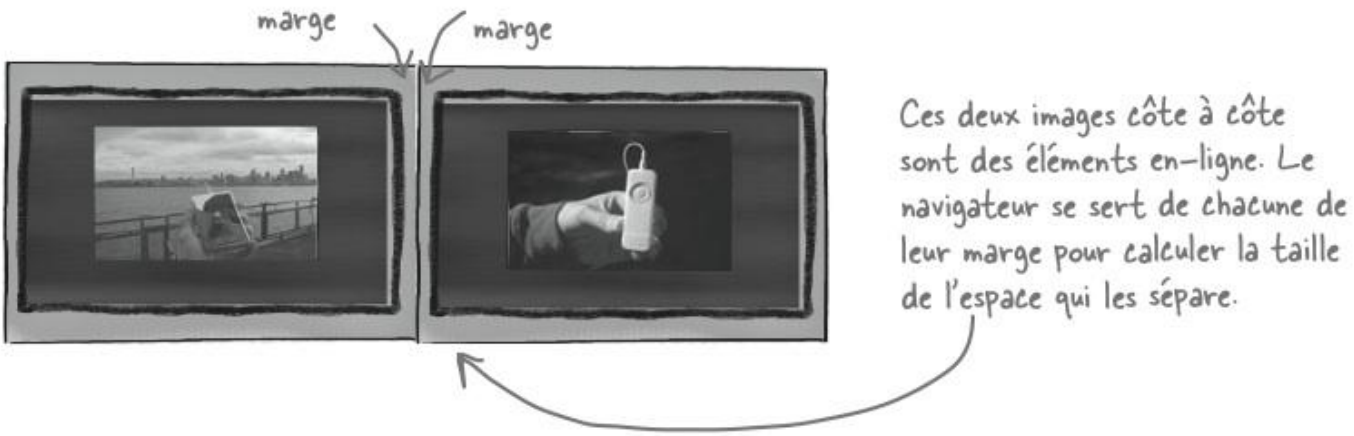
4.10 Le flux : placement des éléments sur une page

Le navigateur utilise le flux pour effectuer la mise en pages des éléments HTML. Pour représenter le positionnement en flux normal, on peut imaginer le navigateur parcourant (logiquement) la page de code HTML du début à la fin et retranscrivant son contenu au fur et à mesure des balises rencontrées. Comme dans la lecture d'un texte, il place verticalement les éléments dans la page, commençant en **haut de l'écran pour aller jusqu'en bas**, et horizontalement **de gauche à droite**, sur la totalité de l'espace disponible et nécessaire en largeur comme en hauteur.

Pour les éléments de bloc, il place un saut de ligne entre chaque. Le premier élément d'un fichier est le premier à s'afficher, un saut de ligne lui succède, puis le second élément s'affiche, suivi d'un saut de ligne, et ainsi de suite de haut en bas du fichier.

Les éléments *en-ligne* se placent les uns à côté des autres depuis le coin en haut à gauche jusqu'à celui en bas à droite. Quand le navigateur doit placer côte à côte des éléments *en-ligne* ayant chacun une marge, il met suffisamment d'espace entre les deux pour que les deux marges soient prises en compte. Si la marge de l'élément de gauche mesure 10 pixels et celle de l'élément de droite mesure 20 pixels, l'espace entre les deux éléments mesurera bien 30 pixels.

Exemple de deux éléments *en-ligne* :



source : http://lewebpedagogique.com/langemelanie/files/2014/05/Livre-HTML-CSS.pdf

Exemple de deux éléments placés verticalement :

source : http://lewebpedagogique.com/langemelanie/files/2014/05/Livre-HTML-CSS.pdf

4.11 Eléments flottants dans la page

Il est possible d'insérer des blocs flottants pour lesquels l'ordre d'empilement est différent de celui du flux normal. Les blocs flottants sont disposés entre les blocs non positionnés et les blocs positionnés dans l'ordre suivant:

1. L'arrière-plan et les bordures de l'élément racine du document ;
2. Les blocs qui descendent les uns à la suite des autres et qui sont situés dans le flux normal, dans l'ordre dans lequel ils apparaissent dans le code HTML ;
3. Les blocs flottants ;
4. Les éléments enfants positionnés, dans leur ordre d'apparition dans le code HTML ;

La procédure à suivre pour faire flotter un élément est la suivante :

```
# id_flot {
width: 200px;
float: right;
}
```

1. Donner un identifiant unique (`id="id_flot"`) à l'élément que vous voulez faire flotter
2. Donner une largeur à cet élément, par exemple 200px
3. Ajouter la propriété float.

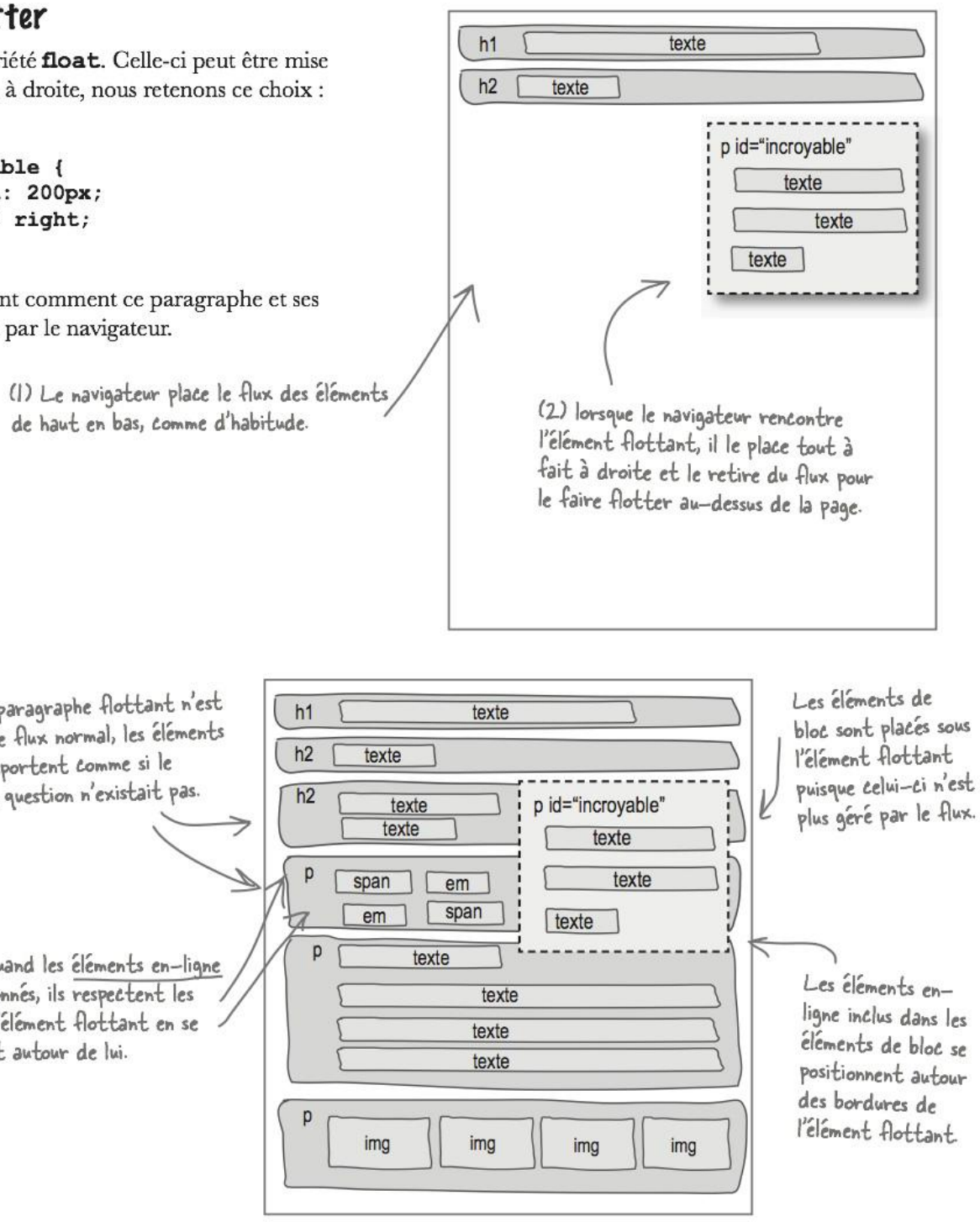
Le résultat est le suivant :

Faites-le flotter

Ajoutons la propriété **float**. Celle-ci peut être mise à gauche ou bien à droite, nous retenons ce choix :

```
#incroyable {
   width: 200px;
   float: right;
}
```

Voyons maintenant comment ce paragraphe et ses voisins sont gérés par le navigateur.



source : http://lewebpedagogique.com/langemelanie/files/2014/05/Livre-HTML-CSS.pdf

4.12 Vérifier la validité du code

Le code HTML doit être conforme et valide. Il est donc important de valider le code html avec un validateur en ligne :

- https://validator.w3.org/#validate_by_upload

V. Quelques liens utiles

5.1 Tutoriels HTML/CSS

- https://www.w3schools.com/
- https://developer.mozilla.org/fr/docs/Web/HTML
- Openclassroom - Première page web
- fr.html.net - Apprendre html/css/php
- Pompage.net - CSS

5.2 Liens utiles

- HTML Color Codes
- Font-Squirel - Des fonts libres de droit
- Dafont - Des fonts pas forcément libres

5.3 Ressources libres

- OpenWebDesign - Structures libres de droit
- Arcsin Web Templates - CSS libres de droit