



HTML CSS

COURS THEORIQUE

Antoine Lucsko

Octobre 2011

Généralités

Le HTML signifie HyperText Markup Langage. C'est un langage de balisage, qui permet de créer de l'hypertexte et de structurer sémantiquement le contenu d'une page web.

Le web trouve son origine dans les années 1989, lorsque Tim Berners-Lee rédige un document hypertexte au CERN. L'année suivante on pouvait compter dans le monde entier 130 sites web...

À partir de 1990, Tim Berners-Lee développe les trois principales technologies du web : les adresses web, le Hypertext Transfer Protocol (HTTP) et le Hypertext Markup Language (HTML) . Puis dans la période de 1991 à 1994 un nombre considérable de sites Internet sont créés.

Le W3C (World Wide Web consortium), naît sous l'impulsion de Tim Berners-Lee en 1994. Il a pour objectif de donner de la cohérence au projet de standardisation des langages pour le web comme le HTML, XML, CSS, PNG, SVG (graphique vectoriel adaptable),...

Les standards du web ont mis du temps à s'imposer et, on a pu assister durant toutes ces années à ce que l'on appelle « la guerre des navigateurs ». Aujourd'hui on peut dire que les standards définis par le W3C sont mieux intégrés par les navigateurs.

En France le collectif Openweb, où l'on peut trouver Tristan Nitot, actuellement président de Mozilla-Europe, se charge de promouvoir les normes du W3C.
(Tristan Nitot sur le sujet du eG8 : <http://vimeo.com/24178794>)

L'évolution des normes

Le XHTML est né en 1998, c'est le langage que l'on utilise actuellement. Il a des aspects assez complexes, notamment il se veut une reformulation du XML pour donner le XHTML.. On peut cependant lui accorder un point fondamental : c'est la grande rigueur qu'il impose dans son écriture.

A partir de 2004, un tournant s'est opéré, et un nouveau groupe de travail « dissident », le **WHATWG** (Web Hypertext Application Technology Working Group) se forme parallèlement au W3C. Il travaille sur une nouvelle version du HTML.

Le W3C intègre en 2007 officiellement le **WHAT** afin de créer le HTML 5. Les travaux sur la norme du XHTML 2 sont alors abandonnés.

La moralité de cette histoire, c'est que le XHTML demeure tout à fait standard et aura pour successeur le HTML 5.

Les navigateurs

Les navigateurs web ont évolué depuis le début du web.

Microsoft s'implique dans les recommandations du W3C depuis 3 ans. Il sort en 2006 IE7 , puis IE8 en 2009, et aujourd'hui : IE9. Il faut se rappeler que IE6 s'est imposé pendant plus de 6 ans, sans réelle évolution...

Depuis 2005, on observe cependant une baisse de l'utilisation d'IE au profit des navigateurs dits alternatifs comme Mozilla Firefox. Fin 2008 Google sort son propre navigateur : Chrome, basé sur le même moteur de rendu que Safari . Ces derniers navigateurs sont très respectueux des normes du W3C.

On peut tester le bon respect des recommandations du W3C de son navigateur préféré à l'aide du test [Acid3](#) (ce test est élaboré par le WaSP, groupe de concepteurs de site Internet professionnels qui respectent et promeuvent les normes du W3C).

Il existe quatre groupes de navigateurs web :

- IE qui est basé sur le moteur de rendu **Trident**
- Mozilla Firefox,..basé sur **Gecko**
- Opéra,.. basé sur **Presto**
- Chrome, Safari,.. basé sur **Webkit**

Remarques :

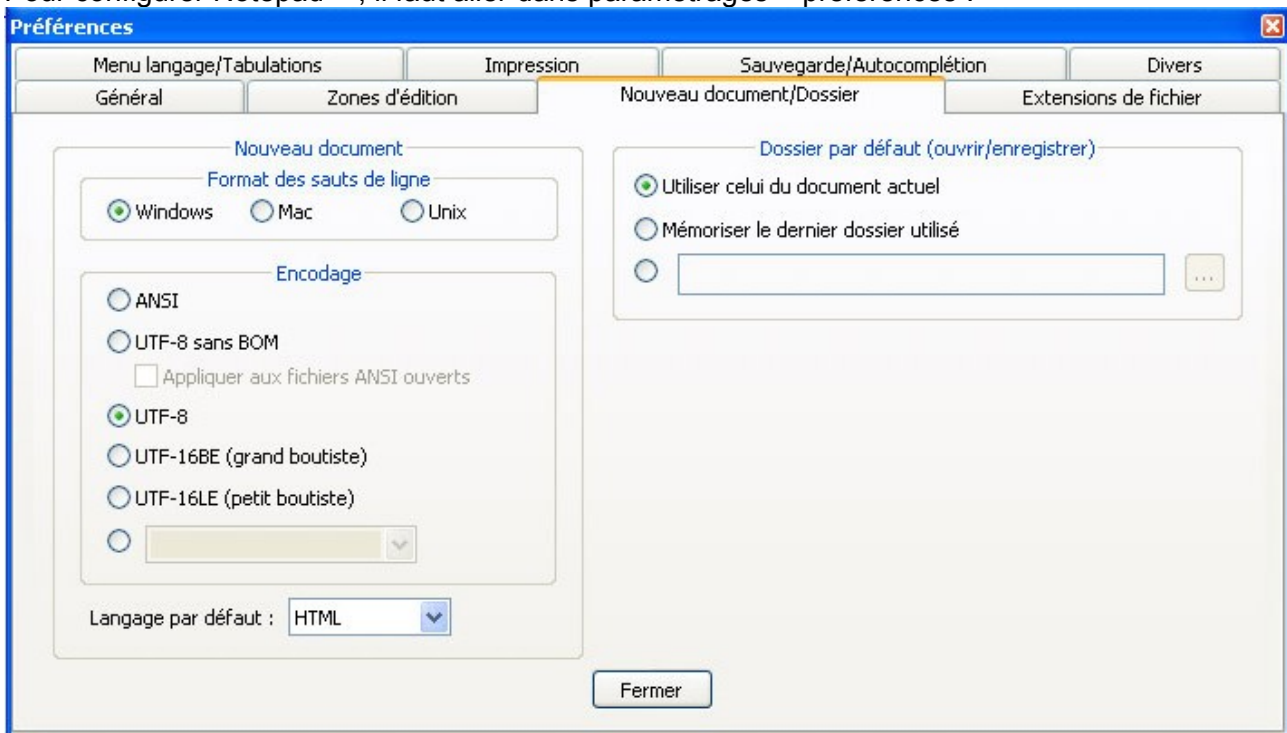
Aujourd'hui IE6 n'est utilisé que par 5% des internautes. C'est un navigateur comportant un grand nombre de bugs, que l'on ne considérera pas dans ce cours.

Organisation

Pour l'écriture du code on utilisera [Notepad++](#)

Les documents seront en UTF-8 normal, il faut paramétrer Notepad++ pour que tous les nouveaux documents ouverts soient en UTF-8. Rappelons que l'UTF-8 est un format de codage des caractères.

Pour configurer Notepad++, il faut aller dans paramétrages > préférences :



Il faut ensuite créer un dossier html sur le bureau, dans lequel on créera les sous-dossiers correspondant aux différents chapitres du cours.

Nommage des fichiers

Pour qu'une page web et un fichier css soient reconnus par les navigateurs, il faut mettre les bonnes extensions aux fichiers HTML et CSS. *Attention au nommage des fichiers, le nom des fichiers est en alphanumérique, sans espace, sans accent et ne commence pas par un chiffre.*

Pour créer un fichier HTML, il faut indiquer l'extension **.html**

Pour les fichiers CSS on indiquera l'extension **.css**

→ **exemple**

index.html pour un fichier HTML et global.css pour un fichier CSS

Introduction

Le cours porte sur le (X)HTML 1.0 strict. Ce langage de balise sert à structurer les pages web en parties ou blocs, ayant un sens précis : paragraphe, listes, tableaux, etc.

Les CSS (Cascading Style Sheet ou feuille de style en cascade) décrivent la présentation, la forme et l'habillage du document HTML.

Ces deux langages s'utilisent ensemble pour constituer des pages web graphiques.

Initiation au HTML

Structure d'une page HTML

Un document HTML est constitué :

- d'un **DTD** (voir définition ci-dessous),
- d'un **<head>**, en-tête du document HTML,
- d'un **<body>** corps du document ;

Ces deux derniers étant contenus dans l'élément HTML **<html>**, racine du document.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "
http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">

<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>Titre de la page</title>
</head>

<body>
</body>

</html>
```

Définition DOCTYPE ou DTD

C'est une contraction de Document Type Declaration. Il est nécessaire pour le bon fonctionnement des pages : il indique au navigateur la grammaire que l'on utilise pour construire les pages. Il informe, également, le [validateur W3C](#) de la version du HTML utilisée pour les pages.

Si on a pas de DTD, les navigateurs se mettent en mode « Quirk », ce qui peut se traduire par « habitudes bizarres » en français ; dans ce cas les pages web ne s'affichent pas correctement.

Les 5 règles du XHTML 1.0 strict

- Chaque élément HTML comporte une balise ouvrante et fermante. Toute balise ouvrante doit obligatoirement être **fermée**. De même, les balises toute seule se ferme : ``, `
`, `<hr />`,...

- **Les balises et les attributs de balises s'écrivent en lettres minuscules.**
`Un nom de la page`
- Les valeurs d'attributs de balise sont dans des côtes simples ou doubles
`je suis une orange ou une banane`
- **Chaque attribut à une valeur**
`<input type= "radio " checked=" checked" />`
- **Les éléments doivent être correctement imbriqués.**
`<p>Je suis un paragraphe un peu long</p>`

Remarques

Pour le XHTML1.0 strict les attributs et éléments de présentation ne sont plus autorisés. Ils doivent être remplacés par des styles CSS ; l'attribut target des liens et les iframe ne sont pas autorisés (ni frameset et frame).

Encodage des caractères

Les caractères sont encodés en UTF-8 (unicode). Il faut préciser l'encodage des caractères dans le <head> et enregistrer le fichier HTML en UTF-8.

`<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />`

Et enregistrer le fichier dans le même encodage de caractères.

→ voir exemple 1

Entités HTML

Les entités HTML servent à afficher des caractères spéciaux, ou des caractères du langage HTML lui-même.

`<` ; < (lt : less than)

`>` ; > (gt : gretter than)

`…` ; ...

On trouvera une liste des entités HTML [ici](#)

→ voir exemple 2

Les types des éléments HTML

Il existe deux familles d'élément HTML, la famille de **rendu de type en ligne** et la famille de **rendu de type bloc**.

Définition des éléments de type de rendu blocs (ou de type bloc)

Ils distinguent des parties entières comme des paragraphes, titres, etc. Ils occupent/réservent tout l'espace de leur conteneur.

Leur propriété permet de les dimensionner. On peut également les positionner, dans le flux du document.

Définition des éléments de type de rendu en ligne (ou de type en ligne)

Ils sont prévus pour enrichir localement des portions de texte (liens hypertextes, emphase, etc). Ils se dimensionnent par leur contenu.

→ voir exemple 3

Contenus autorisés selon les types des éléments HTML

Un élément de type bloc peut contenir un ou plusieurs éléments de type bloc et/ou des éléments de type en ligne, à l'exception des éléments suivants :

`<p>`, `<h1>`, `<h2>`, ... En effet, ces derniers n'acceptent que des éléments de type en ligne.

Un élément de type en ligne ne peut contenir que des éléments de type en ligne.

Il existe deux sortes d'éléments de type en ligne : **remplacés et non remplacés**.

Les éléments de type remplacés acceptent des attributs de dimensions (`height` et `width`). Il s'agit des éléments : ``, `<input />`, `<textarea>`, `<select>` et `<object>`

Les autres éléments n'ont pas de dimension à proprement parler, ils n'occupent que l'espace nécessaire à leur contenu. C'est le cas de ``, ``, `<a>`.

On peut passer d'une structure de type en ligne en type bloc à l'aide de la propriété CSS `display`, que l'on verra plus loin.

Les principales balises HTML standards

type en ligne

`<a>` : lien hypertexte, elle s'accompagne de la balise `href` qui renferme la cible du lien et `title` qui renseigne la destination du lien.

Exemple : `Ouvrir la page`

`` mettre en emphase un texte

Exemple : `important`

`` inclut une image dans le document html. Elle s'accompagne des attributs `alt` (pour un texte alternatif) et `src` (pour indiquer le chemin vers l'image).

Exemple : ``

`<q>` : à utiliser pour les citations courtes.

Exemple `<q>Quand on est intelligent, il est plus facile de faire l'imbécile que l'inverse. </q>`

`` : conteneur en ligne dépourvu d'un sens précis. Il permet de regrouper d'autres éléments au fil du texte.

Exemple : `Un texte `

`` : indique un renforcement du texte.

Exemple : ` Je suis un texte en gras `

type bloc

`<blockquote>` : à employer pour une citation longue

Exemple : `<blockquote><p> Bien que tout vice verse dans le coeur humain le poison de l'adversaire, c'est l'envie qui permet au serpent de cracher son venin le plus secret et de vomir la peste de sa méchanceté pour la faire partager.</p>`

`<p>Grégoire Le Grand</p>`

`</blockquote>`

`<div>` : conteneur générique de type bloc. Elle est utilisée pour structurer le document HTML

Exemple : `<div id="banner"></div>`

<dl> : liste de définition. Permet de distinguer les titres des définitions **<dt>** des éléments de définition **<dd>**

Exemple :

```
<dl>
    <dt> Une définition précise</dt>
    <dd>Une description de la définition</dd>
</dl>
```

<form> : balise délimitant un formulaire. Elle s'accompagne de l'attribut **action** qui donne le chemin vers le fichier qui traitera le formulaire, de l'attribut **name** qui permet de donner un nom au formulaire et de l'attribut **method** qui permet de spécifier la manière dont on échange les données avec le serveur.

Exemple :

```
<form action="changer.php" name="form" method="post" >
    <h2><input type="text" value="" name="nom" size="15" /></h2>
    <p><input type="submit" value="modifier" name="modifier" />
</form>
```

<h1>,<h2>,<h3>,...<h6> : le HTML prévoit 6 niveaux de titres, hiérarchiquement placés sous un seul et unique titre principale **<h1>**

<h1>Le titre du site</h1>

<h2>Sous titre</h2>

, : désignent respectivement les listes ordonnées et les listes à puce. Elles comportent les éléments de liste ****

Exemple :

```
<ul>
    <li> item 1</li>
    <li> item 2</li>
</ul>
```

<p> : désigne un paragraphe

Exemple : <p> je suis un paragraphe</p>

<table> : tableau contenant des données ou résultats. Elle comporte les éléments **<tr>** pour les lignes du tableau et **<td>** pour les cellules du tableau.

Exemple :

```
<table >
    <tr>
        <td> Cellule </td><td> Cellule </td>
    </tr>
    <tr>
        <td> Cellule </td><td> Cellule </td>
    </tr>
</table>
```


Le modèle de boîte des éléments HTML

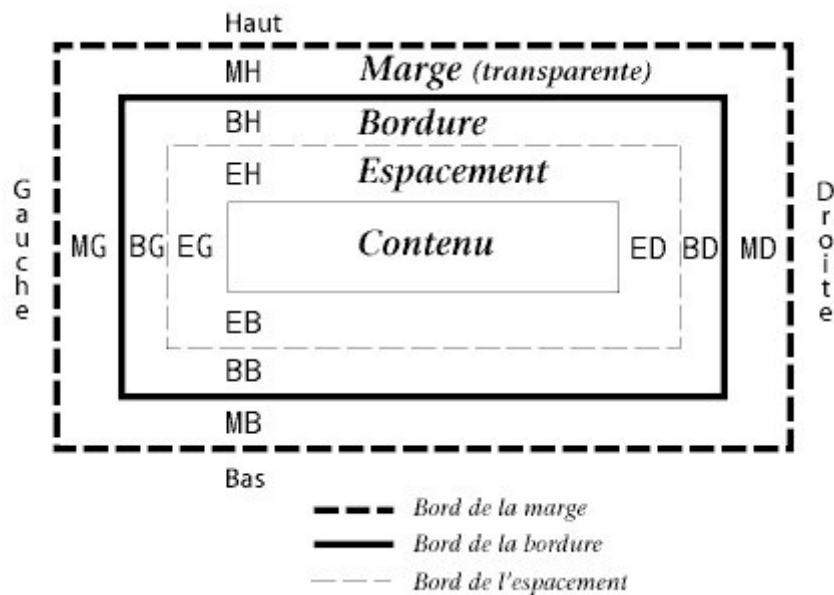
Définition du modèle de boîte

Selon la spécification W3C, tout élément structuré par une balise HTML se présente sous forme d'une boîte définie par :

- une largeur et une hauteur : height et width, c'est la zone de contenu
- une bordure : border
- une marge interne : padding, elle fait partie de l'arrière plan de la boîte.
- une marge externe : margin, elle se situe hors de la boîte.

Définition du calcul de la largeur d'une boîte

Pour calculer la largeur d'une boîte, il faut additionner les valeurs suivantes : largeur, marges interne gauche et droite et bordures gauche et droite. Les marges externes ne sont pas prise en compte dans ce calcul.



Le flux du document ou flux courant

L'ordre dans lequel apparaissent les éléments HTML est celui dans lequel les boîtes sont affichées et s'empilent (verticalement ou l'une à la suite de l'autre) dans le document. Ce mécanisme de positionnement est propre au flux du document.

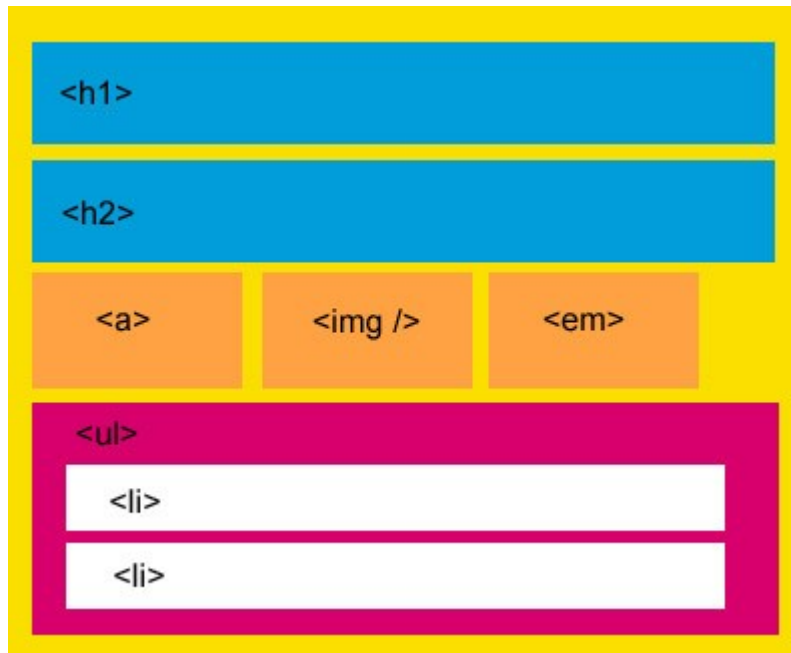
Définition du positionnement dans le flux

Un élément se place dans un conteneur le plus haut possible et le plus à gauche possible. Il est dépendant de son élément frère.

Si on a des éléments de type bloc, ils se placent l'un en dessous de l'autre,

Si on a des éléments de type en ligne ils se placent l'un à la suite de l'autre si ils ont la place et sinon, ils vont à la ligne en dessous.

→ Exemple de l'occupation des éléments HTML dans le flux courant.



Ancêtre, Parents, Enfants, Frères

Chaque élément HTML est imbriqué dans une hiérarchie de conteneur.

Définitions ancêtre

Un élément ancêtre est un élément qui contient un élément ou une hiérarchie d'éléments.

Définitions parent

Un élément qui contient directement un autre élément, ou le premier ancêtre dans la hiérarchie.

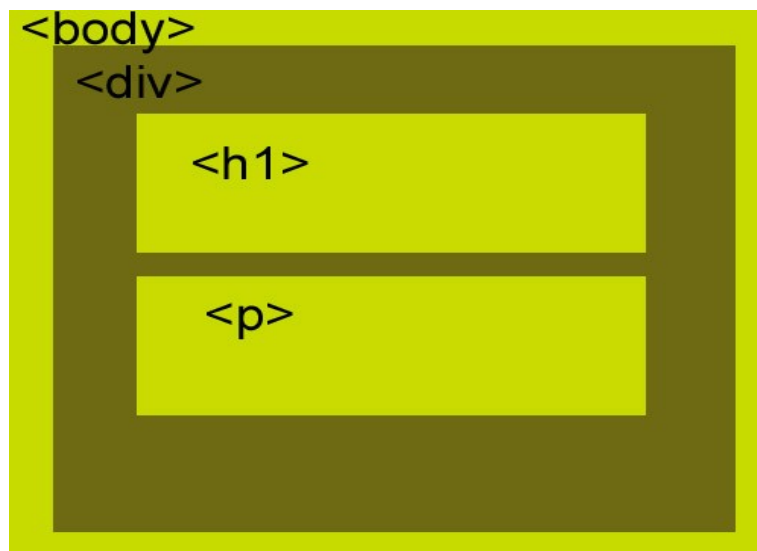
Définitions enfant

Un élément contenu directement dans un autre élément

Définitions frère

Un élément contenu avec d'autres éléments dans un même conteneur, possède des frères.

Ci-dessous, on a `<body>` est un ancêtre de `<div>`, `<h1>` et `<p>`. `<body>` est parent de `<div>`. `<h1>` et `<p>` sont frères. `<h1>` et `<p>` sont enfants de `<div>`.



Initiation au CSS

Définitions de base en CSS

Les CSS s'écrivent dans une syntaxe particulière. Elle commence par un sélecteur, suivi d'un bloc de déclaration qui comporte les propriétés, l'ensemble constituant une règle.

Les commentaires dans le code CSS s'écrivent entre les symboles suivants :

/ je suis un commentaire */*

Définition simple Cascading Style Sheet

La cascade est la résolution du problème des différentes sources de styles qui s'appliquent à un même document. Chaque origine des styles a une priorité bien définie : auteur, utilisateur et agent utilisateur (navigateur).

Les styles de l'agent utilisateur (navigateur) n'agiront que dans les domaines où rien n'est spécifié par l'auteur ou l'utilisateur.

On retiendra par ordre de priorité : **utilisateur (internaute) > auteur (notre feuille de style) > agent utilisateur (navigateur)**

Définition propriété

C'est une syntaxe définie explicitement par la spécification CSS. Elle définit la mise en forme de l'élément auquel elle s'applique. Syntaxe : « **clef : valeur** ».

Définition sélecteur

Il indique l'élément sur lequel vont s'appliquer les déclarations des propriétés CSS. On distingue plusieurs types de sélecteurs :

- le sélecteur d'élément comme **p, h1, div**, etc
- le sélecteur de classe : **.orange** (il commence nécessairement par un point)
- le sélecteur d'identifiant : **#container** (il commence nécessairement par un #)
- le sélecteurs d'attribut : **a[title]**

Plusieurs sélecteurs peuvent précéder un bloc de déclaration.

Exemple :

```
p, h1, .orange, #container{
    font-size:15px;
    margin:0 15px;
}
```

Résumé

Règle

sélecteur	propriété	valeur
#container{	width:750px;	
	margin:0 auto;	
	} déclaration	
}		

- Mise en place des styles

Insertion des styles dans le <head>

On écrira les styles entre les balises <style>

<head>

...

<style type="text/css">

...

</style>

</head>

L'inconvénient de cette méthode est sa portée limitée au document html dans lequel on l'écrit.

Lier les styles à partir d'une feuille de style séparée

On écrira dans le <head>

<head>

...

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

...

</head>

Dans ce cas, il faut créer un fichier avec l'extension .css. L'intérêt de la méthode est que l'on factorise les CSS pour l'ensemble d'un projet en utilisant la même feuille de style pour autant de fichiers html que l'on veut.

L'attribut media désigne le type de média souhaité : screen pour l'écran du PC, print pour imprimer la page web, etc.

Classes et identifiants

Définition classe

Une classe est un nom que l'on choisit librement (caractères alphanumériques, sans accent ni espace et commençant par une lettre). C'est un sélecteur qui s'applique sur les éléments HTML possédant cette classe. Convention : dans le code une classe commence par un point

Exemple :

```
.orange{
    color:#FF6141 ;
}
<span class="orange">Je suis un texte</span>
```

Définition identifiant

C'est un sélecteur unique qui ne s'applique qu'à un seul élément HTML

Exemple :

```
#banner{
    background-color:#CCC ;
}
<div id="banner"> Je suis la bannière du site </div>
```

Positionnement absolu

C'est une propriété CSS très bien interprétée par l'ensemble des navigateurs. Pour positionner un élément en absolu, il faut mettre la propriété **position:absolute** ;

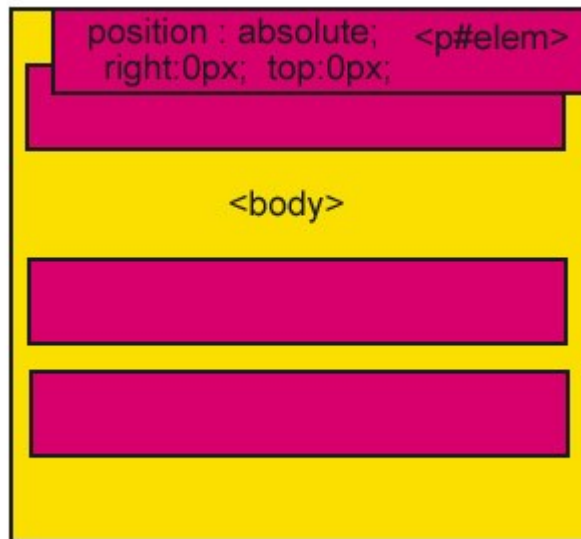
```
#col-left{
    position:absolute ;
}
```

Définition position absolu

Lorsqu'un élément est positionné en absolu il sort du flux courant et se positionne par rapport à son premier ancêtre positionné.

Pour positionner un élément il faut également lui fixer des valeurs sur ses axes des x et des y. On utilise les valeurs top, left, right ou bottom pour positionner l'élément.

Exemple :



l'élément HTML p#elem est positionné par rapport au body

Remarque : Un élément positionné en absolu, quel que soit son rendu, possède alors des caractéristiques particulières : boîtes dimensionnables (width, height,...), boîte qui occupe exactement l'espace de son contenu, pas de fusion des marges.

Définition position relative

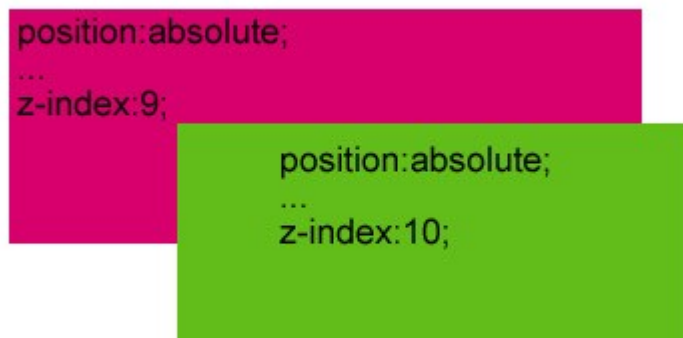
Un élément positionné en relatif, ne sort pas du flux courant. Il possède par contre la propriété « positionné » et peut alors servir de repère pour d'autres éléments contenus et positionnés en absolu.



Les éléments positionnés en absolu, contenu dans le `<div#container>` positionné en relatif, se positionnent par rapport à ce dernier.

Définition z-index

Les éléments positionnés se placent en fonction d'un repère (x,y,z) où z est la profondeur. La propriété z-index a pour valeur un nombre entier qui représente l'ordre d'empilement. Plus cet entier est grand plus la couche (profondeur) sera haute.



Plus l'index est grand, pour un élément positionné en absolu, plus l'élément HTML sera au dessus

Faire un site sur deux colonnes en absolu

On peut utiliser la position absolu pour structurer un site sur deux colonnes. On donne ici, à titre d'exemple, un code qui permet de faire un site fluide sur deux colonnes.

(voir : exemple 4)

Inclure des images dans un document HTML

**Inclure directement une image dans le document : **

Pour inclure une image dans un document HTML, on utilise l'élément HTML suivant :

Attention au chemin : il peut être relatif ou absolu. Dans l'exemple ci-dessus, il est relatif. Un chemin absolu donnerait : <http://www.monsite/images/poisson.jpg>, par exemple.

L'attribut alt est nécessaire dans les cas suivants :

- si l'image est cassée (non accessible), alors le texte alternatif s'affiche ;
- dans le cas de l'accessibilité : les malvoyants ne voient pas les sites avec des navigateurs classiques, ils ont des navigateurs qui affichent le contenu du site en mode texte, seul son texte alternatif apparaît. Il est donc important pour l'accessibilité de bien renseigner une image.

Notons également que l'attribut alt est un renseignement important dans le référencement des contenus par les moteurs de recherche, une image en elle-même n'étant pas référencée.

Pour les images, on utilisera, la propriété CSS vertical-align qui agit sur le positionnement vertical du texte. Les valeurs possibles de la propriété sont : top, left, right, bottom ou une valeur en px.

Par exemple, pour descendre une image tout en bas dans un conteneur on utilisera :

```
img{  
    vertical-align:bottom ;  
}
```

Conseil : on mettra une image dans le code à partir du moment où elle apporte une information qui complète un texte ou l'illustre.

Mettre une image en arrière plan

Dans le cas d'une image décorative on utilisera plutôt une image d'arrière plan.

```
#fond-banner {  
    background:url(..images/fond.jpg) no-repeat;  
}
```

Attention au chemin : il faut sans doute sortir du dossier css où théoriquement se trouve la feuille de style. Dans ce cas, on pensera à bien écrire le chemin relatif vers l'image. On doit sortir du dossier css et aller dans le dossier image, ce qui se traduit par "../" pour sortir du dossier.

Polices

La propriété ci-dessous indique quelle police on doit utiliser. Il faut mettre plusieurs choix, dans le cas où la première police ne serait pas disponible dans le système d'exploitation de l'internaute, on prendrait le choix 2 etc. :

font-family: "Choix 1", choix2, "Choix 3", générique;

Ci-dessous, la liste des polices que l'on pourra utiliser. Attention : les polices composées sont entre des côtes simples ou doubles :

Polices à empattements **sans-sérif**

Arial, Helvetica, "Trebuchet MS", Verdana, "Arial Black", sans-serif.

Polices à empattements **sérif**

"Times New Roman", Times, Georgia, serif.

Pour aller plus loin on pourra voir [cet article](#) sur alsacréation.

Unités

On utilisera les unités relatives suivantes px ou em « cadratin » (1 em = taille M dans la police de référence). Attention : la propriété font-size qui donne une taille pour le texte s'hérite. Sa valeur est donc transmise à tous les éléments descendants de l'élément considéré.

La particularité des em est d'être une taille de référence. Ainsi si un élément possède une taille de texte de 2em, et qu'un descendant possède une taille de texte de 3em, alors par rapport à son ancêtre il aura une taille $2 \times 3 = 6$ em.

Idem pour les %. Ce mécanisme n'existe pas avec les pixels (px). On peut également utiliser les % pour dimensionner les blocs. Par contre, on préférera définir en pixel ou em la taille du texte.

Les flottants

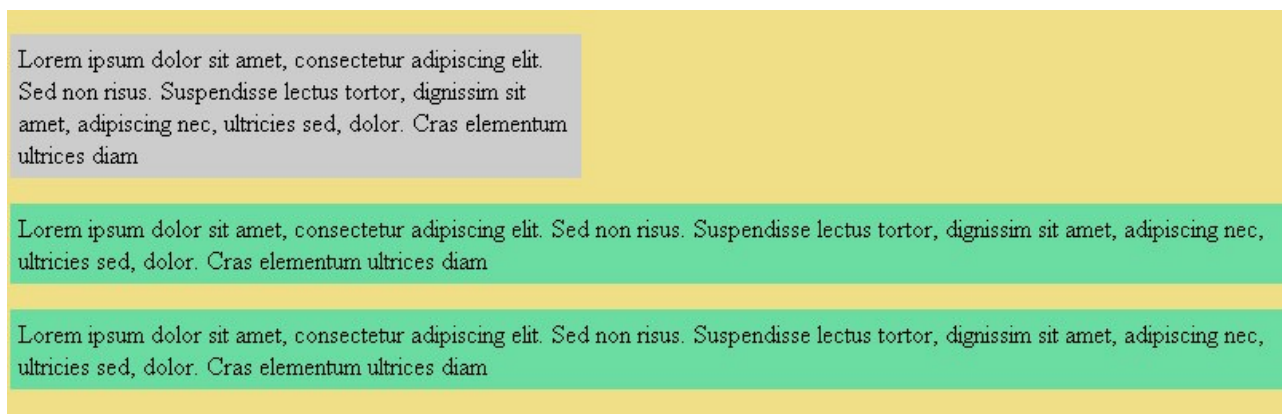
Initialement prévue pour les images ou les textes, on utilise cette propriété, pour positionner les éléments HTML en général. Elle est assez délicate à mettre en place, et nécessite une bonne compréhension de sa définition pour l'utiliser correctement.

Définition : flottant

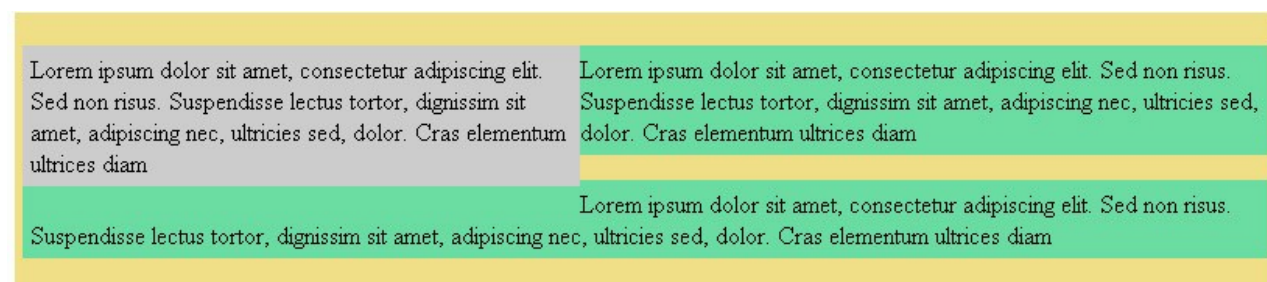
Un élément flottant est ôté du flux et placé à l'extrême gauche ou droite (float:left ; ou float:right) de son conteneur, tout en demeurant sur sa hauteur de ligne initiale dans le flux.

Remarque :

Attention : un élément flottant ne sort que partiellement du flux. Les éléments précédant le bloc flottant ne sont pas affectés par celui-ci. Les éléments qui suivent, quant à eux, sont affectés et se réorganisent dans le flux en conséquence. En effet, ils n'interagissent plus avec lui, sauf leurs contenus textuels ou images qui « coulent » autour de la boîte flottante.



paragraphes dans le flux, sans propriété de positionnement.



Le premier paragraphe est positionné en flottant. Il se place dans le flux au même niveau que les autres paragraphes, et les textes (contenu textuel) coulent autour du bloc flottant.

Contexte de formatage

On peut positionner deux blocs côté à côté en flottant, cependant, ils n'auront plus d'influence sur les éléments qui suivent. De même, les conteneurs de ces flottants n'occuperont plus l'espace dimensionné par les contenus des blocs flottants.

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed non risus. Suspendisse lectus tortor, dignissim sit amet, adipiscing nec, ultricies sed, dolor. Cras elementum ultrices diam

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed non risus. Suspendisse lectus tortor, dignissim sit amet, adipiscing nec, ultricies sed, dolor. Cras elementum ultrices diam

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed non risus. Suspendisse lectus tortor, dignissim sit amet, adipiscing nec, ultricies sed, dolor. Cras elementum ultrices diam

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed non risus. Suspendisse lectus tortor, dignissim sit amet, adipiscing nec, ultricies sed, dolor. Cras elementum ultrices diam

Deux blocs flottants dans un conteneur (div#container).

Dans l'exemple ci-dessus, il y a dépassement des flottants. Le conteneur ne calcule pas sa hauteur en fonction des hauteurs des flottants, car ils ne sont plus dans le flux.

On peut utiliser pour contourner ce « problème » **un contexte de formatage**. Il permet d'étendre le calcul de hauteur du conteneur à ses enfants flottants. Ainsi, une solution consiste à attribuer au conteneur la propriété « overflow:hidden; » ou « overflow:auto; ». Dans tous les cas, il est préférable de ne pas mettre de marge latérale au div#container, les résultats pouvant être d'un navigateur à l'autre différents.

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed non risus. Suspendisse lectus tortor, dignissim sit amet, adipiscing nec, ultricies sed, dolor. Cras elementum ultrices diam

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed non risus. Suspendisse lectus tortor, dignissim sit amet, adipiscing nec, ultricies sed, dolor. Cras elementum ultrices diam

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed non risus. Suspendisse lectus tortor, dignissim sit amet, adipiscing nec, ultricies sed, dolor. Cras elementum ultrices diam

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed non risus. Suspendisse lectus tortor, dignissim sit amet, adipiscing nec, ultricies sed, dolor. Cras elementum ultrices diam

Deux flottants, dans un div#container avec un contexte de formatage : overflow:hidden ;

On peut également créer un contexte de formatage en utilisant la propriété « clear ».

Elle interdit à un élément de se placer sur la même ligne qu'un bloc flottant et force celui-ci à se placer en dessous.

Si on a deux blocs dans le flux d'un élément, la premier bloc étant flottant et le deuxième sans propriété de positionnement et sans contexte de formatage on a la situation suivante :

bloc flottant

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed non risus. Suspendisse lectus tortor, dignissim sit amet, adipiscing nec, ultricies sed, dolor. Cras elementum ultrices diam

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed non risus. Suspendisse lectus tortor, dignissim sit amet, adipiscing nec, ultricies sed, dolor. Cras elementum ultrices diam

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed non risus. Suspendisse lectus tortor, dignissim sit amet, adipiscing nec, ultricies sed, dolor. Cras elementum ultrices diam

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed non risus. Suspendisse lectus tortor, dignissim sit amet, adipiscing nec, ultricies sed, dolor. Cras elementum ultrices diam

On introduira, entre le bloc jaune et vert, un saut de ligne avec la règle suivante :

```
hr.clear{
    clear:both ;
    visibility:hidden ;
}
```

```
<p class="jaune >...</p>
<hr class="clear" />
<p class="vert >...</p>
```

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed non risus. Suspendisse lectus tortor, dignissim sit amet, adipiscing nec, ultricies sed, dolor. Cras elementum ultrices diam

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed non risus. Suspendisse lectus tortor, dignissim sit amet, adipiscing nec, ultricies sed, dolor. Cras elementum ultrices diam

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed non risus. Suspendisse lectus tortor, dignissim sit amet, adipiscing nec, ultricies sed, dolor. Cras elementum ultrices diam

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed non risus. Suspendisse lectus tortor, dignissim sit amet, adipiscing nec, ultricies sed, dolor. Cras elementum ultrices diam

La propriété clear impose aux éléments qui suivent de se placer à la suite des flottants dans le flux. Remarquons, au passage que cette technique crée un contexte de formatage, car le bloc vert n'a pas de propriété de positionnement et « étire » donc la boîte conteneur.

Propriété display

Un élément HTML a un type de rendu déterminé. Pour modifier son rendu, on dispose de la propriété « display ». Par exemple, lorsque l'on veut étendre la surface réactive d'un lien hypertexte, qui est de type en ligne, on utilise la propriété « display:block ; » pour lui donner le type bloc. Ainsi, ce dernier se comportera comme un bloc et occupera tout l'espace de son conteneur.

```
#menu{
    width:250px;
    border:1px solid black;
    list-style-type:none;
}
a{
    display:block;
```

```
    text-decoration:none;
}
```

```
<ul id="menu">
    <li><a href="#" title="aller sur la page de bienvenue">bienvenue en Ardèche</a></li>
    <li><a href="#" title="voir le diaporama">voir le diaporama</a></li>
</ul>
```

Dans ce dernier exemple, les `` occupent 250px de largeur et les liens hypertextes, apriori n'occupent que l'espace du texte qu'ils enrichissent. Si ils sont en `display:block` ; , alors ils occuperont tout l'espace de leur conteneur, comme tout bloc, et feront 250px de large.

Organisation

Pour les outils d'analyse du code on utilisera [firebug](#)

Pour la largeur d'un site Internet on conseille 960px de large. En effet, cette largeur est facilement divisible, ce qui permet de faire de nombreux petits blocs dans la page.

Annexes

Propriétés CSS minimales pour les besoins du cours

Pour une liste complète des propriétés CSS on peut aller voir [ici](#)

Dans un premier temps on utilisera les propriétés CSS ci-dessous, et on utilisera l'unité px pour les valeurs

Propriétés de couleurs, pour les valeurs on peut écrire soit le nom de la couleur (black, blue, green, white, red...), soit en indiquant le code hexadécimal, soit en RGB

color :navy;

Propriétés de font, il faut mettre au moins 3 polices. Si l'utilisateur n'a pas la police 1 alors on utilisera la police 2, etc.

font-family : "Arial Black", Arial, Verdana, sans-serif ;

Indique la taille du texte

font-size : 12px ;

Graisse du moins fort au plus fort

font-weight:bold ;

Décoration du texte, valeurs possibles : underline : souligné, overline : ligne, au-blink : clignotant

text-decoration :underline ;

Texte en capital, uppercase : tout mettre en majuscules, lowercase : tout mettre en minuscules, capitalize : début des mots en majuscules

text-transform : uppercase ;

Hauteur de ligne, indique la hauteur de ligne en px

line-height:15px ;

Propriétés de font raccourci on a respectivement graisse, taille du texte, hauteur de ligne et police.

font:bold 14px/18px Arial, Helvetica, sans-serif;

Propriétés d'arrière plan, propre aux éléments de type de rendu bloc.

couleur de fond, la valeur peut s'exprimer en hexadécimal ou valeur prédéfini par les CSS

background-color :navy;

image de fond, le chemin de l'image peut être relatif ou absolu.

background-image: url("images/logo.png");

notation raccourci pour donner une couleur de fond une image, non répétée et en haut à gauche (top/bottom, right/left mais aussi 0% 100% pour positionner l'image)

background:#FFF url("images/logo.png") no-repeat top left;

largeur hauteur pour les boîtes

width, height, min-width, max-width, min-height, max-height

width:100px ;

width:25% ;

marges externes (margin) et internes (padding)

margin-top, margin-left, margin-right, margin-bottom

padding-top, padding-left, padding-right, padding-bottom

margin-top:15px ; padding-bottom:15px ;

marges externes et internes en **notation raccourci**, on commence par **top right bottom et left**.

margin:10px 5px 0 3px ;

padding:10px 5px 3px 2px ;

bordure en notation raccourci

border:1px solid black ;

position absolu s'utilise avec les propriétés top, left, right ou bottom et z-index pour indiquer un niveau d'empilement des éléments positionnés.

position:absolute ;

top:10px ;

left:15px ;

z-index:2;

position relative positionne un élément, l'élément ne sort pas du flux courant.

position:relative ;

propriétés de listes, la valeur none retire la puce à appliquer sur les

list-style-type:none ;

propriété qui permet de changer le rendu d'un élément, valeur : block, inline.

display:block ;

propriété qui permet d'aligner des éléments textuels dans une boîte (utiliser, souvent, pour les images), valeur : top, baseline, sub, super, text-top ou valeur en px, %

vertical-align:bottom ;