

# Les boîtes css

© Eric Nême

## RAPPEL

### 1) Qu'est-ce qu'une boîte ?

Toutes les balises html peuvent être considérées comme des boîtes.

Exemple de code (on s'intéresse ici à la balise body et son contenu):

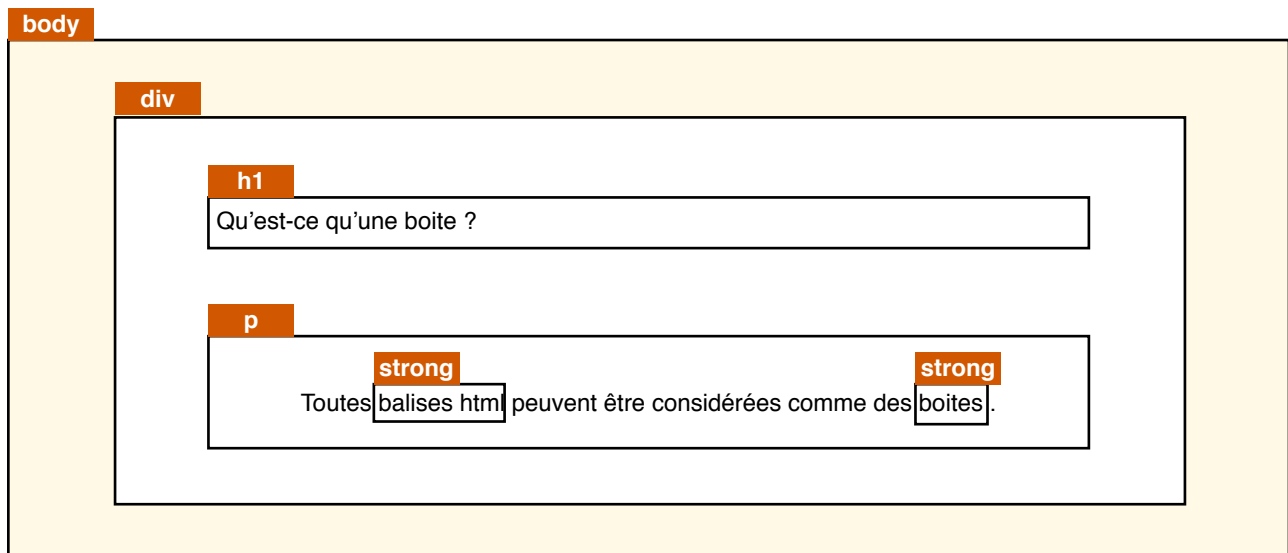
```
<body>
  <div>
    <h1>Qu'est-ce qu'une boîte ?</h1>
    <p>
      Toutes les <strong>balises html</strong> peuvent être considérées comme des <strong>boîtes</strong>.
    </p>
  </div>
</body>
```

Dans cet exemple il y a 6 boîtes : body, div, h1, p, strong, strong.

### 2) Imbrications

On parle d'imbrication quand une balise html est située à l'intérieur d'une autre.

Les 6 boîtes de notre exemple sont organisées d'une certaine manière du fait des imbrications.



Pour décrire cette organisation on peut utiliser les termes suivants :

- Enfant : div est l'enfant de body, h1 est enfant de div, etc.
- Parent : body est le parent de div, div est le parent de h1 et p, etc.
- Frère : h1 et p sont des éléments frères.
- Ancêtre : div est l'ancêtre de h1, p, strong et strong.
- Descendant : h1, p, strong et strong sont les descendants de div et body.

Voici la définition de ces termes selon le w3c (<http://www.yoyodesign.org/doc/w3c/css2/conform.html>):

#### Enfant

On appelle un élément A l'enfant de l'élément B, si et seulement si B est le parent de A ;

#### Frère

On appelle un élément A un frère d'un élément B, si et seulement si A et B ont un même parent. Si l'élément B A apparaît avant B dans l'arbre du document, on dit que A est le frère précédent et B le frère suivant

#### Descendant

On appelle un élément A un descendant d'un élément B si A est soit (1) un enfant de B, soit (2) un enfant d'un autre élément C qui est lui-même un descendant de B ;

#### Ancêtre

On appelle un élément A l'ancêtre d'un élément B, si et seulement si B est un descendant de A ;

### 3) Flux d'affichage

On appelle flux naturel l'ordre d'apparition et l'organisation de l'affichage et des éléments contenus dans les balises par un navigateur.

Le navigateur affiche les éléments au fur et à mesure de la lecture du code html : de haut en bas et de gauche à droite.

Si le code html contient des caractères blancs (espaces, tabulations, retours à la lignes) le navigateur n'en tiendra pas compte lors de l'affichage.

En fait il ne retient qu'un **seul espace**, celui qui permet de séparer les mots à l'intérieur des balises.

Par exemple : `<p>toutes les balises</p>` s'affiche «toutes les balises».

Dans notre exemple le résultat d'affichage du code html est le suivant :

#### Qu'est-ce qu'une boîte ?

Toute les **balises html** peuvent être considérées comme des **boîtes**.

On constate :

- Qu'il existe un retour à la ligne après «Qu'est-ce qu'une boîte ?»

Ce retour ne peut pas être expliqué par un retour à la ligne dans le code source puisqu'on vient de préciser que le navigateur ne tenait aucun compte des caractères blancs de ce type pour l'affichage. La seule explication vient des balises elles-mêmes.

Ainsi certaines balises sont dites de **type bloc (block)** lorsqu'elles provoquent des retours à ligne et de **type en ligne (inline)** lorsqu'elle n'en provoque pas.

Dans notre exemple visuellement on peut dire : h1 et p sont des balises de type bloc alors que strong est une balise de type en ligne.

Si on rajoutait à la suite du code un élément frère à div on constaterait visuellement que la balise div est également de type bloc car cet élément apparaîtrait à la ligne.

Comment connaître le type bloc ou en ligne d'une balise ?

La documentation que je vous ai fournie et les sites internet référençant les balises mentionnent souvent ces caractéristiques.

- Que le texte dans `<h1>` s'affiche en grand et gras et celui dans `<strong>` en gras.

Les navigateurs intègrent un minimum de styles qui permettent traduire de manière visuelle et «basique» la sémantique des balises. Ainsi la balise `<strong>` signifiant «important» s'affiche en gras, `<h1>` étant le titre principal s'affiche en grand et gras.

En conséquence si on veut changer l'aspect de <h1> il faudra passer par un style css. Par exemple pour <h1>, annuler l'affichage en gras et réduire la taille du texte à celle de <p> pourrait passer par :

```
h1 {  
  font-weight : normal;  
  font-size : 1em;  
}
```

Les styles des navigateurs ajoutent également à la plupart des balises de type bloc des marges externes et internes.

Une bonne habitude consiste à faire un «reset» de ces marges en listant toutes les balises de type bloc du document ex :

```
body, div, h1, p {  
  margin : 0;  
  padding : 0;  
}
```

Cette déclaration est à placer en premier. Rien n'est définitif, on peut tout à fait redéfinir ces marges par la suite, par exemple :

```
h1 {  
  margin-bottom : 12px;  
  padding : 5px;  
}
```

### Modification des types de balise

La propriété css «display» permet de modifier le comportement «naturel» des balises :

Par exemple :

h1 { display : inline; } transforme les balises h1 en balise de type en-ligne.

strong { display : block; } transforme les balises strong en balise de type bloc.

Attention, dans notre exemple si on écrit «h1 { display : inline; }» le paragraphe restera sous le texte de <h1>. En effet le paragraphe (<p>) est de type bloc, et refusera donc de se placer à la gauche de <h1>. Il faut donc rendre p également de type en ligne :

h1, p { display : inline; }

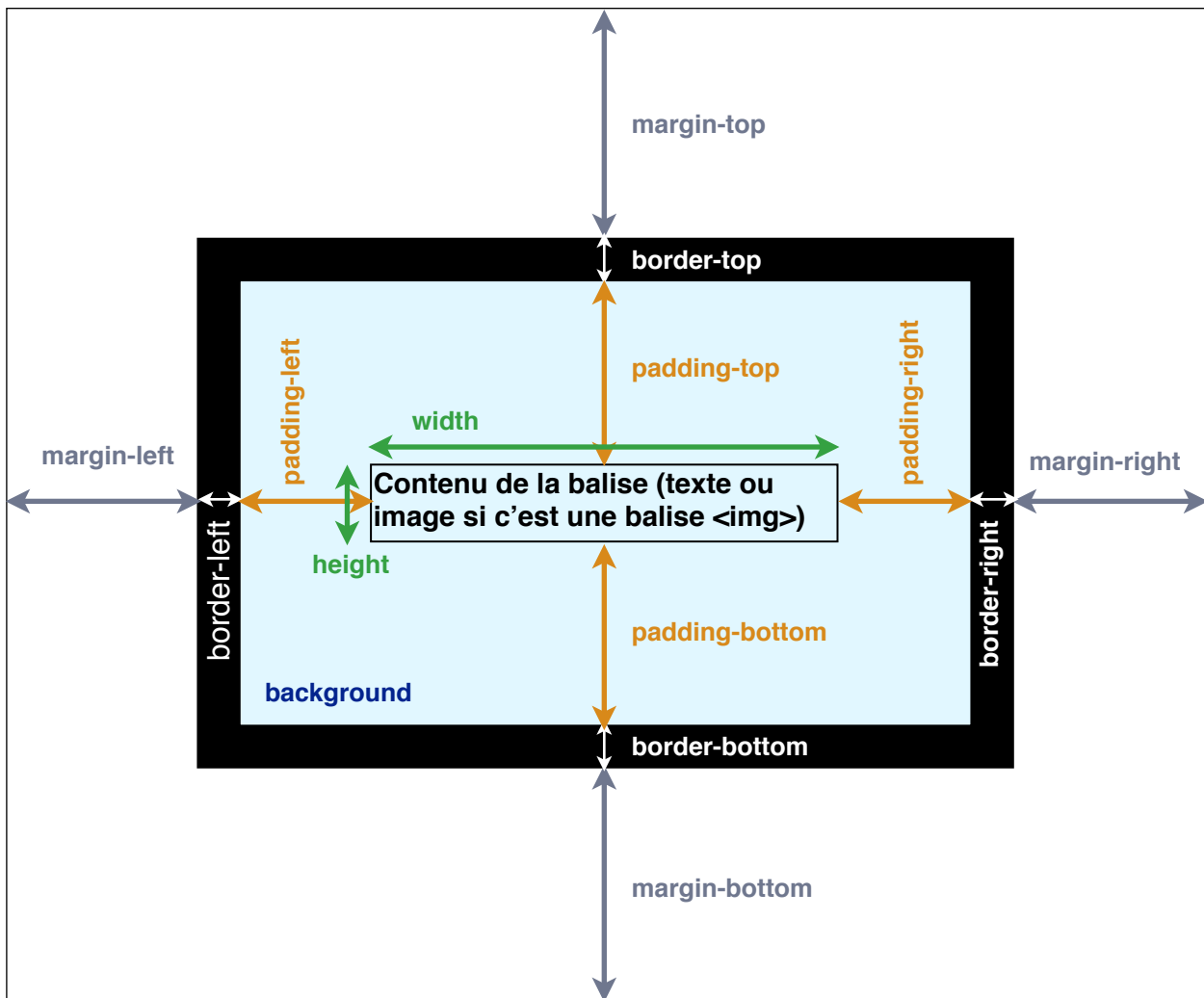
La propriété float permet également de «contrarier» l'effet des types des balises. Je vous renvoie à l'archive «flottementPositionnement.zip» téléchargeable sur mon serveur pour plus de détail.

## PROPRIÉTÉS ET DIMENSIONS D'UNE BOITE DE TYPE BLOC

Comme toute balise est une boîte, on peut à l'aide des feuilles de style css intervenir sur l'aspect de ces boîtes. Une boîte de type bloc a les propriétés suivantes :

- Hauteur (height)
- largeur (width)
- marges externes (margin)
- marges internes (padding)
- bordures (border), fond (background).

## 1) Représentation d'une boîte type



*Les fines bordures représentent les limites des marges externes et celles du contenu. Elles ne sont là que pour les besoins du schéma, elles ne s'affichent pas dans le navigateur.*

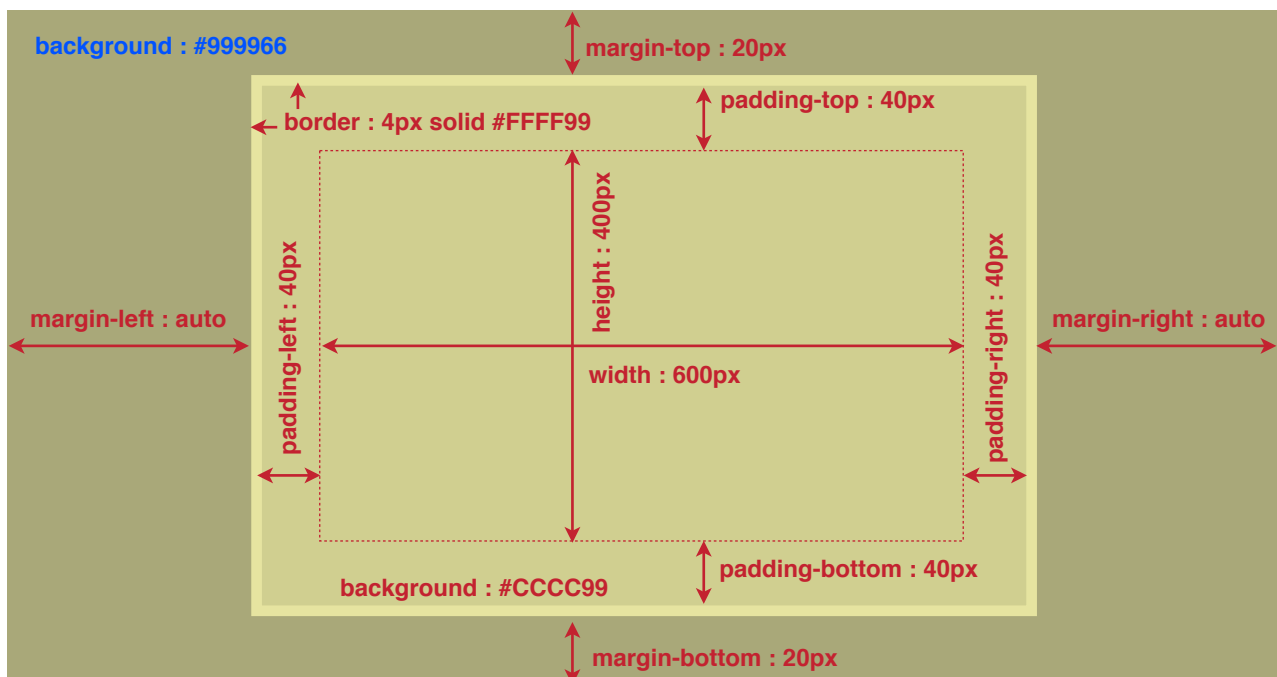
De ce schéma on peut en déduire :

- Que les propriétés largeur (width) et hauteur (height) fixent la largeur et la hauteur du **contenu de la balise**.  
Cas particuliers :
  - Si le contenu est une image (balise <img>) les propriétés width et height n'ont pas de sens puisqu'elles sont définies par les dimensions de l'image elle-même.
  - Body est une boîte particulière qui représente toute la zone d'affichage du navigateur. On ne peut donc pas fixer la largeur et la hauteur de body, elles dépendent directement de la taille de la fenêtre du navigateur. Ainsi en faisant varier la taille de la fenêtre on fait varier les dimensions de body.
- Que le fond (background : couleur et/ou image) n'apparaît qu'à l'intérieur de la zone délimitée par les bordures. Si la bordure est en pointillés le fond apparaît entre chaque pointillés
- Que les marges externes (margin) sont toujours transparentes.  
Pour body les marges externes n'ont aucun sens.

## 2) Application à notre code

Commençons par les boîtes body et div en plaçant les styles suivants :

```
body {  
    background : #999966;  
}  
div {  
    width : 600px;  
    height : 400px;  
    margin : 20px auto;  
    border : 4px solid #FFFF99;  
    padding : 40px;  
    background : #CCCC99;  
}
```



*En rouge les propriétés de div, en bleu celles de body*

- Les marges externes droite et gauche à auto permettent le **centrage horizontal** de la boîte div. Tout repose sur le principe général suivant :

**Pour toute balise de type bloc :**

$\text{margin-left} + \text{border-left} + \text{padding-left} + \text{width} + \text{padding-right} + \text{border-right} + \text{margin-right} = \text{width du parent}$

**Dans notre cas pour la boîte div :**

$\text{auto} + 4\text{px} + 40\text{px} + 600\text{px} + 40\text{px} + 4\text{px} + \text{auto} = \text{width de body}$

soit :  $\text{auto} + 688\text{px} + \text{auto} = \text{width de body}$

**Quelle est la largeur de body ?**

On en sait rien... Tout dépend de la taille de fenêtre du navigateur !!!

**Utilité des marges externes à auto ?**

Le navigateur va calculer **lui-même** ces valeurs en appliquant le principe, exemple :

Si body fait 800px de large :  $56\text{px} + 688\text{px} + 56\text{px} = 800\text{px}$  (les marges externes droite et gauche = 56px)

Si body fait 960px de large :  $136\text{px} + 688\text{px} + 136\text{px} = 960\text{px}$  (les marges externes droite et gauche = 136px)

**Grâce à la valeur auto, la boîte div est toujours centrée quelle que soit la taille de la fenêtre du navigateur.**

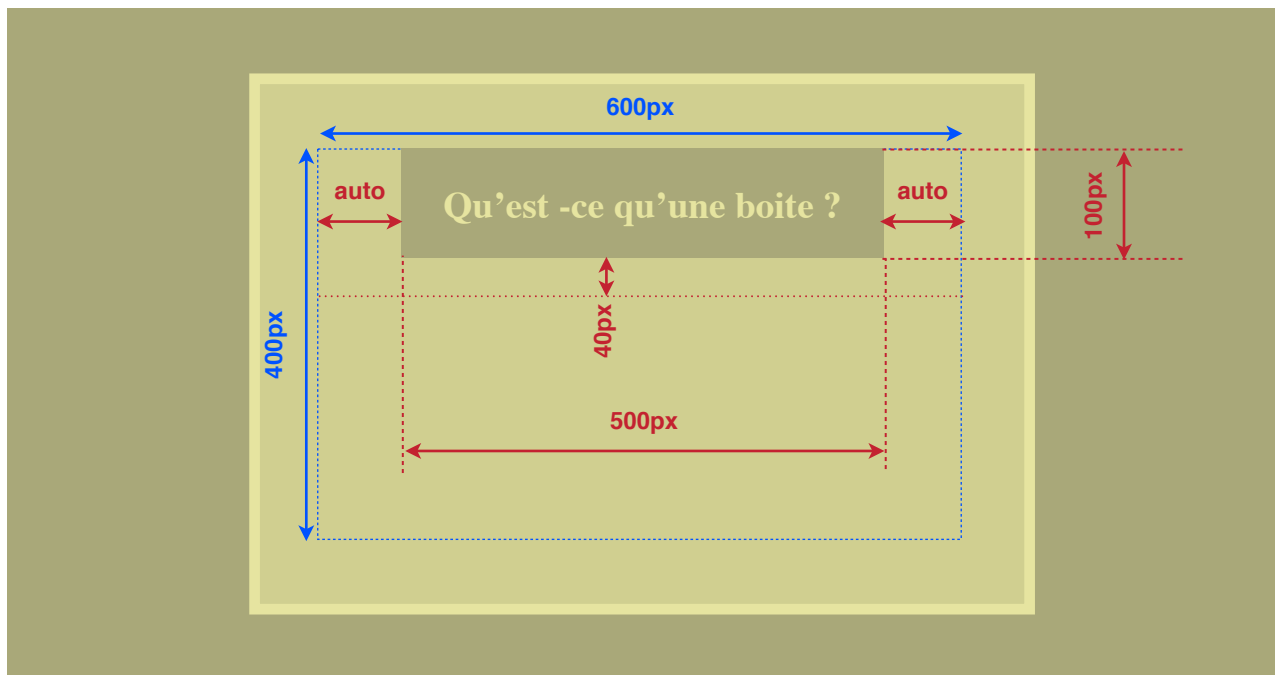
- La zone délimitée par les propriétés width et height de div :

Cette zone sera occupée par les **descendants de div**, à savoir, h1 et p. Ces deux éléments ne pourront pas sortir de cette zone (sauf dans certains cas très particuliers).

Dans le schéma, j'ai symbolisé cet espace avec un trait fin rouge en pointillé qui n'apparaît pas lors de l'affichage dans un navigateur.

### 3) Placement de l'enfant h1

```
h1 {
  width : 500px;
  height: 100px;
  line-height: 100px;
  background : #999966; // Couleur de fond identique à celle body
  color: #FFFF99; // Couleur du texte identique à celle de la bordure de div
  margin : 0 auto 40px auto;
  text-align: center;
}
```



*En rouge les propriétés de h1, en bleu celles de div*

Retour sur le principe précédent appliqué cette fois-ci à h1

**h1 à pour parent div et div à une largeur de 600 px**

$\text{auto} + 0 + 0 + 500 + 0 + 0 + \text{auto} = 600$  (h1 n'a pas de bordure ni de marge interne d'où les 0)

soit :  $50 + 500 + 50 = 600$

Les marges externes droite et gauche de h1 sont donc de 50 px

On aurait pu écrire directement : `h1 { margin : 0 50px 40px 50px }` puisque cette fois-ci la largeur de div est fixe.

Centrage horizontal du texte «Qu'est-ce qu'une boîte ?»

On a fixé la largeur du contenu de h1 à 500px, mais la largeur du texte est inférieure à 500px.

La seule méthode permettant de centrer un texte à l'intérieur de la zone de contenu est de faire appel à la propriété css `text-align`.

## Centrage vertical du texte «Qu'est-ce qu'une boîte ?»

Voici une astuce permettant de centrer verticalement un texte à l'intérieur de la zone de contenu.

**ATTENTION : cette technique n'est valable que pour un texte tenant sur une seule ligne.**

Il suffit d'utiliser la propriété `line-height` en lui donnant la même valeur que celle de la propriété `height` (ce qui suppose qu'il faut mettre en place la propriété `height`) :

`height: 100px;`

`line-height: 100px;`

## Les 40 px de la marge externe du bas

Cette marge «repoussera» vers le bas de 40px l'élément frère à `h1` suivant c'est-à-dire `p`

## 4) Placement des frères de `h1`

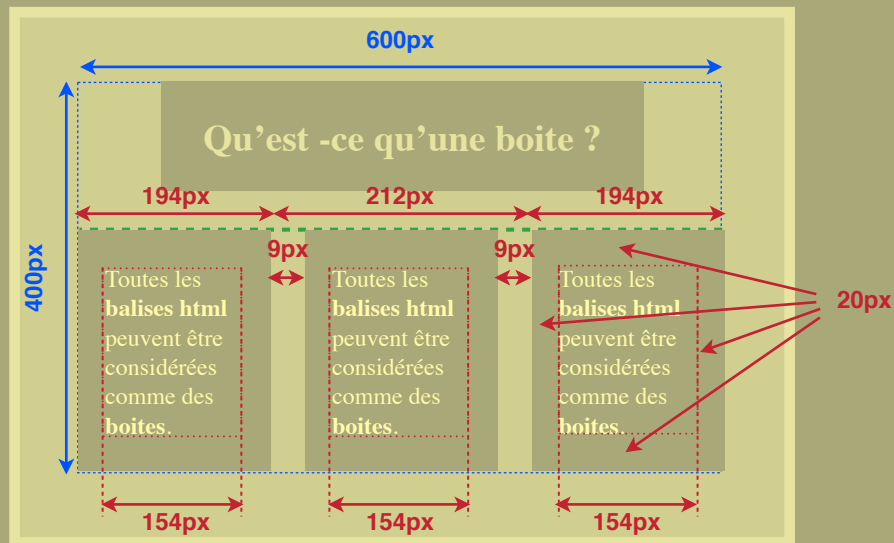
Nous allons modifier un peu le code html pour rajouter 2 frères à `h1`

```
<body>
  <div>
    <h1>Qu'est-ce qu'une boîte ?</h1>
    <p>
      Toutes les <strong>balises html</strong> peuvent être considérées comme des <strong>boîtes</strong>.
    </p>
    <p id= «colonne_centre»
      Toutes les <strong>balises html</strong> peuvent être considérées comme des <strong>boîtes</strong>.
    </p>
    <p>
      Toutes les <strong>balises html</strong> peuvent être considérées comme des <strong>boîtes</strong>.
    </p>
  </div>
</body>
```

`h1` à donc maintenant 3 frères (`p`, `p`, `p`) et le `p` du milieu «est étiqueté» avec l'id «colonne\_centre»

Le but est de fabriquer 3 colonnes :

```
p {
  margin: 0; // Pour effacer les marges que le navigateur place par défaut
  width : 154px;
  float: left;
  padding : 20px;
  background : #999966;
  font-size: 1.5em; // Pour augmenter un peu le corps des paragraphe
  color: #FFFF99;
}
#colonne_centre { // Pour le paragraphe avec l'id «colonne_centre»
  margin: 0 9px;
}
```



*En rouge les propriétés des p, en bleu celles de div, en vert celle de h1*

## Rappel de la propriété float

Lorsqu'un élément est flottant il indique aux éléments de type bloc qui le suivent que l'espace à sa gauche ou à sa droite est libre. Si le(s) éléments suivants vont tenter d'occuper cet espace devenu libre.

Float : left positionne l'élément à gauche et rend l'espace à sa droite libre.

Float : right positionne l'élément à droite et rend l'espace à sa gauche libre.

La référence de placement à gauche et à droite est l'élément ancêtre le plus proche de type bloc.

Ainsi pour le premier p l'ancêtre de type bloc le plus proche est l'élément div, il se positionnera le plus à gauche possible à l'intérieur de cette zone (en bleu sur le schéma).

Pour le second p la référence est toujours div, mais il ne pourra pas «chevaucher» le premier p donc il se positionnera juste à sa droite.

## Calcul de la largeur occupée par les 3 paragraphes avec... toujours le même principe

Paragraphe 1 :  $0 + 0 + 20px + 154px + 20px + 0 + 0 = 194px$

Paragraphe du centre :  $9px + 0 + 20px + 154px + 20px + 0 + 9px = 212px$

Paragraphe 3 :  $0 + 0 + 20px + 154px + 20px + 0 + 0 = 194px$

Largeur totale :  $194px + 212px + 194px = 600px$  (soit la largeur du contenu du bloc ancêtre le plus proche : le div)