Mise en boîte (partie 1/2)

Informations sur le tutoriel

Ajouter à mes tutoriels favoris (11 fans)

Auteur : M@teo21
Licence :

Plus d'informations

Popularité

Visualisations : 200 707 353

Appréciation des lecteurs : 15
15
60
888

Publicité

Historique des mises à jour

- *Le 08/08/2010 à 03:23:12*
  Correction conjugaison.
- *Le 03/08/2010 à 01:07:48*
  Correction orthographique, #2605
- *Le 31/07/2010 à 18:20:55*
  Correction orthographique, ticket n°2593

Vous êtes en forme ? Tant mieux, c'est maintenant que tout se joue. Eh oui, jusqu'ici vous pouviez au mieux suivre mes petits exemples et inventer les vôtres, mais pas encore créer votre site web de toutes pièces.
Les 2 chapitres que vous allez lire vont être un véritable tournant dans ce cours : grâce à eux, vous saurez créer pour de bon votre site web, avec son design et tout et tout ! Et que ceux qui auraient peur de se retrouver tout seuls se rassurent : je ne vous abandonnerai pas. J'ai prévu de vous montrer toute la marche à suivre dans un TP

Le titre de ce chapitre est assez évocateur, nous allons nous préparer à "mettre en boîte" notre site, ce qui veut dire qu'il sera bientôt prêt !
Et, comme j'ai repéré parmi vous quelques étourdis qui n'écoutaient pas quand j'ai parlé de balises *inline* / *block*, je pense qu'un petit rappel ne ferait de mal à personne
Sommaire du chapitre :

- Block et Inline c'est pas pareil !
- La taille
- Les bordures
- Les marges
- *Q.C.M.*

Block et Inline c'est pas pareil !

Tout au long des chapitres précédents, vous avez entendu à plusieurs reprises les mots "inline" et "blocks". J'ai insisté sur le fait que c'était important à comprendre, car tout se jouerait là-dessus par la suite...
Eh ben devinez quoi, c'est maintenant que tout se joue sur cette différence. Si vous n'êtes pas capables de faire la différence entre une balise inline et une balise block, vous allez être facilement largués. Vous ne m'en voudrez pas, donc, d'avoir pris la précaution de vous faire ce rappel
Bon, de quoi parle-t-on ici ?

On parle des balises. On peut classer les balises XHTML dans une des 2 catégories suivantes : **inline et block** (en français : "En ligne" et "Bloc").
Ce que j'attends de vous, c'est que vous sachiez reconnaître si une balise est inline ou block.

Par exemple, ?
Oui, c'est une balise **block**.
Et <a> ?
Ca c'est une balise **inline**.
Mais comment je reconnais une balise inline d'une balise block ?

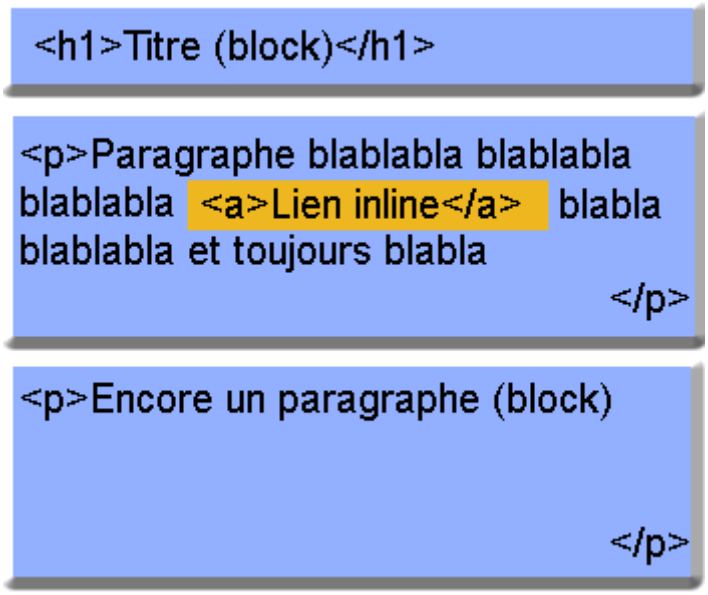
C'est en fait assez facile :

- block : une balise de type "block" sur votre page web crée automatiquement un retour à la ligne avant et après. Il suffit d'imaginer tout simplement un bloc. Votre page web sera en fait constituée d'une série de blocks à la suite les uns des autres. Mais vous verrez qu'en plus, il est possible de mettre un block à l'intérieur d'un autre, ce qui va augmenter considérablement nos possibilités pour créer le design de notre site !
- inline : une balise de type "inline" se trouve obligatoirement à l'intérieur d'une balise "block". Une balise inline ne crée pas de retour à la ligne, le texte qui se trouve à l'intérieur s'écrit donc à la suite du texte précédent, sur la même ligne (c'est pour cela que l'on parle de balise "en ligne").

Est-ce que crée un retour à la ligne ?
Oui, c'est donc une balise de type block.
Est-ce que <a> crée un retour à la ligne ?
Non, le texte contenu dans cette balise s'écrit à la suite du texte précédent, il reste sur la même ligne. C'est donc une balise de type inline.

Fastoche, isn't it ?

Pour les sceptiques, voici un petit schéma que je vous ai concocté (c'est donc moche à souhait, mais c'est pas le problème) C'est grâce à ce schéma que je visualise les choses dans ma tête, donc si vous voulez rentrer dans ma tête, regardez-le bien :



Sur fond bleu, vous avez tout ce qui est de type block.
Sur fond jaune, vous avez tout ce qui est de type inline.

Comme vous pouvez le voir, les blocks sont les uns en-dessous des autres. J'aurais pu imbriquer des blocks mais je ne veux pas trop compliquer les choses pour le moment
La balise inline <a> se trouve à l'intérieur d'une balise block (je vous avais dit que c'était obligatoire), et le texte vient s'insérer sur la même ligne.

Quelques exemples

Afin de mieux vous aider à assimiler quelles balises sont inline et quelles balises sont block, voici un petit tableau listant quelques balises courantes.

Balises blocks	Balises inline
<p>	
<blockquote>	
<h1>	<q>
<h2>	<a>
	<sup>
	<img />
...	...

Ce tableau n'est pas complet, loin de là. Si vous voulez avoir la liste complète des balises qui existent, et savoir si elles sont inline ou block, reportez-vous à l'annexe donnant la liste de toutes les balises XHTML.

Les balises universelles

Vous les connaissez déjà car je vous les ai présenté il y a quelques chapitres. Ce sont des balises qui n'ont aucun sens particulier (contrairement à <p> qui veut dire "paragraphe", "important" etc...).
Le principal intérêt de ces balises c'est de pouvoir appliquer une class (ou un id) pour le css quand aucune autre balise ne convient.

Il existe 2 balises génériques, et comme par hasard la seule différence entre les deux c'est que l'une d'elle est inline, l'autre est block :

- (*inline*) : on l'a déjà utilisée, si vous vous souvenez bien. Je m'en suis servi pour vous montrer quelques exemples de propriétés CSS au milieu d'une ligne. Vous souvenez-vous du code ci-

dessous ?

Code : HTML

```
<p>Comme l'a dit <span class="nom">Neil Armstrong</span> un certain 20
```

S'il existait une balise XHTML <nom></nom>, je l'aurais utilisée. Mais comme ce n'était pas le cas, j'ai dû me rabattre sur un et lui appliquer une class que j'ai inventée ("nom").

- (*block*) : c'est aussi une balise qui n'a aucun sens, comme span, sauf que celle-là est de type block. On va pas mal s'en servir dans les chapitres qui suivent, notamment pour créer le design de votre site. Vous allez voir qu'un design, en fait, c'est une série de blocks qu'on dispose comme on veut.

Respectez la sémantique !

Les balises universelles sont "pratiques" dans certains cas, certes, mais attention à ne pas en abuser. Je tiens à vous avertir de suite : beaucoup de webmasters mettent des et des trop souvent, et oublient que d'autres balises plus adaptées existent.
Voici 2 exemples :

- **Exemple d'un span inutile** :
  Je ne devrais jamais voir dans un de vos codes :

  ... alors qu'il existe la balise qui sert à ça !
- **Exemple d'un div inutile** :
  De la même manière, ceci est complètement absurde :

  ... puisqu'il existe des balises faites spécialement pour les titres (<h1><h2><h3>...)

Oui, vous allez me dire qu'au final le résultat (visuel) est le même. Je suis tout à fait d'accord.
Mais, tandis que les balises span et div ne signifient rien de particulier, les balises strong et h1 veulent dire quelque chose ! Strong signifie "important" et h1 signifie "titre principal".

En XHTML, on vous demande d'utiliser tant que possible des balises qui ont un sens : on appelle ça respecter la sémantique.
L'avantage c'est que si vous respectez ceci, votre code aura une certaine "logique". Lorsque le robot de Google viendra référencer votre site, il "comprendra" le sens des balises et cela pourra améliorer votre position dans le moteur de recherche 

La taille

Nous allons pour le moment travailler uniquement sur des balises de type "block".

Pour commencer, nous nous intéressons à la taille des blocks. Contrairement à un inline, un block a des dimensions précises. Il a une largeur et une hauteur.
Ce qui fait, Ô surprise, qu'on dispose de 2 propriétés CSS :

- width : c'est la largeur du block. A exprimer en pixels (px) ou en pourcentage (%).
- height : c'est la hauteur du block. Là encore, on l'exprime soit en pixels (px), soit en pourcentage (%).

Par défaut, et c'est très important à savoir, un block prend 100% de la largeur disponible. Regardez par exemple ce code XHTML qui n'utilise aucune propriété CSS :

Code : HTML

```
<p>Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Aliquam lacini
```

Essayer !

Rien de nouveau sur cet exemple. Cependant, il est très important de noter que le paragraphe (comme toutes les balises block) prend 100% de la largeur disponible par défaut.

Maintenant, pimentons d'un peu de CSS pour modifier la largeur des paragraphes. Le CSS suivant dit : "Je veux que tous mes paragraphes aient une largeur de 50%".

Code : CSS

```
p{
   width: 50%;
   text-align: justify; /* Texte justifié pour mieux voir la largeur du bl
```

Testons l'effet sur le paragraphe de l'exemple précédent :

Essayer !

Les pourcentages seront utiles pour créer un design qui s'adapte automatiquement à la résolution du visiteur. Toutefois, il se peut que vous ayez besoin de créer des blocks ayant une dimension précise en pixels :

Code : CSS

```
p{
   width: 250px;
   text-align: justify;
}
```

Essayer !

Minimum et maximum

Il vous sera aussi très pratique de demander à ce qu'un block ait une taille minimale ou maximale.C'est là qu'entrent en jeu les 4 propriétés CSS suivantes :

- min-width : largeur minimale
- min-height : hauteur minimale
- max-width : largeur maximale
- max-height : hauteur maximale

Attention cependant, ces propriétés ne fonctionnent pas sous IE 6 et fonctionnent partiellement sous IE 7. Evitez de les utiliser tant qu'IE 6 est encore assez répandu.

"Couper" un block avec un overflow

Supposons que vous ayez un long paragraphe et que vous vouliez (pour une raison qui ne regarde que vous) qu'il fasse 250px de large et 100px de haut.

Code : CSS

```
p{
   width: 250px;
   height: 100px;
   text-align: justify;
}
```

Essayer !

Attends, je croyais que le paragraphe ferait 100px de haut moi !?

Oui, mais si le texte n'a pas assez de place, le block va s'agrandir de manière à ce que tout soit visible. Si vous voulez "couper" votre paragraphe pour qu'il soit d'une dimension exacte de 250x100, vous allez devoir utiliser la propriété CSS *overflow*.
Les valeurs que peut prendre *overflow* sont les suivantes :

- visible (par défaut) : si le texte dépasse les limites de taille, le block est agrandi de manière à ce que tout soit visible. C'est ce que nous venons de constater.
- hidden : si le texte dépasse les limites, il sera tout simplement coupé. On ne pourra pas voir tout le texte.
- scroll : là encore, le texte sera coupé s'il dépasse les limites. Sauf que cette fois, le navigateur mettra en place des barres de défilement pour qu'on puisse voir tout le texte. C'est un peu comme un cadre à l'intérieur de la page.
- auto : mode "pilote automatique" :p. En gros, c'est le navigateur qui décide ou pas de mettre des barres de défilement (il en mettra si c'est nécessaire).

Avec ce CSS, nous allons pouvoir tester *overflow* :

Code : CSS

```
p /* Tous les paragraphes auront ces propriétés CSS */{
   width: 250px;
   height: 100px;
   text-align: justify;
}/* Et chacun des paragraphes aura en plus une valeur d'overflow différent
.coupe{
   overflow: hidden;
}
.barres_defilement{
   overflow: scroll;
}
.auto{
   overflow: auto;
}
```

Essayer !

En hidden, on ne peut pas voir tout le texte.
En scroll, le navigateur a mis des barres de défilement verticales et horizontales (alors qu'on n'a pas besoin de la barre horizontale).
En auto, le navigateur s'est dit "Bah tiens, j'ai pas besoin de la barre horizontale, je vais l'enlever".
En pratique, vous vous en doutez, on se sert plutôt de *overflow: auto;*
L'overflow vous sera très utile si vous avez besoin de créer des "cadres" à l'intérieur de votre page web.
Il existe une ancienne balise HTML, <iframe>, qui donne à peu près le même résultat. Toutefois, cette balise n'existe plus en XHTML (elle est dépassée). Il vous faudra donc utiliser un overflow à la place.

Les bordures

Le CSS vous offre un large choix de bordures pour décorer votre page. Tellement large qu'il existe pas mal de propriétés CSS, et que si je vous en faisais la liste complète maintenant vous sauriez plus trop où donner de la tête.
Alors, pour aller plus vite à l'essentiel, j'ai décidé de vous montrer uniquement la méga-propriété qui regroupe

tout. Vous vous souvenez d'une méga-propriété ? Oui, on en a vu une : background, c'était pour le fond de votre page web. Retournez voir le chapitre correspondant si vous avez besoin de vous rafraîchir la mémoire sur les méga-propriétés.

Là, on va directement étudier la méga-propriété, car en pratique c'est ce qu'on fait le plus souvent.
La méga-propriété en question, c'est border. C'est elle qui permet d'indiquer quelle bordure on applique au block.

Comme toute méga-propriété qui se respecte, vous pouvez lui mettre plusieurs valeurs (ici, il y a 3 valeurs à utiliser). Vous n'êtes pas obligés de mettre toutes les valeurs (vous pouvez vous contenter de 2, ou d'une seule), et l'ordre dans lequel vous mettez ces valeurs n'a aucune importance.
N'oubliez pas cela, car beaucoup de gens croient qu'il y a un ordre à respecter et se prennent la tête, alors qu'on s'en fout totalement 

Comme je viens de vous le dire, pour border on peut utiliser jusqu'à 3 valeurs pour modifier l'apparence de la bordure :

- La largeur : indiquez la largeur de votre bordure. Mettez une valeur en pixels (comme 2px), ou utilisez un des mots-clé suivants :
  - thin : bordure fine
  - medium : bordure moyenne
  - thick : bordure épaisse

  C'est le navigateur qui choisit le nombre de pixels qui correspond à thin, medium et thick. En général, c'est à peu près pareil sur tous les navigateurs.
  Personnellement, j'ai plutôt tendance à utiliser des valeurs en pixels au lieu de "thin" et compagnie... Mais c'est une question de goût après tout 
- La couleur : c'est la couleur de votre bordure. Utilisez, comme on l'a appris, soit un nom de couleur ("black", "red"...), soit une valeur hexadécimale (#FF0000), soit une valeur rgb (rgb(198, 212, 37))
- Le type de bordure : là, vous avez pas mal de choix. Votre bordure peut être un simple trait, ou des pointillés, ou encore des tirets etc... Voici les différentes valeurs disponibles :
  - none : pas de bordure (par défaut)
  - solid : un trait simple.
  - dotted : pointillés.
  - dashed : tirets.
  - double : bordure double.
  - groove : en relief.
  - ridge : effet 3D.
  - inset : autre effet 3D (on a l'impression que le block forme un creux).
  - outset : encore un autre effet 3D (on a l'impression que le block est surélevé).

Testons maintenant un peu tout ça 

Code : CSS

```
p
{
   width: 300px; /* Tous les paragraphes font 300px */
   text-align: justify; /* Tous les paragraphes sont justifiés */
}

.rien
{
   border: none;
}
.solid
{
   border: 2px solid black;
}
.dotted
{
   border: 2px dotted green;
}
.dashed
{
   border: 2px dashed red;
}
.double
{
   border: 4px double maroon;
}
.groove
{
   border: 4px groove teal;
}
.ridge
{
   border: 4px ridge fuchsia;
}
.inset
{
   border: 3px inset black;
}
.outset
{
   border: 3px outset black;
}
```

Essayer !

Il y a matière à s'amuser, comme vous pouvez le constater 

En haut, à droite, à gauche, en bas...

Qui a dit que vous étiez obligés de mettre la même bordure aux 4 côtés de votre block ?
Taratata, si vous voulez mettre des bordures différentes en fonction du côté (haut, bas, gauche ou droite), vous pouvez le faire sans problème. Dans ce cas, vous devrez utiliser ces 4 propriétés :

- border-top : bordure en haut.
- border-bottom : bordure en bas.
- border-left : bordure à gauche.
- border-right : bordure à droite.

Ce sont aussi des méga-propriétés, elles fonctionnent comme border mais ne s'appliquent donc qu'à un seul côté.
Voici quelques tests de bordures.

Code : CSS

```
p
{
   width: 350px;
}
h2
{
   border-bottom: 2px solid black;
}

.haut_bas
{
   border-top: 1px dashed red;
   border-bottom: 1px dashed red;
}
.points
{
   border-top: 3px dotted blue;
   border-left: 2px solid green;
   border-right: 2px solid green;
   border-bottom: 3px dotted blue;
}
.tres_solide
{
   border-left: 6px solid black;
   border-right: 6px solid gray;
}
```

Essayer !

En particulier, vous remarquerez que j'ai appliqué une bordure en bas du titre. Ca n'a rien à voir avec un soulignement.
Je vous l'ai dit plus haut : par défaut, un block (comme le titre) prend 100% de la largeur disponible. En l'occurence, le block prend la largeur de toute la fenêtre. Le fait de mettre une bordure en bas du titre révèle qu'il prend effectivement toute la largeur.

C'est une technique décorative assez "classe" que d'utiliser un border-bottom sur un titre. Le titre crée ainsi une ligne de séparation, ce qui rend votre page plus structurée. En plus, c'est facile à utiliser.
Après, on aime ou on n'aime pas...
... Moi j'aime 
Au fait, si vous voulez appliquer la même bordure aux 4 côtés de votre block, ne vous fatiguez pas à utiliser ces propriétés. Utilisez plutôt border qui s'applique à tous les 4 côtés simultanément.

Ca marche aussi sur des balises inline !

Je vous ai dit que toutes les propriétés CSS que nous voyons dans ces chapitres s'appliquent le plus souvent aux blocks... Mais pas toujours. Par exemple, on peut utiliser border sur des balises inline (même si on le fait un peu plus rarement).

Il y a une balise inline sur laquelle on utilise fréquemment border : c'est <img /> (pour les images). En effet, vous vous souvenez que si vous transformez une image en lien, elle se voit dotée d'une immonde bordure bleue :

Code : HTML

```
<p>
   Vous souhaitez vous rendre vers un endroit magique ? Rien de plus simple
   <a href="http://www.siteduzero.com"><img src="images/hawai.jpg" alt="Pho
</p>
```

Essayer !

Maintenant, avec la propriété border du CSS vous allez pouvoir éviter ce petit désagrément :

Code : CSS

```
a img /* Toutes les images contenues dans un lien */
{
   border: none; /* Pas de bordure */
}
```

Essayer !

Le CSS, c'est magique

Pour ceux qui auraient (déjà) oublié, j'ai utilisé l'imbrication. Si je tape "a img", cela signifie "Toutes les balises <img> contenues dans un lien <a>"... Et si vous réfléchissez bien, c'est exactement ce qu'on veut : enlever la bordure qui apparaît autour des images qui sont contenues dans des liens.

A la place de "border:none", j'aurais pu utiliser "border:0px" ou encore "border:0" (ça marche aussi). D'ailleurs, la plupart des webmasters ont tendance à mettre "border:0" tout simplement parce que c'est le plus court. N'oubliez pas : les webmasters sont des humains comme les autres. Moins ils en font, mieux ils se portent

Les marges

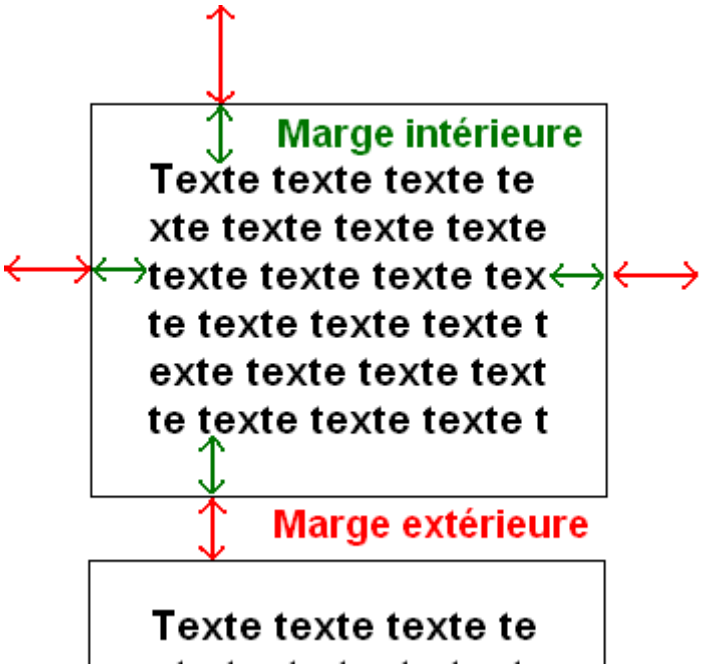
Ah, les joies des marges
C'est pas le moment de vous endormir les amis

Bien comprendre les marges, c'est vital. Si vous ne savez pas précisément comment fonctionnent les marges des blocks, vous serez bien embêtés lors de la création de votre design !

Tous les blocks possèdent des marges. Ce qui est important, c'est de savoir qu'il existe **2 types de marges** :

- Les marges intérieures
- Les marges extérieures

Regardez bien ce schéma :



Sur ce block, j'ai mis une bordure pour qu'on repère mieux ses bords.

L'espace entre le texte et la bordure est la marge intérieure (en vert).
L'espace entre la bordure et le prochain block est la marge extérieure (en rouge).

En CSS, on peut modifier la taille des marges avec les 2 propriétés suivantes :

- padding : indique la taille de la marge intérieure. A exprimer en général en pixels (px).
- margin : indique la taille de la marge extérieure. Là encore, on utilise le plus souvent des pixels.

Pour bien voir les marges, prenons 2 paragraphes auxquels je mets simplement une petite bordure :

Code : CSS

```
p
{
   width: 350px;
   border: 1px solid black;
   text-align: justify;
}
```

Essayer !

Comme vous pouvez le constater, il n'y a par défaut pas de marge intérieure (padding). En revanche, il y a une marge extérieure (margin). C'est cette marge qui fait que 2 paragraphes ne sont pas collés et qu'on a l'impression de "sauter une ligne".

Les marges par défaut ne sont pas les mêmes pour toutes les balises block. Essayez d'appliquer ce CSS à des balises qui contiennent du texte par exemple, vous verrez que là il n'y a par défaut ni marge intérieure, ni marge extérieure !

Supposons que je veuille rajouter une marge intérieure de 12px aux paragraphes :

Code : CSS

```css
p
{
   width: 350px;
   border: 1px solid black;
   text-align: justify;
   padding: 12px; /* Marge intérieure de 12px */
}
```

Essayer !

Maintenant, je veux que mes paragraphes soient plus espacés entre eux. Je rajoute la propriété margin pour demander à ce qu'il y ait 50px de marge entre 2 paragraphes :

Code : CSS

```css
p
{
   width: 350px;
   border: 1px solid black;
   text-align: justify;
   padding: 12px;
   margin: 50px; /* Marge extérieure de 50px */
}
```

Essayer !

Mais ??? Une marge s'est mise à gauche aussi !

Bah oui, *margin* (comme padding d'ailleurs) s'applique aux 4 côtés du block.
Si vous voulez indiquer une marge en haut, en bas, à gauche et à droite, il va falloir utiliser des propriétés plus précises... Et vous allez voir que les créateurs du CSS se sont pas foulés, ça fonctionne comme pour border

En haut, à droite, à gauche, en bas... Et on recommence !

Ce qui serait bien, ce serait que vous reteniez les traductions suivantes en anglais :

- top : haut
- bottom : bas
- left : gauche
- right : droite

Comme ça, vous pouvez retrouver toutes les propriétés de tête.
Je vais quand même vous faire la liste des propriétés pour margin et padding, histoire que vous soyez sûrs que vous avez compris le principe.

Voici la liste pour margin :

- margin-top : marge extérieure en haut.
- margin-bottom : marge extérieure en bas.
- margin-left : marge extérieure à gauche.
- margin-right : marge extérieure à droite

Et la liste pour padding :

- padding-top : marge intérieure en haut.
- padding-bottom : marge intérieure en bas.

- padding-left : marge intérieure à gauche.
- padding-right : marge intérieure à droite.

Dites, c'est moi ou j'ai l'impression de me répéter ?

Bon, pour tester on va utiliser margin-bottom pour indiquer un espace en bas de chaque paragraphe, ce qui nous évitera d'avoir une marge à gauche comme tout à l'heure.
Et aussi, pour le fun, je vais rajouter une marge à gauche pour le titre (h1) afin qu'il soit un peu décalé.

Code : CSS

```
p
{
   width: 350px;
   border: 1px solid black;
   text-align: justify;
   padding: 12px;
   margin-bottom: 50px;
}

h1
{
   margin-left: 30px;
}
```

Essayer !

Juste une question... Quelle est la différence entre un *margin-left* et un *text-indent* ?

Bonne question ça

Vous vous souvenez que, pour décaler un peu le titre, on avait utilisé dans un chapitre de la partie II la propriété *text-indent*. Ici, *margin-left* semble donner le même résultat...
Visuellement le résultat est le même, certes. Si vous mettez une bordure, vous verrez une petite différence : la bordure de gauche est décalée avec un margin-left, alors que ce n'est pas le cas avec text-indent.

En pratique, je vous conseille de n'utiliser le *text-indent* que sur des paragraphes, pour décaler la première ligne. Le reste du temps, jouez avec les margin et les padding pour placer comme bon vous semble vos blocks.

Centrer des blocks

Il est tout à fait possible de centrer des blocks, c'est même très pratique pour réaliser un design centré quand on ne connaît pas la résolution du visiteur.

Toutefois, et c'est très important, cela ne fonctionne que si vous avez indiqué une largeur précise (*width*) au block. Si vous n'avez pas indiqué de largeur, le block ne sera pas centré !

Prenons le cas de nos petits paragraphes. On leur a déjà donné une largeur précise ; maintenant si on veut les centrer sur notre écran nous allons mettre la valeur "auto" à la propriété *margin*, comme ceci :

Code : CSS

```
p
{
   width: 350px; /* On a indiqué une largeur (obligatoire) */
   border: 1px solid black;
   text-align: justify;
   padding: 12px;
   margin: auto; /* On peut donc demander à ce que le block soit centré ave
   margin-bottom: 20px;
}
```

Essayer !

Ainsi, le navigateur centre automatiquement nos paragraphes. N'oubliez pas, je le dis et je le répète, que le margin:auto ne peut fonctionner que si vous avez indiqué une largeur précise pour votre block !

Q.C.M.

Laquelle de ces balises est de type inline ?

- ◯ acronym
- ◯ h5
- ◯ dl

Laquelle de ces balises est une balise universelle ?

- ◯ div
- ◯ ul
- ◯ img

width permet de définir...

- ◯ La hauteur d'un block
- ◯ La marge d'un block
- ◯ La largeur d'un block

Quelle apparence aura la bordure créée par ce code ?

Code : CSS

```
border:4px outset black;
```

- ◯ La bordure créera un effet surélevé
- ◯ La bordure sera double
- ◯ La bordure aura un effet creux

padding-bottom est la marge...

- ...intérieure gauche
- ...intérieure basse
- ...extérieure haute

margin-right est une marge...

- ...intérieure gauche
- ...intérieure droite
- ...extérieure droite

Je veux que tous mes liens soient entourés d'une bordure double à gauche et à droite, de couleur verte. Que manque-t-il à ce CSS ?

Code : CSS

```
a
{
border-left:3px double green;
}
```

- Il manque border-top
- Il manque border-right
- Il manque border-bottom

Quand a-t-on le droit d'utiliser un margin:auto pour centrer un block ?

- N'importe quand
- Quand on a défini une largeur précise pour le block
- Quand le block utilise un overflow:auto;

Correction !

Statistiques de réponses au QCM

Toutes ces notions théoriques sur les blocks sont un peu barbantes, je peux le comprendre.
Néanmoins, c'est un passage obligé et (bonne nouvelle) c'est la dernière barrière avant la création du design de votre site. En gros, une fois que vous aurez compris ça, vous pourrez enfin voler de vos propres ailes.

Bref, il est plus que jamais nécessaire d'être attentif. D'autant plus que ce n'est pas terminé : il vous reste à lire la seconde partie de ce chapitre.
Si j'ai un bon conseil à vous donner : ne vous pressez pas inutilement. Prenez le temps d'être certain d'avoir tout compris. Et surtout...

Faites des tests
Faites des tests
Faites des tests
Faites des tests
Faites des tests
Faites des tests
Faites des tests

Ca commence à rentrer là ? 
En effet, vous ne pouvez pas apprendre convenablement si vous ne vous exercez pas. Prenez des exemples de CSS de ce chapitre, et essayez de les modifier. C'est en pratiquant que les choses rentrent dans la tête, et pas

autrement