

# La gestion des erreurs et des problèmes HTML

Par CIPHERED



[www.openclassrooms.com](http://www.openclassrooms.com)

*Licence Creative Commons 6 2.0  
Dernière mise à jour le 17/01/2010*

## Sommaire

Sommaire .....	2
ENCORE UN LINUXIEN !! .....	1
La gestion des erreurs et des problèmes HTML .....	3
Les problèmes de mise en page et les déformations .....	3
Si ça ne fonctionne pas .....	3
À guetter sur le validateur .....	4
Les problèmes de positionnement .....	4
La barre d'outils Web Developer .....	4
Utiliser la barre d'outils .....	5
On attaque le problème .....	6
Les problèmes d'espacement et de centrage .....	10
La balise <body> .....	10
Les listes .....	11
Les paragraphes .....	11
L'exercice 2 .....	11
Centrer tout le design .....	14
D'autres problèmes récurrents .....	14
Pourquoi mon image ne s'affiche pas ? .....	14
Pourquoi la propriété vertical-align ne fonctionne-t-elle pas ? .....	15
Centrer verticalement dans un bloc .....	15
Pourquoi mes objets ne suivent-ils pas le z-index que j'ai défini ? .....	16
Pour aller plus loin .....	16
w3schools.com .....	16
Dans les tutoriels .....	16
Partager .....	17



# La gestion des erreurs et des problèmes HTML



Bien salut,

Si vous vous trouvez ici, c'est probablement parce qu'il vous arrive souvent de rencontrer des problèmes dans la conception de pages HTML. Eh bien ce tutoriel est là pour vous aider non seulement à régler ces problèmes, mais aussi pour que vous puissiez bien les comprendre.

Bien entendu, ce tutoriel ne remplace pas la [FAQ du forum XHTML/CSS](#). Il vous servira plutôt de référence pour les cas particuliers. Ce que vous lirez ne vous dira pas explicitement comment résoudre vos erreurs ou vos problèmes. C'est en parcourant ce texte que vous pourrez adopter vos propres méthodes de résolution de problèmes et ainsi vous faire gagner du temps.

Ne soyez pas déçus en fin de page si vous avez toujours des questions, car la résolution de problèmes est un sujet très abstrait et très vaste. J'essaierai de vous expliquer les problèmes les plus courants.

Dernier point, j'ai mis ce tutoriel dans la catégorie **Difficile**. Les exercices et les marches à suivre ne sont pas difficiles en elles-mêmes, mais vous devez être assez expérimentés en HTML et en CSS pour pouvoir saisir l'intégralité de ce tutoriel. J'invite tout de même les débutants à y jeter un coup d'œil, car il y a quelques points qui leur seraient utiles, tout spécialement la première sous-partie.



Certaines captures d'écran de ce tutoriel seront légèrement différentes de ce que vous verrez chez vous, mais ce n'est que mon thème modifié pour Firefox.

Prêts ? Allons-y !

Sommaire du tutoriel :



- [Les problèmes de mise en page et les déformations](#)
- [Les problèmes de positionnement](#)
- [Les problèmes d'espacement et de centrage](#)
- [D'autres problèmes récurrents](#)
- [Pour aller plus loin](#)

## Les problèmes de mise en page et les déformations

Donc vous êtes en train de coder, et vous décidez de voir ce que ça donne. Vous sauvegardez votre document, puis vous ouvrez le navigateur pour admirer votre travail.

Sauf que... ce que vous voyez ne correspond pas vraiment à ce que vous aviez imaginé. Sur le coup, ça peut être assez décourageant. Mais il faut voir le problème d'un côté positif. Ce n'est qu'un autre petit défi à relever qui vous donnera plus d'expérience.

Lorsque je suis confronté à un problème de mise en page, je regarde d'abord à partir d'où le problème se manifeste. Normalement, tout ce qui se trouve avant la ligne de code erronée s'affichera bien. Regardez où la déformation se produit. L'erreur se situera presque toujours à cet endroit et se règlera facilement.

*Si ça ne fonctionne pas*

Parfois le problème est un peu plus complexe. il peut venir d'ailleurs. Et ce n'est pas toujours évident d'examiner une page de code, surtout si elle fait dans les 500 lignes. Dans ce cas, il y a un outil qui peut trouver votre problème plus vite que vos yeux, et on l'appelle le [validateur XHTML](#).

Bien sûr, les résultats de la validation sont en anglais et assez difficiles à comprendre si vous n'êtes pas familiers avec la langue. Mais l'outil vous indiquera à quelles lignes ça ne va pas. Avec un logiciel comme Notepad++ qui numérote vos lignes, vous serez en mesure de retrouver la ligne donnée et de corriger l'erreur ensuite.

Procéder de cette façon vous épargnera bien des maux. Les erreurs d'inattention sont les plus courantes.

### À guetter sur le validateur

Vous vous retrouverez peut-être perdus devant tous les résultats du validateur. Pour l'instant, le but ici n'est pas de corriger TOUTES les erreurs, seulement celles qui causent des problèmes d'affichage.

Si vous avez :

Code : HTML

```
<p>  
  <var><strong>Hello World !</var></strong> <!-- Les balises var  
  et strong entrecroisées -->  
</p>
```

vous aurez probablement :

**XML Parsing Error: Opening and ending tag mismatch: var line 2 and strong**

**XML Parsing Error: Opening and ending tag mismatch: strong line 2 and var**



Pour ceux qui n'auraient jamais entendu parler de la balise **var** (elle est assez rare), sachez que ce n'est qu'une balise **inline** qui définit un texte variable, différent de l'habituel. Elle est comparable à la balise **span**.

Chaque erreur porte sur l'une des deux balises croisées. Si vous lisez **line 2**, l'erreur se situe à la ligne 2 de votre code. Un éditeur avancé comme Notepad++ (que vous devriez déjà avoir) vous aidera à trouver la ligne si elle est plus loin dans le document.

Ensuite, si vous avez ce code :

Code : HTML

```
<divHello World !</div>
```

cela générera plusieurs erreurs, dont celle vue précédemment, mais aussi celle-ci :

**end tag for "divHello" omitted, but OMITTAG NO was specified**

Évidemment la balise **divHello** n'existe pas, il est donc assez simple de dire que l'erreur a été causée par une balise non fermée. Vérifiez aussi les autres erreurs car elles afficheront assez souvent l'extrait fautif et vous pourrez trouver le problème.

L'erreur **end tag for "element" omitted** s'affichera aussi si vous avez omis de fermer un élément avec une balise **</element>** ou encore si vous avez oublié la barre oblique à la fin de la balise, par exemple avec la balise **<img src="" />** .

Quoi qu'il arrive, ne soyez pas désappointés si vous voyez 10 erreurs au validateur après avoir modifié quelques caractères d'un code parfaitement valide. Le validateur est un script, il ne peut donc pas savoir que vous avez oublié de fermer une balise et vous dira que toutes les fermetures suivantes ne ferment pas la bonne chose. Bref, c'est sûrement une seule erreur qui en crée plusieurs.

## Les problèmes de positionnement

Nous allons ici voir une sorte de problème plus complexe. Il arrive parfois que vous vouliez positionner un élément mais que pour une raison inconnue, il ne s'affiche pas à l'endroit désiré. C'est habituellement parce que vous lui avez mal indiqué sa position.

## La barre d'outils Web Developer



C'est un outil que j'utilise très souvent lorsque je suis sur les forums du SdZ pour résoudre vos problèmes. Avec cette barre, je peux modifier le HTML et le CSS affichés à l'écran pour ainsi pouvoir tester des solutions. Je peux facilement trouver ce qui ne va pas.

La barre d'outils est le nec plus ultra des gadgets du développeur web. Les critiques sont là pour en témoigner.

#### Citation : Ryan Boots sur [addons.mozilla.org](https://addons.mozilla.org)

This isn't a toolbar - it's a Swiss Army knife for web development. I don't know how many times it's occurred to me to try and view this or that through the toolbar, and more often than not it's right there. A required download for every web developer.

#### Citation : Traduction

Ce n'est pas une barre d'outils - c'est plutôt un couteau suisse pour le développement Web. Je ne sais combien de fois j'ai dû m'en servir pour voir ceci ou cela, et plus souvent qu'autrement c'est juste là. Un téléchargement nécessaire pour chaque développeur web.

La barre a été développée par Chris Pederick et est disponible dans plusieurs langues. Soyez certains d'être sur la partie française du site de modules complémentaires avant de télécharger.

### Télécharger la barre d'outils Web Developer

[Ou visiter le site officiel du module](#)

### Utiliser la barre d'outils



Désactiver - Cookies - CSS - Formulaires - Images - Information - Divers - Entourer - Redimensionner - Outils - Voir Source - Options



Par défaut, la barre affiche le texte à côté des icônes. Pour prendre en compte les plus petites résolutions, je l'ai désactivé pour cette capture d'écran.

Le module a énormément de fonctions, je ne pourrai donc pas tout vous détailler. Mais voici les fonctions que j'utilise le plus souvent.

#### L'édition du code

Dans les menus **CSS** et **Divers**, vous aurez accès à l'édition HTML et CSS. Pratique pour essayer une solution CSS ou ajouter un texte de test sur une page que l'on ne peut modifier. Les modifications sont provisoires. Elles s'annuleront dès que vous quitterez la page ou fermerez la fenêtre.

L'édition du code est semblable à l'extension Firebug, pour ceux qui connaissent. Elle ne permet pas de naviguer dans le code facilement, mais la modification se fait très rapidement.

#### L'encadrement coloré des éléments

Le menu **Entourer** permet d'entourer rapidement certains éléments de bordures colorées afin de bien voir la place qu'ils prennent. Parfois, il arrive qu'un bloc prenne trop ou pas assez de place et qu'il en pousse un autre. Vous pourrez ainsi l'entourer et voir la place qu'il prend.

Par exemple, si un élément ne s'affiche pas, vous pouvez l'entourer d'une couleur qui contraste le thème du site. Si vous ne repérez pas la bordure, c'est peut-être parce que vous avez oublié de donner une dimension à votre objet (il fait 0 pixel par 0 pixel, il est donc invisible) ou encore qu'il se cache derrière un autre.

Le menu en question propose déjà pas mal d'options, mais pour ceux qui en veulent plus, vous pouvez même **Entourer un élément personnalisé**. Cliquez sur le lien. Une fenêtre s'ouvre. Vous pouvez ainsi taper le nom de l'élément que vous voulez entourer. Si vous tapez par exemple `div` et choisissez la couleur verte, tous les `div` s'entoureront d'une bordure verte. Vous pouvez aussi viser des éléments spécifiques, comme dans le CSS ! Taper `div#specifique` entourera le `div` à l'id *specifique*.

## Le redimensionnement de la page

Je ne passerai pas vingt minutes sur le menu **Redimensionner**, mais je tenais à ce que vous sachiez qu'il vous permet de redimensionner votre fenêtre à des dimensions très précises afin que vous puissiez très facilement tester votre site sur différentes résolutions.

## Les autres fonctions

Pour les autres fonctions, ne vous en faites pas, essayez-les et vous comprendrez bien assez vite leur fonctionnement. 😊

## On attaque le problème

Nous voici déjà à ce premier exercice. J'ai déjà eu à résoudre un problème semblable sur le forum et je pense que vous pourrez bien vous en sortir. Vous aurez besoin de la barre d'outils Web Developer.

## Aller à l'exercice 1

Si vous avez un peu d'expérience, tout devrait bien se passer. Si vous débutez, je vous laisse me suivre dans ma démarche.

## Le code

Actuellement le code est très simple. Tout est dans la page et le CSS est intégré entre les balises `<style></style>` .

**Secret (cliquez pour afficher)**

### Code : HTML

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8"
/>
<title>La gestion des erreurs et des problèmes HTML - Exercice
1</title>

<style type="text/css">
<!--
body {
  background-color: #F5F5F5;
}
div#entete {
  width: 750px;
  height: 90px;
  background-color: #999999;
}
div#entete_gauche {
  float: left;
  width: 400px;
  height: 110px;
}
div#entete_gauche img {
  display: block;
  width: 400px;
  height: 90px;
  background-color: #666666;
}
div#entete_droite {
  float: right;
  width: 350px;
  height: 90px;
  text-align: center;
}
div#menu {
```

```

float: left;
width: 150px;
background-color: #000000;
}
div#menu ul {
margin: 0;
padding: 0;
}
div#menu li {
line-height: 22px;
padding-left: 5px;
border-bottom: 1px solid #FFFFFF;
list-style-type: none;
font-family: Verdana, Geneva, sans-serif;
font-size: 12px;
color: #FFFFFF;
}
div#contenu {
width: 580px;
margin-left: 150px;
padding: 10px;
background-color: #CCCCCC;
}
p {
margin: 0;
text-indent: 25px;
}
-->
</style>

</head>

<body>

<h1>La gestion des erreurs et des problèmes HTML</h1>
<h2>Exercice 1</h2>

<div id="entete">
  <div id="entete_gauche"></div>
  <div id="entete_droite">Bienvenue sur notre site !</div>
</div>

<div id="menu">
  <ul>
    <li>Lien 1</li>
    <li>Lien 2</li>
    <li>Lien 3</li>
    <li>Lien 4</li>
  </ul>
</div>

<div id="contenu">
  <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit.
Cras ac viverra quam. Praesent suscipit, sem ut condimentum
iaculis, justo purus consequat elit, fringilla dignissim nisi quam
ac sapien. Vestibulum ante ipsum primis in faucibus orci luctus et
ultrices posuere cubilia Curae; Ut et augue turpis, eget tincidunt
justo. Sed auctor quam pharetra lorem consequat a consectetur nibh
aliquet. Praesent sit amet ipsum et est vulputate fringilla ac vel
massa. Donec ac lectus mauris, vel consequat odio. In porttitor
venenatis dui, in pharetra dolor hendrerit ut. Aliquam vitae felis
quis ipsum porttitor blandit. Donec tempor porttitor velit, a
tempor quam auctor quis. Phasellus venenatis vestibulum sem et
mattis. Pellentesque augue metus, tristique lacinia egestas id,
consectetur accumsan mauris. Suspendisse potenti.</p>
  <p>Nam fringilla feugiat tortor, eget varius felis sagittis
in. Sed vulputate neque ut lectus ullamcorper sit amet rutrum
purus aliquet. Mauris laoreet pellentesque quam, suscipit
tincidunt dolor iaculis vel. Class aptent taciti sociosqu ad

```

```

litora torquent per conubia nostra, per inceptos himenaeos.
Curabitur non arcu diam. Mauris in eros sem. Donec at mauris
dolor, sed placerat dui. Sed consequat metus sed augue posuere
lobortis. Phasellus dictum sodales felis quis interdum. Donec nec
magna sed orci pretium volutpat. Proin mauris dui, mattis sit amet
sollicitudin quis, tempus quis urna. In cursus iaculis velit eu
porttitor. Proin nibh dolor, ultrices vel consequat nec, ultricies
id neque. Sed et vehicula orci. Nulla varius leo sed diam
porttitor sed faucibus neque pretium.</p>
</div>

<h2>Le résultat voulu</h2>
<div></div>

</body>
</html>

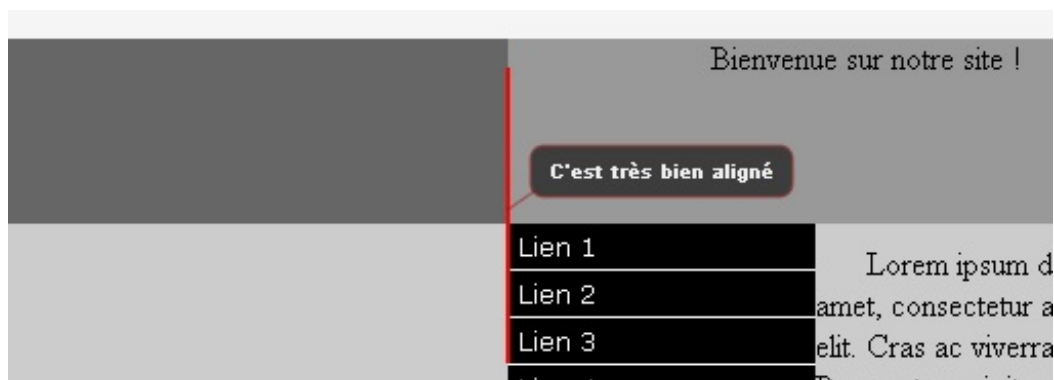
```

Un œil très exercé couplé à un esprit très entraîné pourrait voir le problème, mais on considérera que ce n'est pas le cas de tout le monde.

### Regarder le bloc mal positionné

Si on regarde le menu, on peut voir qu'il n'est pas positionné normalement. Il n'est pas centré par rapport à d'autres éléments, c'est-à-dire que tous les éléments sont à des distances inégales de lui. Bref, il est impossible d'obtenir une telle position facilement. Je ne crois pas non plus que le codeur ait pu poser le bloc là en lui mettant même des marges "par accident". C'est donc presque assuré que le menu est poussé par quelque chose d'autre qui prend sa place et qui le force à se déplacer. On ne voit pourtant rien.

C'est là que vous devrez remarquer le petit détail. Voyez-vous que la bordure gauche du menu est très bien alignée sur la bordure droite du logo (un carré gris en l'occurrence 🤔) ? On peut donc suspecter le logo.

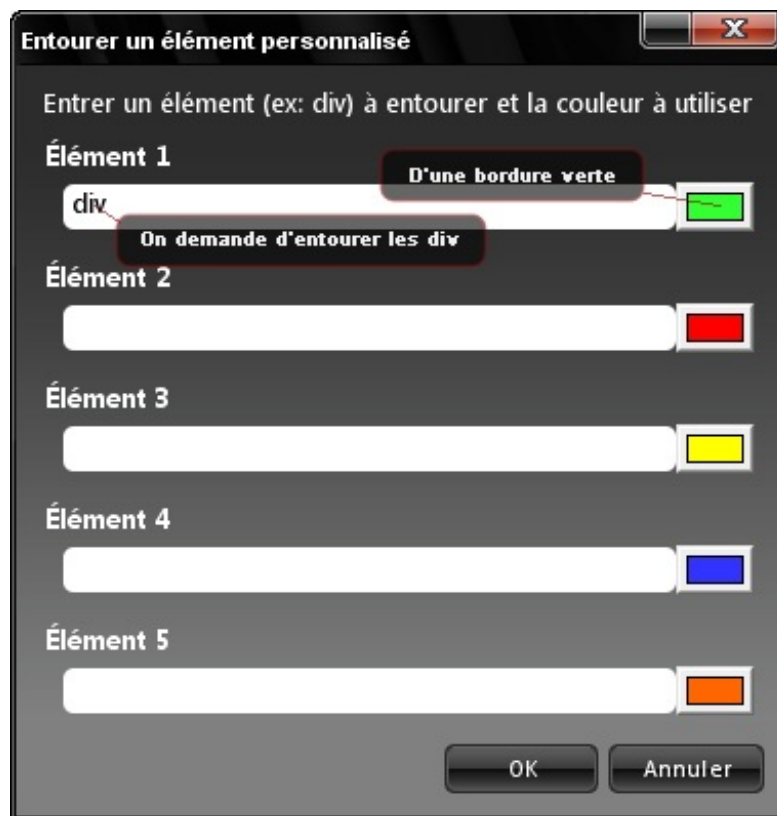


Le problème ici est que le logo est entièrement visible. Donc, s'il poussait le menu, on verrait bien du gris foncé qui prend la place. Il faut donc se tourner vers un objet qu'on ne voit pas nécessairement, mais qui a approximativement la même taille que le logo. C'est là qu'on utilisera notre nouvelle amie, la barre d'outils.

### Web Developer rentre dans la place

La barre d'outils Web Developer est bien en vue. On va lui indiquer d'entourer en vert tous les éléments `<div>`. Dans le menu **Entourer**, sélectionnez **Entourer un élément personnalisé**. La fenêtre suivante s'ouvre :





En cliquant sur **OK**, vous verrez que chaque `div` a été entouré d'une bordure verte. Avec le temps, vous pourrez remarquer de plus en plus facilement celles qui ont des dimensions ou des tailles bizarres.

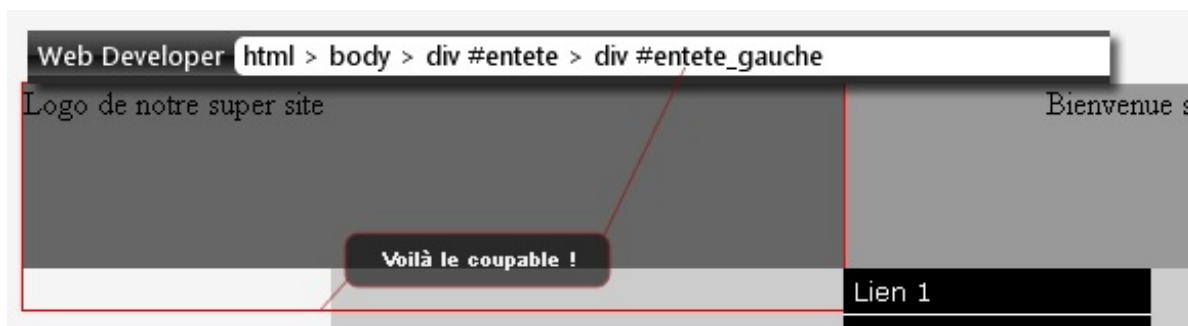


Avec Web Developer, vous pouvez entourer n'importe quel élément de la page. Vous pouvez aussi spécifier une classe (par exemple **.information**) ou un ID (par exemple **#entete**). Web Developer se chargera d'entourer les éléments demandés. Bref, vous pouvez appeler des éléments comme vous le faites dans le CSS !



Lorsque vous aurez repéré le rectangle suspect, nous allons utiliser un autre outil afin de repérer le bloc et d'avoir accès à ses informations. Dans le menu **Entourer**, cliquez à nouveau sur **Entourer un élément personnalisé** afin de le désactiver, puis sélectionnez **Entourer l'élément survolé**, juste au-dessus.

Cette fonction fera en sorte que l'élément que vous survolez s'entoure d'une bordure rouge, et son `id` ou sa classe ainsi que ses éléments parents s'affichent dans une barre d'outils temporaire.



### Trouver une solution

La question, ici, est « pourquoi ce bloc prend-il toute cette hauteur ? » L'image, qui est le seul élément à l'intérieur, fait 90 pixels de hauteur. Le bloc parent `div#entete` fait également 90 pixels de hauteur. Le bloc `div#entete_gauche` semble être trop haut pour rien. Il suffit de le réduire à 90 pixels.

Code : CSS

```
div#entete {
  width: 750px;
  height: 90px;
  background-color: #999999;
}
div#entete_gauche {
  float: left;
  width: 400px;
  height: 90px; /* La hauteur est passée de 110 pixels à 90 pixels */
}
div#entete_gauche img {
  display: block;
  width: 400px;
  height: 90px;
  background-color: #666666;
}
```

### Tester la solution

Courage, on a presque fini. 🤔

On va une dernière fois utiliser la barre d'outils Web Developer. La plupart du temps vous pourrez essayer vos solutions directement dans votre éditeur, sauvegarder votre page puis la voir dans le navigateur.

Mais d'autres fois, surtout si vous aidez un Zér0 sur le forum, vous n'aurez pas accès à la modification de sa page. Plutôt que de tout refaire sur votre ordinateur, vous pouvez vous servir des outils d'édition de codes HTML et CSS. Dans notre cas, on aura besoin du CSS.



Attention ! À partir de ce point, lorsque vous ouvrirez la console d'édition CSS, veillez à ne pas changer d'onglet, car vos modifications seront effacées ! Si vous voulez revenir sur cette page pendant que vous travaillez, ouvrez une nouvelle fenêtre.

Donc, allez dans le menu **CSS** puis sélectionnez **Éditer les CSS**. La console CSS s'ouvre et vous montre le CSS de la page. Allez à la ligne où se trouve l'erreur, puis corrigez-la de la même manière que si vous le faisiez dans l'éditeur. Magie ! 🎩 La modification s'applique ! Et normalement, vous devriez voir le résultat voulu !

## Les problèmes d'espacement et de centrage

Il arrive parfois que certains éléments créent des espaces qu'on ne veut tout simplement pas. Cette courte section vous expliquera ou plutôt listera les éléments qui créent des marges indésirables.

### La balise <body>

La balise `<body>` a des marges d'environ 10 pixels de chaque côté. Pour l'éliminer, il suffit de mettre un `margin: 0;` à l'élément dans le CSS. Cette solution vous permettra de coller votre contenu directement sur les bords de la fenêtre.

### Les listes

Un problème récurrent est que les listes ont tendance à créer des espaces vides au-dessus des menus horizontaux. Les listes peuvent aussi poser des problèmes lorsqu'elles sont horizontales et qu'il faut les centrer.

Une liste horizontale a par défaut une marge extérieure en haut et une marge intérieure (`padding-left`) à gauche. Ainsi, si vous centrez votre liste horizontale, elle sera toujours décalée d'une vingtaine de pixels, voire plus, sur la droite. Vous pouvez donc remettre le `padding-left` à 0.

Si votre liste est verticale et que vous avez des puces, utiliser cette astuce tassera votre texte sur la gauche, et soit les puces disparaîtront de la page, soit elles se placeront à l'extérieur de la bordure du bloc.

Pour éviter ce genre de problème, il existe un attribut CSS. Vous pouvez choisir si vous voulez que la puce soit à l'intérieur ou à l'extérieur des marges définies. L'attribut à utiliser est **`list-style-position`**. Les valeurs possibles sont **`inside`** et **`outside`**.

<code>list-style-position: outside;</code>	<code>list-style-position: inside;</code>
<ul style="list-style-type: none"><li>• Élément 1</li><li>• Élément 2</li><li>• Élément 3</li><li>• Élément 4</li><li>• Élément 5</li></ul>	<ul style="list-style-type: none"><li>• Élément 1</li><li>• Élément 2</li><li>• Élément 3</li><li>• Élément 4</li><li>• Élément 5</li></ul>

### Les paragraphes

Les paragraphes aussi causent souvent des problèmes. Souvent, le premier paragraphe contenu dans un bloc créera une marge extérieure en haut du bloc. Mettre **`margin-top: 0;`** au bloc qui contient le paragraphe ne fonctionnera pas. C'est au paragraphe qu'il faut ajouter l'attribut. Si vous voulez tout de même préserver des espaces entre vos paragraphes, pensez à ajouter un **`margin-bottom`**, et vous ne devriez pas avoir de problème.

On appelle cela le **problème de fusion de marges**. Les marges du paragraphe sont appliquées au bloc qui le contient.

## L'exercice 2

Nous voici rendus au deuxième exercice de ce long mini-tuto. 😊

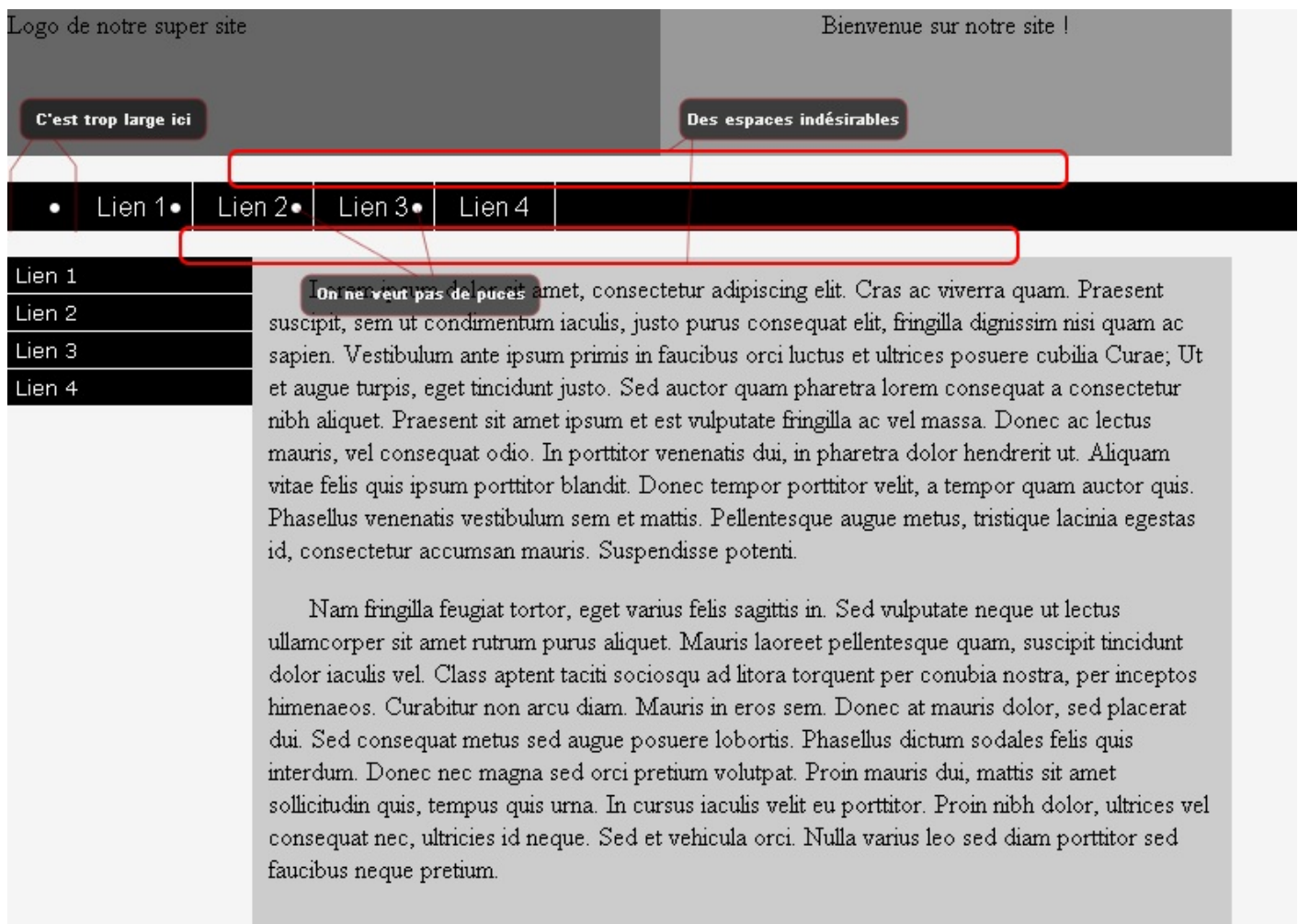
Il est assez simple et vous pourrez le terminer grâce aux connaissances acquises plus haut.

### Aller à l'exercice 2

Réussi ? En fait j'ai ajouté un petit détail concernant les puces, mais vous devriez savoir comment les enlever facilement. Pour ceux qui n'ont pas résolu le problème, suivez ce que j'ai fait. Je n'ai pas détaillé autant que pour l'exercice 1, mais maintenant vous devriez être en mesure d'utiliser la barre d'outils Web Developer. Je me trompe ?

### Identifier les problèmes

Premièrement, nous allons identifier tous les problèmes de la page :



### Régler les problèmes

Dans ce cas-ci, nous avons appris la nature de la plupart des problèmes ci-dessus, ça sera donc simple.

- **Les espaces indésirables** : c'est parce que les listes ont par défaut des marges en haut et en bas. On doit les remettre à zéro.
- **L'espace à gauche** : les listes ont une marge intérieure à gauche afin que les puces soient visibles. On n'en a pas besoin, donc on met **padding-left** à zéro.
- **Les puces de la liste** : dans une liste, si on ne veut pas de puces, on doit passer par **list-style-type: none** pour les enlever.

### Tester les solutions

Sortez la barre d'outils Web Developer, on va s'entraîner à l'utiliser. Avec l'édition CSS, on va aller ajouter les lignes nécessaires. Vous aurez aussi besoin de la commande **Entourer l'élément survolé** du menu **Entourer**. Passez avec le curseur sur la liste et aux environs jusqu'à ce que vous puissiez en ressortir son ID.

Munis de cette information, nous pourrions trouver où ajouter les lignes de code.

**Secret (cliquez pour afficher)**

Code : CSS

```
body {
  background-color: #F5F5F5;
}
div#entete {
  width: 750px;
  height: 90px;
```

```
background-color: #999999;
}
div#entete_gauche {
float: left;
width: 400px;
height: 90px;
}
div#entete_gauche img {
display: block;
width: 400px;
height: 90px;
background-color: #666666;
}
div#entete_droite {
float: right;
width: 350px;
height: 90px;
text-align: center;
}
ul#menu_h {
width: 750px;
height: 30px;
line-height: 30px;
background-color: #000000;
margin: 0; /* On enlève les espaces en haut et en bas */
padding: 0; /* On enlève l'espace sur la gauche */
list-style-type: none; /* On enlève les puces */
}
ul#menu_h li {
float: left;
padding: 0 15px;
border-right: 1px solid #FFFFFF;
font-family: Arial, Helvetica, sans-serif;
color: #FFFFFF;
}
div#menu {
float: left;
width: 150px;
background-color: #000000;
}
div#menu ul {
margin: 0;
padding: 0;
}
div#menu li {
line-height: 22px;
padding-left: 5px;
border-bottom: 1px solid #FFFFFF;
list-style-type: none;
font-family: Verdana, Geneva, sans-serif;
font-size: 12px;
color: #FFFFFF;
}
div#contenu {
width: 580px;
margin-left: 150px;
padding: 10px;
background-color: #CCCCCC;
}
p {
margin-top: 0;
text-indent: 25px;
}
```

Maintenant, les marges n'ont presque plus de secrets pour vous. Comme je l'ai dit dans l'introduction le sujet est très vaste, il est donc possible que je ne vous aie pas apporté de clé pour tous vos maux de tête. Ne lâchez pas, il reste encore du contenu à lire.

## Centrer tout le design

À la demande de plusieurs, je vais terminer cette sous-partie en traitant du centrage de toute la page. En CSS, il est possible de centrer un bloc (p, ul, div) avec **margin:auto**. Le but est d'égaliser les marges de chaque côté, afin de faire en sorte que l'élément soit bien centré dans son parent.

### Différence entre text-align:center et margin:auto

Ces deux éléments donnent à peu près le même résultat, un centrage. Sauf qu'il y a tout de même une différence.

- **text-align: center**  
**text-align** sert à centrer le contenu **inline** d'un bloc. Donc, si vous avez un **div** qui contient du texte, des images et un tableau, que vous lui appliquez **text-align:center**, le texte et les images se centreront, mais pas le tableau. Pourquoi ? Le texte et les images sont de type **inline**, et le tableau est de type **block**, ou **bloc**, tout dépend du terme que vous utilisez.
- **margin: auto**  
Au contraire de **text-align:center**, **margin:auto** centre les éléments de type **block**. Donc, si vous désirez centrer un tableau, il ne faut pas mettre un **text-align:center** à l'élément parent, mais bien un **margin:auto** à l'élément à centrer.

Maintenant, si vous désirez centrer tout le design, le mieux à faire serait de l'encadrer au complet dans un **div** puis de centrer celui-ci à l'aide de la méthode décrite plus haut.

### Centrer un bloc en position:absolute

Si votre bloc a une position absolue (donc l'attribut **position:absolute**), le moyen de le centrer est un peu plus complexe, mais simple une fois le coup compris. La première chose est de mettre **left:50%**, afin que l'espace depuis la gauche de la page soit la moitié de l'espace disponible. Cependant, essayez et vous verrez quelques problèmes encore. En fait c'est plutôt la bordure gauche de votre bloc qui est au centre, alors que ce devrait être le milieu. Pour cela, il faut tasser le bloc vers la gauche, mais de combien ?

Il nous faudra pour cela trouver la distance entre le bord gauche et le centre de notre bloc. Normalement, c'est facile. Il suffit de diviser sa largeur par 2. Après votre petit calcul, nous devrons utiliser **margin-left** afin de déplacer notre bloc vers la gauche et lui mettre comme valeur le résultat de votre précédent calcul, mais négatif. Vous n'avez peut-être rien compris, alors plus visuellement :

Code : CSS

```
div#bloc_a_centrer {  
    position: absolute; /* Pour les position:absolute uniquement */  
    width: 600px; /* On a défini la largeur */  
    left: 50%; /* On tasse le bloc jusqu'au milieu */  
    margin-left: -300px; /* On aligne le milieu du bloc avec le  
milieu de la page en utilisant un margin-left négatif dont la  
valeur est la moitié de la largeur du bloc à centrer (600 / 2 = 300  
-> -300) */  
}
```

## D'autres problèmes récurrents

Nous voici déjà presque à la fin de ce tutoriel, et nous allons finir avec deux courtes sous-parties. Dans celle-ci, nous verrons d'autres problèmes récurrents.

### Pourquoi mon image ne s'affiche pas ?

Le problème par excellence... Combien de messages sont postés sur les forums par des membres dont l'image ne s'affiche pas ? Ne les comptez pas.

Pour commencer, soyez certains que le lien est correct. Si votre image se trouve dans un sous-dossier, indiquez-le ! Par exemple, la page sur laquelle l'image ne s'affiche pas se nomme **index.html**. À côté de la page, vous avez un dossier nommé **images**. Dans ce dossier, une image nommée **maison.jpg**.

Si dans votre code vous avez :

```

```

c'est plus que certain que ça ne fonctionnera pas. Essayez plutôt :

```

```

Ensuite, si ça ne fonctionne toujours pas, essayez ceci (compatible avec Firefox seulement). Faites un clic droit sur l'image qui ne veut pas s'afficher et sélectionnez **Afficher l'image**. Si votre image ne s'affiche pas, c'est signe que votre lien n'est pas bon.

Cela est peut-être causé par l'extension. Si le nom de votre image est **voiture.jpg**, n'écrivez pas **voiture.JPG** ! Vous devez respecter **à la lettre** la casse (minuscules ou majuscules). Donc si l'extension est **.JPG**, **.jpeg** ou **.JPEG**, ce n'est pas **.jpg**.

### *Pourquoi la propriété vertical-align ne fonctionne-t-elle pas ?*

C'est un sujet qui m'est revenu récemment sur le forum, à l'heure où j'écris ces lignes. Beaucoup de gens croient à tort que **vertical-align** sert à la même chose que l'attribut **valign** qu'on applique aux cellules de tableau.

La propriété **vertical-align** s'applique sur des balises *inline*, comme un bloc de texte. Si je veux qu'une partie de mon texte soit en indice, je peux utiliser les balises **<sub></sub>**, qui sont là pour ça. Sauf que les balises en question utilisent elles-mêmes la propriété **vertical-align** ! Dans son style par défaut (soit le style qui est utilisé par le navigateur si rien n'est défini), on a quelque chose comme ça :

Code : CSS

```
sub {  
    vertical-align: sub;  
}
```

Et non, on n'utilise pas les marges pour les exposants ou les indices !

Plus souvent vous utiliserez **vertical-align** pour des images. Les deux valeurs les plus utiles seront :

- **text-top**
- **text-bottom**

En images, cela donnerait :

L'image est en  **text-top**

L'image est en  **middle**

L'image est en  **text-bottom**

Maintenant, vous saurez pourquoi **vertical-align** ne fonctionnait pas. Mais si vous vouliez centrer verticalement du texte dans un bloc, qui n'est pas nécessairement une cellule de tableau, il est dur de dire que ce tutoriel vous a aidés.

### *Centrer verticalement dans un bloc*

Donc, dans le même ordre d'idées que dans le paragraphe différent, apprenons à centrer du texte et/ou des images verticalement. Pour ce faire, nous allons utiliser une propriété CSS peu connue des débutants, soit **line-height**. En l'utilisant, il est possible de définir la hauteur des lignes de texte. Si vous voulez par exemple modifier l'interligne dans vos paragraphes, c'est pour vous.



Sauf qu'on peut aussi utiliser **line-height** pour centrer des objets verticalement. Par exemple, si votre texte de 20 pixels se retrouve dans un bloc d'une hauteur de 40 pixels, faites **line-height: 40px**. Pourquoi ? Grâce à votre commande, la ligne de texte aura une hauteur de 40 pixels, et le contenu haut de seulement 20 pixels **se centrera verticalement sur la ligne de lui-même, et comme celle-ci est de la même hauteur que votre conteneur, votre contenu sera centré verticalement !** C'est facile, et ça marche.

Si vous avez plus d'une ligne de texte, ça posera problème. Vous pouvez toujours essayer avec un **line-height**, mais ça ne donnera probablement pas le résultat voulu. Dans ce cas, la seule solution est de jouer avec les marges du conteneur et du texte, et d'y aller à l'oeil. Ou bien si vous tenez absolument à centrer votre texte, faites un tableau à une cellule pour entourer le contenu et appliquez-y **valign="middle"**. Mais ce n'est pas moi qui vous l'ai dit. 🤔

### *Pourquoi mes objets ne suivent-ils pas le z-index que j'ai défini ?*

Le **z-index** s'applique sur les éléments positionnés avec l'élément `position`. Par défaut, c'est-à-dire si vous ne spécifiez pas l'attribut, tous les objets sont programmés sur `static`. Cette valeur signifie que l'élément est là, qu'il suit le reste et se place normalement.

Le **z-index** ne fonctionne que sur les positions absolues, relatives ou fixes. Les éléments dotés de ces caractéristiques se positionnent différemment des autres et passent très souvent par-dessus ou en dessous de leurs semblables. C'est pourquoi on a besoin de définir un ordre.

Il existe aussi un moyen de ne pas définir de **z-index**. Quand vous utilisez une position absolue, relative ou fixe, les éléments s'empileront suivant l'ordre dans lequel ils se chargent. Plus l'élément vient avant dans le code, plus il sera en dessous.

Dans :

Code : HTML

```
<div id="bloc_1"></div>
<div id="bloc_2"></div>
```

si on prend en compte le fait que ces deux blocs sont en **position: absolute**, lorsqu'on visitera la page, `div#bloc_1` s'affichera en dessous de `div#bloc_2` car il vient avant.

## Pour aller plus loin

Pour ceux qui voudraient continuer leur apprentissage ou qui n'ont pas résolu leurs problèmes, je vous suggère ici certaines références qui pourraient vous être utiles.

**w3schools.com**



permettra d'essayer ce que vous avez appris de nouveau.

Ce site web est en anglais, mais c'est probablement la référence la plus complète en matière de HTML, CSS, JavaScript, PHP, SQL, ASP, XML et autres langages apparentés. Dans la section HTML, le site liste **toutes** les balises HTML, leurs attributs, et dans quels *Doctype* ceux-ci sont supportés. Il y a aussi un éditeur HTML en direct qui vous

**Aller à w3schools.com**

## Dans les tutoriels

Vous ne vous en doutez peut-être pas, mais les tutoriels du site peuvent vous aider à réaliser de nombreuses choses. Visitez la section **XHTML/CSS** et vous trouverez des tas de choses intéressantes.

Vous pouvez :

- faire un bloc dont la hauteur est 100% ;
- faire un menu horizontal ;



- centrer un bloc verticalement et/ou horizontalement ;
- faire des colonnes de hauteurs égales ;
- et plus !

Voilà, j'espère avoir pu vous aider dans vos déboires. Sinon, soyez à l'affût des mises à jour. Il reste certainement du contenu à ajouter. D'ailleurs, si vous avez des suggestions, vous pouvez les laisser dans les commentaires ou dans le [sujet](#) dédié à ce tutoriel.

Avant de terminer ceci, j'aimerais remercier tous ceux qui ont suggéré des ajouts à ce tutoriel avant la publication initiale, soit :

- [Polok](#)
- [Ko Reika](#)
- [Juliusninet](#)
- [6pril](#)
- [Ssx3max](#)
- [Snouf](#)
- [Davailink](#)

Certains m'avaient demandé de parler de Firebug, mais malheureusement je n'utilise pratiquement pas cet outil sauf pour pouvoir voir le code bien indenté. Je me renseignerai davantage sur l'outil avant d'en traiter dans ce tutoriel.

Bon, j'espère en avoir aidé le plus possible même si la tâche n'était pas simple. Comme je l'ai précédemment mentionné, le sujet est extrêmement vaste et long à couvrir. J'espère que la plupart y ont trouvé leur compte.

### Partager



Ce tutoriel a été corrigé par les [zCorrecteurs](#).