



Html et Css

Objectifs

- ✓ Savoir créer et mettre à jour des pages web en HTML en respectant les normes W3C, et définir l'aspect graphique du site à l'aide des CSS.

01

Les langages Html et Css

HTML et CSS sont des langages permettant la structuration et mise en forme de page web. Les dernières évolutions nous conduisent à HTML 5 et CSS 3.

Pour observer un code HTML, il suffit de se rendre sur une page web quelconque et d'effectuer un clic droit > code source de la page. Il s'agit d'une suite d'instructions qui forme un Code Source.

Le code-source forme une suite de lignes qui sera interprétée et donnera le rendu que vous voyez à l'écran.



➤ Informations

vous ne pourrez pas voir le code PHP d'un site car il s'agit d'un langage [interprétré côté serveur](#)

02

Compatibilité des navigateurs

Voici les statistiques d'utilisations des navigateurs sur les 10 dernières années :

Navigateur	Google Chrome	FireFox	Internet Explorer	Safari	Opera
2016	68 %	19.1 %	6.3 %	3.7 %	1.5 %
2015	63.9 %	21.6 %	8 %	3.8 %	1.5 %
2014	55.7 %	26.9 %	10.2 %	3.9 %	1.8 %
2013	48.4 %	30.2 %	14.3 %	4.2 %	1.9 %
2012	35.3 %	37.2 %	20.1 %	4.3 %	2.4 %
2011	23.8 %	42.8 %	26.6 %	4 %	2.5 %
2010	10.8 %	46.3 %	36.2 %	3.7 %	2.2 %
2009	3.9 %	45.5 %	44.8 %	3 %	2.3 %
2008	Inexistant	36.4 %	54.7 %	1.9 %	1.4 %
2007	Inexistant	31 %	58.6 %	1.7 %	1.5 %
2006	Inexistant	25 %	66 %	Inexistant	1.6 %
-					

Les comportements et l'affichage peuvent être différents d'un navigateur à un autre. Lors de la conception d'un site web, il est important de télécharger tous les navigateurs et de faire plusieurs tests d'affichage.



➤ Informations

Un navigateur permet de se connecter à un nom de domaine (ou adresse IP) afin de consulter des pages web hébergées sur un serveur (serveur = ordinateur allumé 24h/24h) quelque part dans le monde, et raccordé au réseau Internet.

03

L'organisation W3C

Le W3C est une organisation chargée de standardiser et faire évoluer le langage HTML.

Le W3C dicte des normes d'interprétation que les navigateurs doivent suivre (*un peu comme le code de la route*). L'objectif recherché est qu'un même code s'affiche de la même manière sur tous les moteurs de rendus.

Le W3C met à disposition un [espace de validation](#) pour savoir si le code que l'on écrit respecte bien les règles. Un code totalement valide (dit « code propre ») permet une meilleure interprétation des navigateurs et peut contribuer à un meilleur positionnement sur les moteurs de recherche (référencement).

Pendant sa création et lorsque le site est terminé, il est important de « le tester » sur tous les navigateurs et sur l'espace de validation W3C.



➤ Bon à savoir

Les avantages d'un code valide et propre : meilleure compatibilité assurée entre les navigateurs et prédisposition améliorée pour un référencement naturel.

04

Quelques Outils

Avant de démarrer l'apprentissage et l'écriture de code Html / Css, nous allons devoir nous équiper. Avez-vous déjà vu un jardinier sans sa pelle ou son rateau ? et bien c'est pareil pour nous, nous allons récupérer quelques outils sur le web :

- Firefox (navigateur)
- Firebug (extension qui permet d'observer et d'édition un code-source en direct)
- Webdevelopper (extension qui fournit des outils pour le développeur web)
- Notepad++ ou sublime text (éditeur de texte - simple lors d'une première utilisation)

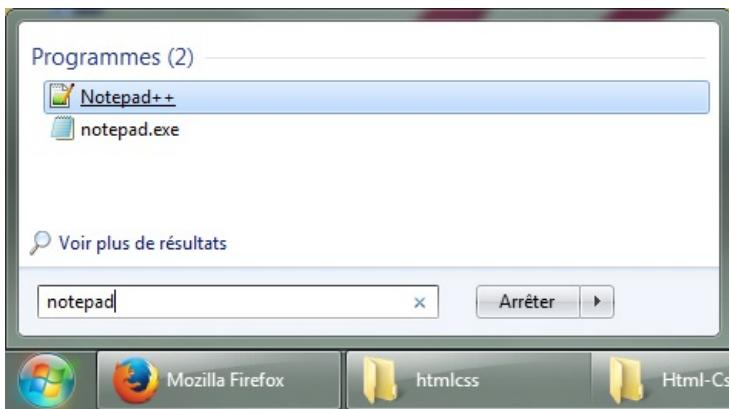
Nous utiliserons donc Firefox pour la suite du cours, cependant si vous êtes attaché à Google Chrome ou un autre navigateur, vous pourrez sans aucun doute télécharger les extensions équivalentes.

05

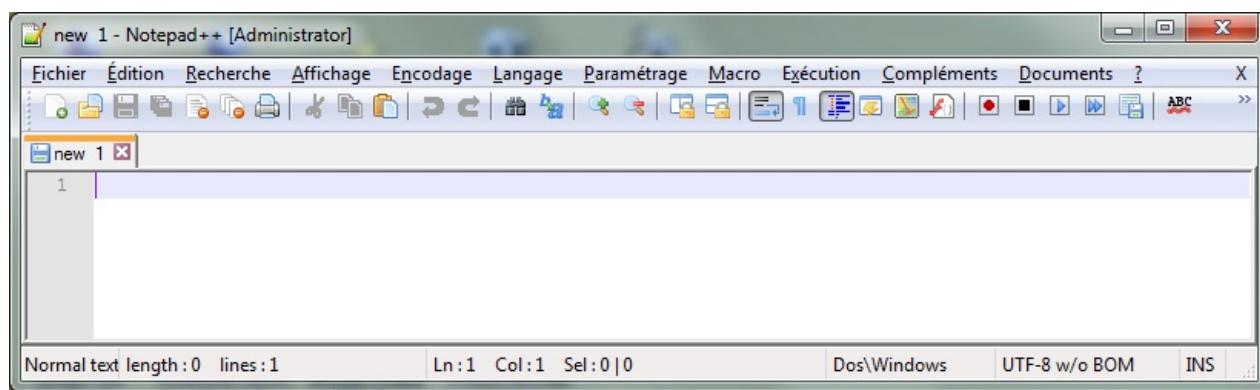
Comment créer un fichier ?

Comment créer un fichier avec notepad ?

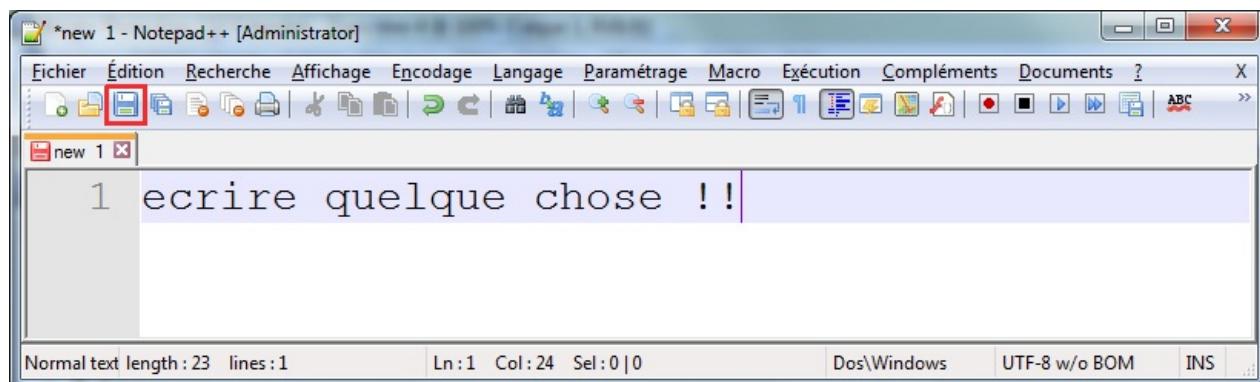
Pour créer un nouveau fichier, vous devez vous rendre dans le logiciel notepad (accessible par le bouton démarrer de windows) :



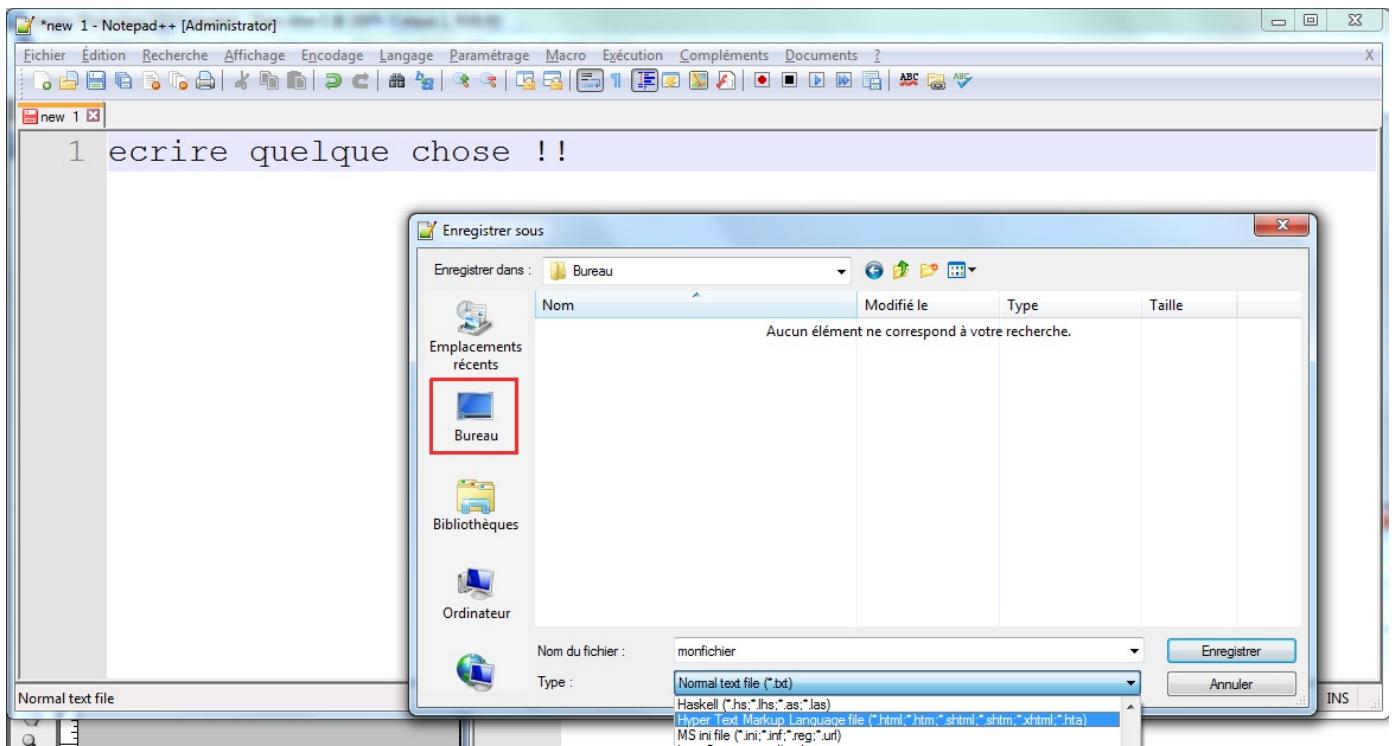
Ensuite vous vous retrouvez dans un nouvel onglet (vous pouvez aussi en ouvrir un nouveau avec **ctrl+n** ou **fichier > nouveau**)



Vous pouvez écrire ce que vous voulez à l'intérieur :



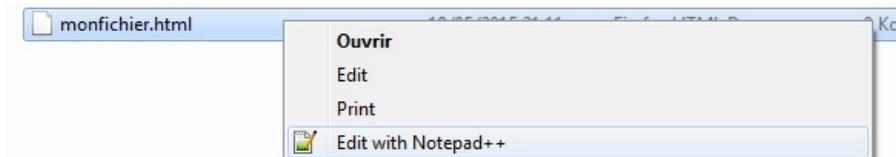
Cliquez sur l'icône en forme de disquette pour enregistrer et choisissez un emplacement pour votre nouveau fichier :



Choisissez l'extension .html afin de créer votre toute première page web !

Vous pouvez double-cliquer sur votre fichier pour le lancer dans le navigateur.

A tout moment, vous pouvez reconsulter et éditer votre fichier :

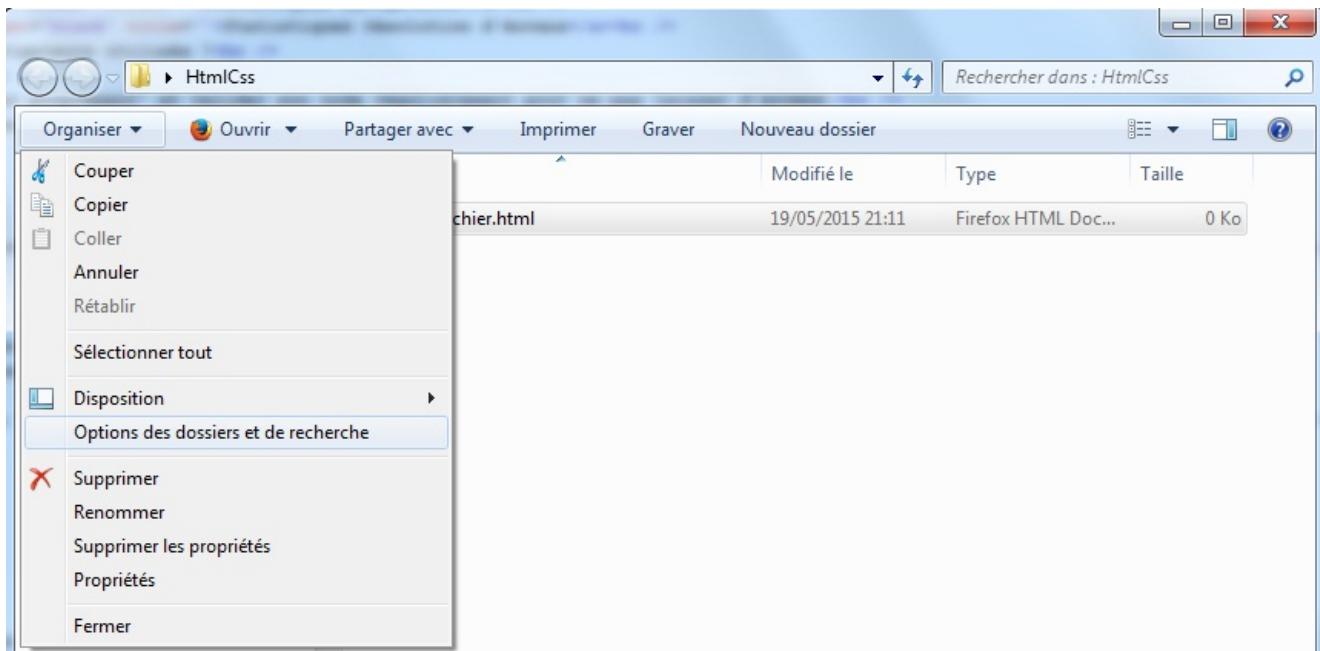


Maintenant que nous pouvons écrire du code dans notepad, il nous faut apprendre la structure d'une page web Html et Css.

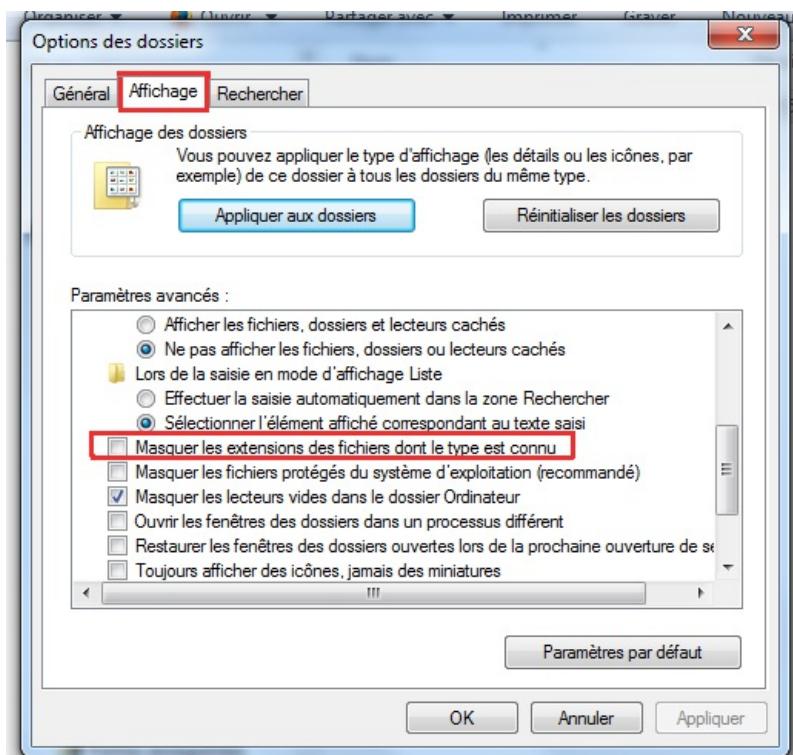
06

Travailler avec les extensions de fichiers visibles

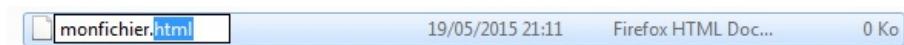
Si vous souhaitez travailler avec les extensions de fichiers visibles (préférable lorsqu'on fait de l'informatique), vous devrez rentrer dans l'un de vos dossiers et cliquer sur le bouton organiser > Options des dossiers et de recherche.



Puis ensuite vous rendre dans l'onglet affichage pour décocher > Masquer les extensions des fichiers dont le type est connu.



Nous apercevons l'extension de vos fichiers :



Structure d'une page Html

Un langage c'est comme une langue, si vous partez en vacances et que vous voulez faire comprendre en Espagne il faudra apprendre et parler Espagnol ! Et bien, si vous souhaitez créer un site web il faudra apprendre et parler html / css. (à la différence que ce sera une langue écrite uniquement)

08

Qu'est-ce qu'un code ?

Un code est un morceau de texte suivant des normes et les conventions d'un langage, introduit à l'intérieur d'un fichier possédant l'extension du langage (exemple : .html, .css, etc.).

l'ensemble de ces lignes de code sont exécutées et lues (*par un navigateur ou l'interpréteur d'un serveur*) et permettent d'obtenir un affichage en retour.

09

Qu'est-ce qu'une balise ?

Une balise est un mot entre chevron, par exemple :

```
<mabalise> </mabalise>
```

?

La balise permettant de contenir des paragraphes s'ouvre avec :

```
<p> </p>
```

?

Cette même balise se ferme avec :

```
</p>
```

Certaines balises sont auto-fermantes (elles ne se ferment pas), par exemple :

```
<mabaliseautofermante>
```

?

Parmis les balises autofermantes nous pouvons noter les suivantes :
 , <hr> , , <meta> , <input> , <link> ...

Qu'est-ce qu'un chevron ?

Un chevron ouvrant c'est ce caractère : <

Un chevron fermant c'est ce caractère : >

10

Qu'est-ce qu'un attribut ?

Un attribut est un mot complémentaire venant se glisser à l'intérieur d'une balise.

Exemple avec l'attribut charset :

```
<meta charset="utf-8">
```

?

La balise Meta, peut servir à plusieurs utilisations...

Nous lui ajoutons donc l'attribut charset pour préciser que nous souhaitons régler l'encodage des caractères sur UTF-8 (*attention il faudra que votre éditeur soit également réglé sur UTF-8 sans bom*).

Un autre exemple avec l'attribut content :



Comme la balise Meta peut servir dans plusieurs situations, nous lui ajoutons donc l'attribut **description** pour préciser que nous souhaitons ajouter une description qui soit prise en compte par Google et les autres moteurs de recherche.

L'attribut **description** de la balise meta permettra de fournir à Google (et aux autres moteurs de recherche) une description.



11

Qu'est-ce qu'une valeur ?

Une valeur est un mot ou un ensemble de mots venant renseigner un attribut.

Dans notre code ci-dessus, les balises apparaissent en bleu, les attributs en blanc, les valeurs en vert. Exemple :



Une valeur est un morceau de texte saisi dans les guillemets d'un attribut. comme ici avec le nom de notre feuille de style css : structure.css.

Cette ligne nous permettra plus tard de lier le fichier Html au fichier Css.

12

Qu'est-ce qu'un commentaire ?

Un commentaire est une portion de texte (souvent une note ou une indication) ignorée par l'interpréteur, exemple :



Les commentaires sont des morceaux de texte ignorés par le navigateur, c'est en quelque sorte des annotations que l'on peut ajouter pour y voir plus clair et se repérer plus rapidement dans un code-source.

Les commentaires sont également très pratiques lors d'un travail d'équipe de manière à ce qu'un nouveau développeur arrivant sur un projet comprenne rapidement avec quelques indications le travail ayant été réalisé.

L'indentation consiste à se décaler (du bord gauche) vers la droite pour indiquer que l'information se trouve à l'intérieur.

```
<p>
  Mon Texte
</p>
```

Dans notre cas « Mon Texte » se trouve à l'intérieur des balises `<p>` et `</p>`.

Il est important de respecter cette méthodologie en entreprise (pour soi, mais aussi lorsque l'on collabore avec plusieurs personnes).

Si vous ne voyez pas l'intérêt de l'indentation, posez-vous cette question : aimeriez-vous que l'on vous partage 200 lignes de code totalement dé-indentées et quasi illisibles ?

Même si cela n'est pas obligatoire pour rendre un code personnel fonctionnel, cela reste hautement recommandé pour améliorer la lecture.

Nous allons créer un premier fichier par l'intermédiaire de notepad++.

Nous nommerons ce fichier :  mapage.html



➤ Informations sur le nommage des fichiers

Les noms de fichiers en informatique ne comportent ni d'espace, ni accent.

Voici le contenu à introduire dans votre fichier :

```
mapage.html
?
<!Doctype html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Ma Page</title>
  </head>
  <body>
    <h1>Mon premier Titre !</h1>
    <p>Et mon premier paragraphe de texte qui l'accompagne !!</p>
  </body>
</html>
```

N'oubliez pas d'enregistrer en sauvegardant votre fichier :

The screenshot shows the Notepad++ interface with the file 'mapage.html' open. The code is as follows:

```
<!Doctype html>
<html>
    <head>
        <meta charset="utf-8" />
        <title>Ma Page</title>
    </head>
    <body>

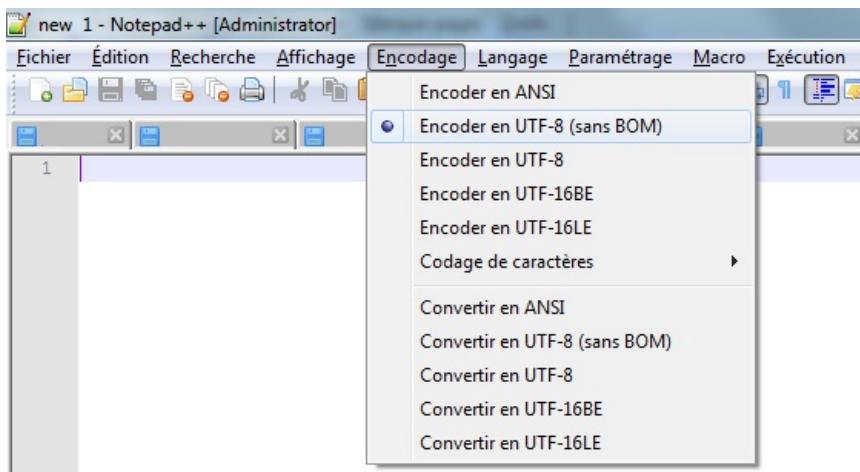
        <h1>Mon premier Titre !</h1>
        <p>Et mon premier paragraphe de texte qui l'accompagne !!</p>

    </body>
</html>
```

Décryptons ensemble les données saisies :

- **Doctype** est une balise obligatoire en début de document et cela peut aussi permettre de préciser son type ainsi que la version du langage Html utilisée. C'est également indispensable pour valider son code source auprès de l'organisation W3C.
- **Html** est également une balise indispensable, elle doit être présente dans tous vos documents pour englober le code de vos pages web.
- **Head** définit les entêtes de votre document, c'est une partie non visible mais nécessaire pour plusieurs raisons. Par exemple, nous pourrons définir un titre de page, se lier avec une feuille de style css ou encore préciser la description de notre site web.
- **Title** permet de donner un titre à votre page, c'est ce que vous verrez apparaître sur l'onglet de votre navigateur
- **body** définit, cette fois, le corps de votre document, c'est la partie visible de votre page web, c'est à cet endroit que nous mettrons du texte, des images, et toute autre information.
- **h1** est une balise qui permet de déterminer le titre du document (en général, nous ne mettons qu'un seul titre par page, par convention liée au référencement).
- **p** est une balise qui permet de créer un paragraphe de texte du document (nous pouvons en mettre autant qu'on le souhaite !).

Les fichiers créés sous notepad doivent posséder l'encodage utf-8 sans bom :



Vous devrez ajouter la balise suivante dans la partie <head> pour éviter des problèmes d'encodage :

```
<meta charset="utf-8">
```

16

Structure d'une page web ! Le code Html

Nous allons créer un nouveau dossier (sans espace, sans accent) nommé : /HtmlCss/

A l'intérieur du dossier /HtmlCss/, nous allons créer un nouveau dossier nommé : /Structure/

Nous allons étudier le code source d'une page web standard

Pour cela, créer un fichier nommé : structure.html

```
structure.html 15/04/2015 14:34 Firefox HTML Doc...
```

Nous travaillerons donc sur le fichier et le chemin suivant : /HtmlCss/Structure/structure.html

```
structure.html

<!Doctype html>
<html>
  <head>
    <title>Ma Page</title>
    <link rel="stylesheet" href="structure.css">
    <meta charset="utf-8">
    <meta name="keywords" content="motclé1 motclé2">
    <meta name="description" content="la description du site web">
    <meta name="author" content="Nom Prenom">
  </head>
  <body>

    <header>
    </header>

    <nav>
    </nav>

    <section>
      <div>
        <span>
          </span>
      </div>
    </section>

    <article>
    </article>

    <footer>
```

```
</footer>  
</body>  
</html>
```

Comme vous pouvez le constater, l'indentation a été respectée

Décryptons ensemble les données saisies :

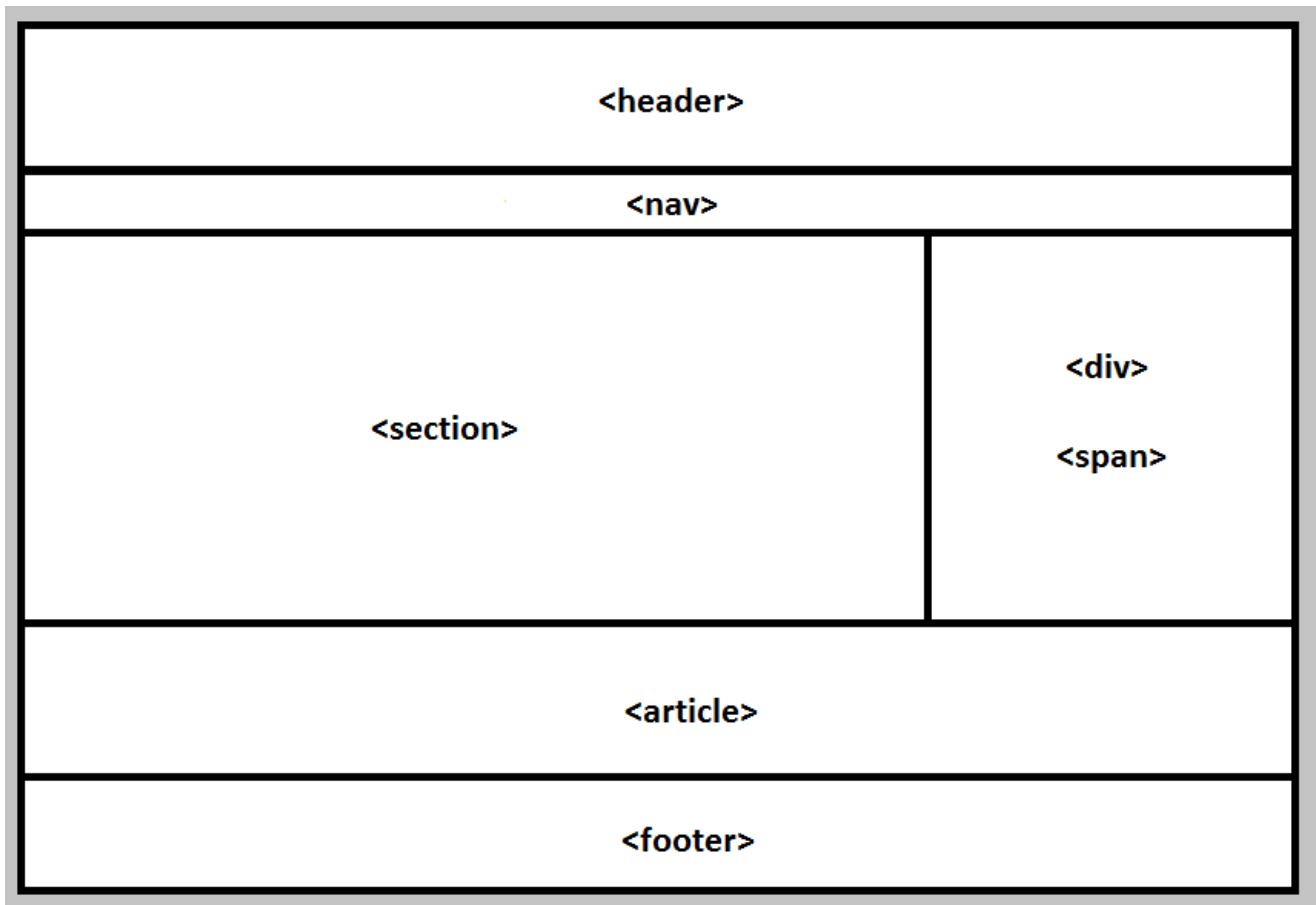
- **Doctype** est une balise obligatoire pour déclarer un document et pour plus tard valider son code sur W3C.
- **Html** est également une balise indispensable, elle doit être présente dans tous vos documents pour englober le code de vos pages web.
- **Head** définit les entêtes de votre document, c'est une partie non visible mais nécessaire pour plusieurs raisons. Par exemple, nous pourrons définir un titre de page, se lier avec une feuille de style css ou encore préciser la description de notre site web.
- **Title** permet de donner un titre à votre page, c'est ce que vous verrez apparaître sur l'onglet de votre navigateur
- **Link** permet de me relier à une feuille de style css (je pourrais écrire du code css à l'intérieur pour mettre en forme ma page web)
- **Meta** permet de donner des informations sur son site web comme l'encodage utilisé, une liste de mots clés décrivant votre site web, l'auteur du site, une description (la description est utilisée par Google, c'est ce qu'on peut apercevoir sous l'adresse url d'un site web dans le moteur de recherche).

Meta description :



- **Body** définit le corps de votre document, c'est la partie visible de votre page web, c'est à cet endroit que nous mettrons du texte, des images, et toute autre information.
- **Header** est une balise permettant de prévoir un emplacement (ou une zone) pour le haut de votre site web. En général, nous y retrouvons un logo, un slogan ...
- **Nav** est une balise permettant de prévoir un emplacement (ou une zone) pour votre menu de navigation. En général, nous y retrouvons des liens.
- **Section** est une balise permettant de prévoir un emplacement (ou une zone) pour le contenu de votre site web. Nous y retrouverons plutôt un titre, du texte, des images, etc.
- **Div** est une balise permettant de prévoir un emplacement (ou une zone) quelconque de votre site web. L'élément div peut regrouper tout et n'importe quoi, alors que l'élément section a plutôt été pensé pour regrouper du contenu apparenté.
- **Span** est une balise permettant de prévoir un petit emplacement secondaire, on l'utilise souvent à l'intérieur d'une autre zone. L'élément span peut regrouper tout et n'importe quoi, il s'agit d'une zone secondaire d'information
- **Article** est une balise permettant de prévoir une zone de texte.
- **Footer** est une balise permettant de prévoir un emplacement (ou une zone) pour le bas de votre site web. En général, nous y retrouvons des liens administratifs (type: mentions légales, plan du site, Conditions Générales, etc), un copyright, etc.

Voici ce que ces balises pourraient donner lors d'un affichage :



➤ Bon à savoir

Lorsque l'on clique sur l'une de nos balises dans notepad++, les balises passent en surbrillance ce qui prouve qu'elles sont bien ouvertes et bien fermées, c'est un réflexe à prendre pour être sûr d'avoir un code-source et une sémantique de qualité.

```

1  <!Doctype html>
2  <html>
3      <head>
4          <title>Ma Page</title>
5          <link rel="stylesheet" href="structure.css">
6          <meta charset="utf-8" />
7          <meta name="keywords" content="motclé1 motclé2">
8          <meta name="description" content="la description du site web">
9          <meta name="author" content="Nom Prenom">
10     </head>
11     <body>
12         <header>
13             ...
14         </header>
15
16         <nav>
17             ...
18         </nav>
19
20         <section>
21             <div>
22                 <span>
23                     ...
24                 </span>
25             </div>
26         </section>
27
28         <article>
29             ...
30         </article>
31
32         <footer>
33             ...
34         </footer>
35     </body>
36 </html>

```

L'ordre de saisie des balises détermine l'ordre d'affichage.

17

Assurer la liaison entre une page html et une feuille de style css

Il n'est pas courant d'écrire un fichier Html sans se lier à une feuille de style Css.

Le code Html détermine la structure de la page (incluant les différentes zones), mais nous devons également écrire du code CSS pour positionner ces zones sur la page, leur donner des tailles, des couleurs, etc.

Reprends les 2 fichiers précédents : structure.html et structure.css

Nous attribuons le même nom aux 2 fichiers toutefois les fichiers pourraient se nommer différemment

Cette ligne (à placer dans le code HTML) vous permettra d'assurer la liaison avec votre feuille de style (fichier css) :

```
<link rel="stylesheet" href="structure.css">
```

Comment être sûr d'être correctement relié à votre feuille de style ?

Et bien, si vous écrivez du code CSS de mise en forme et que vous ne voyez pas votre style sur votre page html, c'est sans doute qu'il y a un problème dans le code ou dans le chemin qui relie les deux fichiers.

Vous pouvez aussi effectuer un clic droit > code source de la page (**ctrl + u** au clavier) et cliquer sur le chemin du fichier « .css » qui est affiché.

```
1 <!Doctype html>
2 <html>
3   <head>
4     <title>Ma Page</title>
5     <link rel="stylesheet" href="structure.css">
6     <meta charset="utf-8" />
7     <meta name="keywords" content="motclé1 motclé2">
8     <meta name="description" content="la description du site web">
9     <meta name="author" content="Nom Prenom">
10   </head>
11   <body>
12     <header>
```

Lorsque vous cliquerez dessus, deux résultats sont possibles :

➤ 1. Soit, vous pouvez observer votre code CSS apparaître et vous savez que votre fichier est bien relié.

Si un problème persiste, cela sera certainement lié au code en lui-même mais pas à la liaison entre les fichiers. Pensez à relire le code html tout comme le code css, vérifiez que vos deux fichiers soient bien enregistrés et que la liaison entre html et css soit bien établie (cela peut aussi être une erreur de chemin de fichier).

➤ 2. Soit, une page d'erreur s'affiche :



Sur un site web static, généralement une seule feuille de style est utilisée.

Un seul code Html peut être personnalisé différemment selon les feuilles de style, voici l'exemple du [zen garden](#).

Vous pourrez afficher le site du zen garden sous différentes couleurs et présentations, le code Html est toujours le même.

C:\wamp\www\dev\tutomooc\web\htmlcss\test1.html - Notepad++ [Administrator]

Fichier Édition Recherche Affichage Encodage Langage Paramétrage Macro Exécution Compléments Documents ?

Exécuter... F5

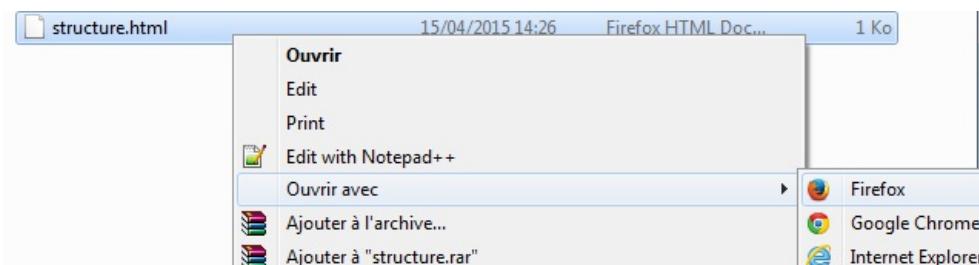
- Launch in Firefox Ctrl+Alt+Shift+X
- Launch in IE Ctrl+Alt+Shift+I
- Launch in Chrome Ctrl+Alt+Shift+R
- Launch in Safari Ctrl+Alt+Shift+F
- Get php help Alt+F1
- Google Search Alt+F2
- Wikipedia Search Alt+F3
- Open file Alt+F5
- Open in another instance Alt+F6
- Open containing folder
- Open current dir cmd
- Send via Outlook Ctrl+Alt+Shift+O
- Modifier/Supprimer Commande...

```

1 <!DOCTYPE html>
2 <html>
3   <head>
4     <title>Ma Page</title>
5     <link rel="stylesheet" href="structure.css">
6     <meta charset="utf-8" />
7     <meta name="keywords" content="motclé motclé">
8     <meta name="description" content="la description">
9     <meta name="author" content="Nom Prenom">
10    </head>
11    <body>
12      <header>
13        Header
14      </header>
15
16      <nav>
17        Nav
18      </nav>
19
20      <section>
21        <div>
22          <span>
23            Section
24          </span>
25        </div>
26      </section>
27
28      <article>
29        Article
30      </article>
31
32      <footer>
33        Footer
34      </footer>
35    </body>
36  </html>

```

Un autre moyen consiste à aller sur l'emplacement de votre fichier pour effectuer un double clic gauche ou un clic droit > ouvrir avec > firefox (ou un autre navigateur).



Le Résultat

/htmlcss/test1.html

Désactiver Cookies CSS Formulaires Images Infos Divers Entourer Fenêtre Outils Code Options

Header
Nav
Section
Article
Footer



➤ Question

J'ai une page blanche à l'écran, avec presque rien !!! c'est normal ?

Oui, rassurez-vous, c'est tout à fait normal :p ! nous n'avons pas encore écrit le code CSS.

Les emplacements et les zones sont bien déclarés mais il faut ajouter une mise en forme afin de les voir apparaître.



➤ Bon à savoir

Le HTML fonctionne la plupart du temps avec le CSS. La combinaison de ces deux langages nous permettent de construire des pages web.

N'oubliez pas d'actualiser votre navigateur (`f5` ou `ctrl + r`) pour rafraîchir votre page.

19

Observer le code-source de la page pour trouver ses erreurs

Avant d'aller plus loin, vous pouvez observer le code-source de la page, vous pouvez effectuer un clic droit > code source de la page dans votre navigateur.

Ou directement avec le raccourci, au clavier : `ctrl + u`.

Regarder votre code-source est très utile, s'il y a une erreur de sémantique, celle-ci apparaît généralement en rouge.

Exemple dans le code source :

```
1 <!Doctype html>
2 <html>
3     <head>
4         <title>Ma Page</title>
5         <link rel="stylesheet" href="structure.css">
6         <meta charset="utf-8" />
7         <meta name="keywords" content="motclé1 motclé2">
8         <meta name="description" content="la description du site web">
9         <meta name="author" content="Nom Prenom">
10    </head>   « < » rencontré alors qu'un nom d'attribut était attendu. Cause probable : « > »
11    <body manquant immédiatement avant.
12        <header>
13            Header
14        </header>
15
16        <nav>
17            Nav
18        </nav>
19
20        <section>
21            <div>
22                <span>
23                    Section
24                </span>
25            </div>
26        </section>
27
28        <article>
29            Article
30        </article>
31
32        <footer>
33            Footer
34        </footer>
35    </body>
36 </html>
```

Nous pouvons constater que la balise body apparaît en rouge et que par conséquent il y a une erreur sur cette ligne !

Exemple dans l'éditeur :

```

1  <!Doctype html>
2  <html>
3      <head>
4          <title>Ma Page</title>
5          <link rel="stylesheet" href="structure.css">
6          <meta charset="utf-8" />
7          <meta name="keywords" content="motclé1 motclé2">
8          <meta name="description" content="la description du site web">
9          <meta name="author" content="Nom Prenom">
10     </head>
11     <body>
12         <header>
13             Header
14         </header>
15
16         <nav>
17             Nav
18         </nav>
19
20         <section>
21             <div>
22                 <span>
23                     Section
24                 </span>
25             </div>
26         </section>
27
28         <article>
29             Article
30         </article>
31
32         <footer>
33             Footer
34         </footer>
35     </body>
36 </html>

```

Un autre indicateur d'erreur : La coloration syntaxique.

Sous notepad les balises apparaissent en bleu, les attributs en rouge, les valeurs en violet. Si les couleurs ne sont pas bonnes, cela peut être révélateur d'une erreur dans le code.

Nous voyons que la balise `header` n'a pas sa couleur habituelle, c'est donc qu'une erreur a été faite avant la ligne n°12.

Vous avez trouvé l'erreur ? et oui, il s'agit du chevron fermant sur la balise body !

D'une manière générale, pensez donc à toujours vérifier votre code-source pour voir si rien n'apparait en rouge avant d'aller plus loin.

Pour cela, il faut se rendre sur Google et saisir : w3c validator.

Nous avons 3 types de validation possibles :

- Par l'url
- Par un upload de fichier
- Par une saisie (copier/coller)

Etant donné que notre site n'est pas en ligne, nous allons éliminer le 1er choix et je vous propose de copier/coller votre code dans le 3e onglet, comme ceci :

The screenshot shows the W3C Markup Validation Service interface. The 'Validate by Direct Input' tab is active. In the main area, there is a text input field containing an HTML document. A red arrow points to this input field. Below the input field is a 'Check' button, which is also highlighted with a red box.

Excellent ! Notre document est valide et conforme et aucune erreur n'est relevée !

The screenshot shows the validation results for the pasted HTML code. A green banner at the top says "This document was successfully checked as HTML5!". The "Result" section shows "Passed". The "Source" section displays the same HTML code as the input field in the previous screenshot. The "Encoding" section shows "utf-8", "detect automatically". The "Doctype" section shows "HTML5", "detect automatically". The "Root Element" section shows "html".

Lorsqu'on crée un site web, c'est pour le mettre en ligne (sur le web) à terme.

En effet, si votre site web reste en local sur votre ordinateur, il risque de ne pas servir énormément puisque personne ne le verra à par vous même !

Lorsque l'on crée un site web, il y a toujours une page d'Accueil (Home Page, page par défaut), pour cela nous créons un fichier qui se nomme : index.html

En effet, c'est la première ressource que va rechercher le serveur lorsque vous accéderez à un site web. Gardez à l'esprit que si vous souhaitez créer une page d'accueil il faudra appeler le fichier index.html

Structure d'un fichier css

22

Structure d'un fichier CSS

Nous allons créer (*ou éditer pour ceux qui l'ont déjà créé*) notre fichier structure.css qui doit se trouver au même endroit (dans le même dossier) que le fichier structure.html pour démarrer !

Pour rappel, nous travaillons à l'intérieur du dossier suivant : /HtmlCss/Structure/structure.css

structure.css	15/04/2015 14:43	Fichier CSS
structure.html	15/04/2015 14:34	Firefox HTML Doc...

Dans notre exemple ci-dessous, nous pouvons apercevoir que le code s'écrit entre des accolades {}, les selecteurs apparaissent en blanc, les propriétés en bleu et les valeurs en vert.

The screenshot shows a code editor window titled "structure.css". The code contains various CSS rules for different HTML elements:

```
/* commentaire en css */
header{
height: 100px;
background: #f98f32;
}
nav{
height: 30px;
background: #20DDF2;
}
section{
min-height: 300px;
background: #F2F220;
}
article{
min-height: 200px;
background: #F2F220;
}
footer{
height: 100px;
background: #f98f32;
}
```

Décryptons ensemble les données saisies :

- **header** cela va permettre de personnaliser et mettre en forme la balise `header` présente dans le fichier structure.html

Nous lui ajoutons la propriété `height: 100px;` cela permet de dire que la hauteur de cette zone fera précisément 100 pixels.

La propriété `background: #f98f32;` (orange) permet de lui attribuer une couleur de fond afin de la voir apparaître.

- **nav** cela va nous permettre de personnaliser et mettre en forme la balise `nav` présente dans le fichier structure.html

`Height` : Hauteur fixe

`background` : Fond.

- **section** cela va nous permettre de personnaliser et mettre en forme la balise `section` présente dans le fichier structure.html

La propriété `min-height` permet de fixer une hauteur minimum. C'est à dire qu'en l'état, la zone `section` est prévue pour faire 300 pixels de hauteur, mais si jamais du texte, image ou tout autre type de contenu est ajouté à l'intérieur, cette zone pourra faire 300, 400, 500, voir 1000 pixels de hauteur et bien plus ! cela sera régi par le contenu à l'intérieur de la zone (et non plus par notre limite d'origine).

`background` : Fond.

- **article** cela va nous permettre de personnaliser et mettre en forme la balise `article` présente dans le fichier structure.html

`min-height` : Hauteur minimum

`background` : Fond.

- footer cela va nous permettre de personnaliser et mettre en forme la balise `footer` présente dans le fichier `structure.html`

height : Hauteur fixe

background : Fond.



➤ Bon à savoir

1 pixel est une unité de mesure.

1 code couleur s'écrit dans les lettres de A à F et numéro de 0 à 9 (composé de 3 ou 6 caractères (chiffre ou lettre, chiffre et lettre)).



➤ Attention

Votre fichier doit absolument être enregistré (vous pouvez vous fier à l'icone représentant une disquette (bleu = enregistré, rouge = non-enregistré)).

```

structure.html x structures.css x
1 header{
2   height: 100px;
3   background: #f98f32;
4 }
5 nav{
6   height: 30px;
7   background: #20DDF2;
8 }
9 section{
10  min-height: 300px;
11  background: #F2F220;
12 }
13 article{
14  min-height: 200px;
15  background: #F2F220;
16 }
17 footer{
18   height: 100px;
19   background: #f98f32;
20 }

```

23

Qu'est-ce qu'un selecteur en css ?

Un selecteur en CSS est un mot reconnu à la lecture du navigateur. Ce mot doit être présent dans le fichier Html pour que la liaison soit faite.

Exemple avec un selecteur de balise :

structure.html :

```
<article>
</article>
```

structure.css :

```
article{ }
```

24

Les commentaires en css

Afin de mettre des annotations dans votre code, vous pouvez utiliser les commentaires en CSS de la manière suivante :

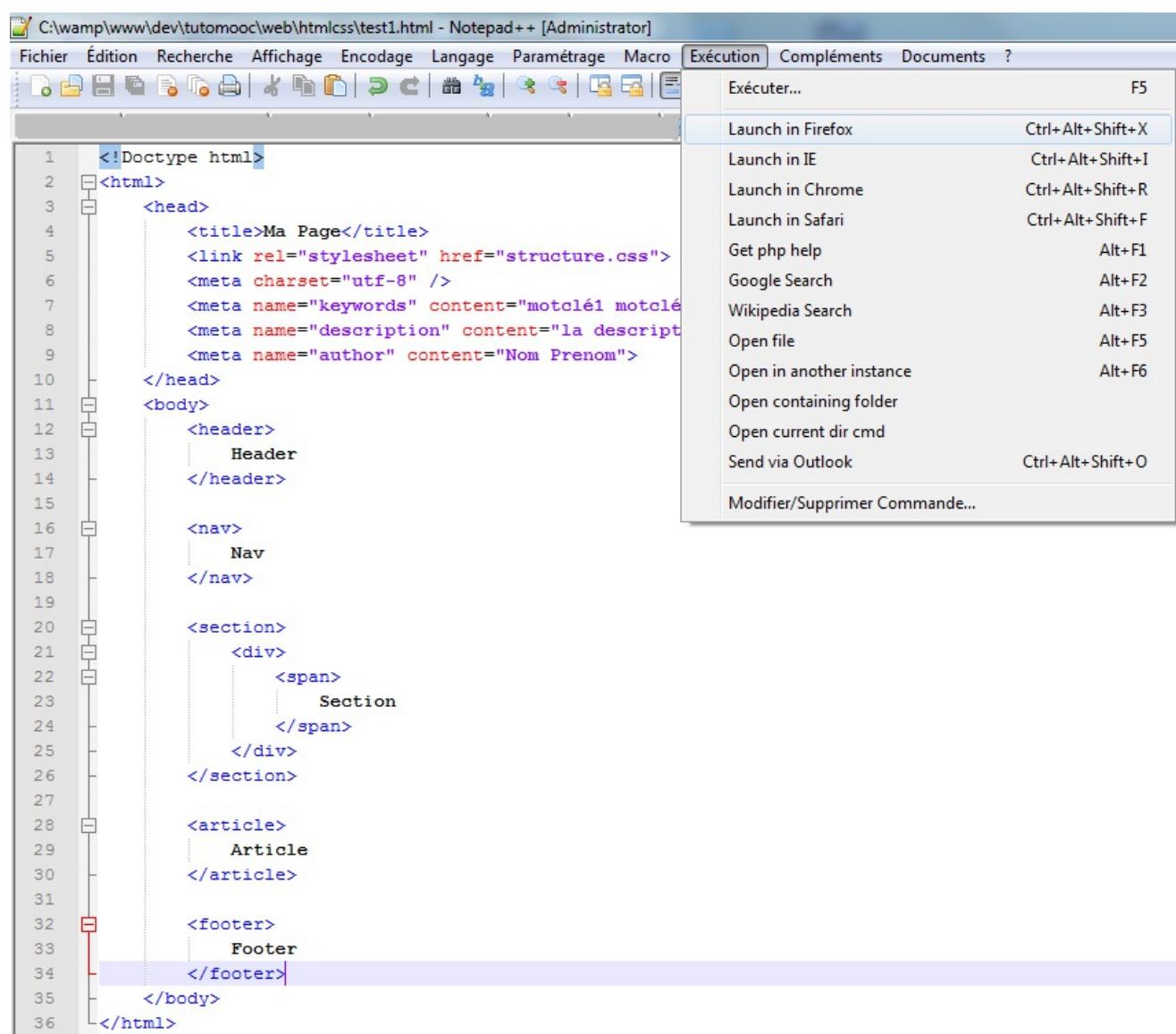
```
/* commentaire en CSS */
```

L'ordre de saisie n'a pas d'importance dans le fichier CSS.

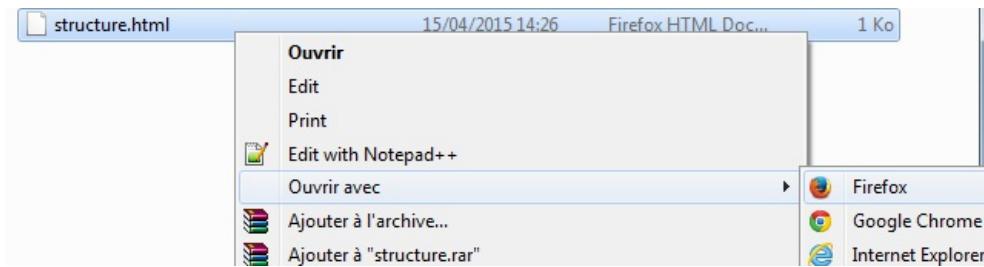
25

Observons le rendu à l'écran dans le navigateur !

Pour observer le rendu de votre page web dans votre navigateur, vous pouvez toujours passer par l'éditeur notepad :



Vous pouvez aussi vous rendre directement sur l'emplacement de votre fichier pour effectuer un double clic gauche ou un clic droit > ouvrir avec > firefox (ou un autre navigateur).



Voici le rendu que vous devriez avoir :

Le Résultat



Nous pouvons remarquer que le Html et Css sont complémentaires et fonctionnent conjointement ensemble !

Question : « c'est moche ta page multicolore ! »

Réponse : nous ne cherchons pas à faire beau pour un premier exemple mais à faire quelque chose de compréhensible.

Question : « je ne comprends pas à quoi ça sert »

Réponse : Et bien, nous colorons (via le css) les zones que nous avons en notre possession (via le html) pour les faire apparaître et pouvoir travailler à l'intérieur.

Question : « ça ne ressemble pas à un site web »

Réponse : Détrompez-vous ! La plupart des sites web sont construits de cette manière, c'est à dire que nous commençons par colorer les zones pour mieux les voir et travailler à l'intérieur, et seulement à la fin nous les "décolorons" pour le rendu graphique.

Un site c'est avant tout une série d'emplacements et de zones : un zoning !

Editer le Html et Css conjoitement

Vous pouvez mettre votre code html et votre code css « côté à côté » sur l'éditeur notepad :

The screenshot shows the Notepad++ interface with two tabs open. The left tab, titled 'structures.css', contains the following CSS code:

```

1 header{
2     height: 100px;
3     background: #f98f32;
4 }
5 nav{
6     height: 30px;
7     background: #20DDF2;
8 }
9 section{
10    min-height: 300px;
11    background: #F2F220;
12 }
13 article{
14    min-height: 200px;
15    background: #F2F220;
16 }
17 footer{
18     height: 100px;
19     background: #f98f32;
20 }

```

The right tab, titled 'structure.html', contains the following HTML code:

```

1 <!DOCTYPE html>
2 <html>
3   <head>
4     <title>Ma Page</title>
5     <link rel="stylesheet" href="structure.css">
6     <meta charset="utf-8" />
7     <meta name="keywords" content="mot clé1 mot clé2">
8     <meta name="description" content="la description du site web">
9     <meta name="author" content="Nom Prenom">
10   </head>
11   <body>
12     <header>
13       Header
14     </header>
15
16     <nav>
17       Nav
18     </nav>
19
20     <section>
21       <div>
22         <span>
23           Section
24         </span>
25       </div>
26     </section>
27
28     <article>
29       Article
30     </article>
31
32     <footer>
33       Footer
34     </footer>
35
36   </body>

```

Pour avoir ces 2 onglets « côté à côté », vous devez effectuer un clic droit sur l'un des deux onglets et cliquer sur « Déplacer vers l'autre vue ».

Les raccourcis

Vous êtes fan des raccourcis clavier ? Vous trouvez cela plus rapide ? voici une liste des raccourcis clavier pour windows, notepad et votre navigateur qui pourront vous rendre service :

NOTE PAD	
Raccourci Clavier	Description
Ctrl+A	permet de sélectionner tout le texte/code
Ctrl+X	permet de couper une partie de texte
Ctrl+C	permet de copier une partie de texte
Ctrl+V	permet de coller une partie de texte
Ctrl+D	permet de dupliquer une ligne
Ctrl+T	permet d'inverser la position d'une ligne avec une autre
Ctrl+Z	permet d'annuler la dernière action
Ctrl+Y	permet de rétablir la dernière action annulée (par le Ctrl+Z)
Ctrl+S	permet d'enregistrer le fichier

Ctrl+G	permet de se déplacer à un numéro de ligne précis (sans avoir à scrollier)
Ctrl+Q (ou Ctrl+Shift+Q)	permet de mettre en commentaire une partie du fichier
Ctrl+F	permet de rechercher à l'intérieur d'un ou plusieurs fichiers/dossiers
Ctrl+H	permet de remplacer un morceau de texte par un autre
Ctrl+L	permet de supprimer une ligne
Ctrl+W	permet de fermer la fenêtre ou l'onglet actuellement ouvert
Ctrl+P	permet d'imprimer le fichier
Ctrl+O	permet d'ouvrir un nouveau fichier
Ctrl+N	permet de créer un nouveau fichier
Ctrl+Tab	permet de se déplacer dans les différents onglets ouverts
Ctrl+espace	permet d'avoir des suggestions : l'auto-complétion !
Ctrl+Alt+Shift+X	permet d'exécuter le code dans firefox
Ctrl+Fleche du haut	permet de scrollier sans changer de ligne (et sans la souris)

WINDOWS	
Raccourci Clavier	Description
Alt+Tab	permet de se déplacer dans les différentes fenêtres ouvertes
Windows+M	permet de revenir sur le bureau
Alt+F4	permet de fermer une fenêtre ouverte
Ctrl+N	Lorsqu'un dossier est ouvert et affiché, cela permet d'ouvrir une nouvelle fenêtre au même emplacement
Ctrl+ double clic	permet d'ouvrir un dossier dans une nouvelle fenêtre
Widows+Fleche	permet de positionner les fenêtres de manière plus précise (sous windows seven)

NAVIGATEUR (FireFox, GoogleChrome, InternetExplorer, Opera, Safari)	
Raccourci Clavier	Description
Ctrl+R F5	permet d'actualiser (rafraîchir) la page
F6+Entrer	permet de revenir sur la page
Ctrl+shift+suppr	permet de vider ou supprimer des données : cookie, cache, etc.
Ctrl+U	permet d'accéder au code-source la page
Ctrl+J	permet d'afficher les téléchargements

Les polices d'écriture

Nous allons créer deux nouveaux fichiers : `texte.html` et `texte.css`

Ces deux nouveaux fichiers seront placés dans notre dossier `/HtmlCss/02-Structure/`.

Ce qui donnera : `/HtmlCss/02-Structure/texte.html` et `/HtmlCss/02-Structure/texte.css`

Voici le contenu de texte.html :

```
texte.html

<!Doctype html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Police css3</title>
    <link rel="stylesheet" href="texte.css">
  </head>
  <body>
    <h1>Mon Titre</h1><!-- Titre principal -->
    <h2>Autre Titre</h2> <!-- SousTitre -->
    <p>Le Lorem Ipsum <strong>est simplement du faux texte</strong> employé dans la composition et la mise en page avant impression.<br>Le
    Lorem Ipsum est le faux texte standard de l'imprimerie depuis les années 1500, <cite>quand un peintre anonyme assembla ensemble des
    morceaux de texte</cite> pour réaliser un livre spécimen de polices de texte.<br>Il n'a pas fait que survivre cinq siècles, mais s'est aussi
    <strike>adapté à la bureautique</strike> informatique, sans que son contenu n'en soit modifié.<br>Il a été popularisé dans les années 1960 grâce
    à la vente de feuilles Letraset contenant des passages du Lorem Ipsum, et, plus récemment, par son inclusion dans des applications de mise
    en page de texte, comme Aldus PageMaker.<br><br>
  </body>
</html>
```

Décryptons ensemble les données saisies :

- **h1** Titre Principal.
- **h2** Sous Titre.
- **p** la balise P permet de déclarer un paragraphe pour y introduire du texte.
- **br** passage à la ligne
- **strong** permet de mettre en gras une partie d'un texte.
- **strike** permet de barrer une partie d'un texte (utile pour les soldes sur les sites ecommerce lorsque les prix sont en promotion).
- **cite** permet d'introduire une citation (italique).

Pour les titres, nous pouvons nous servir des balises suivantes : **h1, h2, h3, h4, h5, h6** (du plus grand au plus petit).

Pour observer votre page, vous pouvez cliquer sur le fichier Html puis vous rendre dans Exécution > Lancer dans FireFox ou effectuer un clic droit sur votre fichier, ouvrir avec > firefox.

Dans le navigateur, cela devrait vous donner le résultat suivant :

Le Résultat

Mon Titre

Autre Titre

Le Lorem Ipsum est **simplement du faux texte** employé dans la composition et la mise en page avant impression.

Le Lorem Ipsum est le faux texte standard de l'imprimerie depuis les années 1500, quand un peintre anonyme assembla ensemble des morceaux de texte pour réaliser un livre spécimen de polices de texte.

Il n'a pas fait que survivre cinq siècles, mais s'est aussi adapté à la bureautique informatique, sans que son contenu n'en soit modifié.

Il a été popularisé dans les années 1960 grâce à la vente de feuilles Letraset contenant des passages du Lorem Ipsum, et, plus récemment, par son inclusion dans des applications de mise en page de texte, comme Aldus PageMaker.

On sait depuis longtemps que travailler avec du texte lisible et contenant du sens est source de distractions, et empêche de se concentrer sur la mise en page elle-même. L'avantage du Lorem Ipsum sur un texte générique comme 'Du texte. Du texte. Du texte.' est qu'il possède une distribution de lettres plus ou moins normale, et en tout cas comparable avec celle du français standard. De nombreuses suites logicielles de mise en page ou éditeurs de sites Web ont fait du Lorem Ipsum leur faux texte par défaut, et une recherche pour 'Lorem Ipsum' vous conduira vers de nombreux sites qui n'en sont encore qu'à leur phase de construction. Plusieurs versions sont apparues avec le temps, parfois par accident, souvent intentionnellement (histoire d'y rajouter de petits clins d'œil, voire des phrases embarrassantes).

— La Légende

Le texte ...

Pour le moment cette police d'écriture est classique, créons un nouveau fichier : **texte.css**.

Ecrivons le fichier **texte.css** ensemble :



```
texte.css
body{
font-family: verdana;
}
```

Dévryptions ensemble le code css ci-dessus :

- **font-family** est une propriété permettant de préciser la police d'écriture à utiliser.

Vous devriez voir le changement de police d'écriture à l'écran.

29

Comment importer une police d'écriture ?

Pour importer une nouvelle police d'écriture, il vous faut d'abord le fichier contenant la bonne police.

Pour cela, trois solutions :

- 1 - Créer vous même votre police sous Illustrator/Photoshop.
- 2 - Récupérer en locale une police déjà existante sur des sites comme : [Google Font](#) - [DaFont](#) - [Fonts For Web](#)
- 3 - Utiliser une police déjà existante en ligne comme : [Google Font](#)

30

Importer une police d'écriture distante

Nous allons créer deux nouveaux fichiers : **texte2.html** et **texte2.css**

Ces deux nouveaux fichiers seront placés dans notre dossier **/HtmlCss/02-Structure/**.

Ce qui donnera : **/HtmlCss/02-Structure/texte2.html** et **/HtmlCss/02-Structure/texte2.css**

Voici le contenu de **texte2.html** :

```
texte2.html

<!Doctype html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Police css3</title>
    <link rel="stylesheet" href="texte2.css">
  </head>
  <body>
    <h1>Mon Titre</h1><!-- Titre principal -->
    <h2>Autre Titre</h2> <!-- SousTitre -->
    <p>Le Lorem Ipsum est simplement du faux texte</strong> employé dans la composition et la mise en page avant impression.<br>Le Lorem Ipsum est le faux texte standard de l'imprimerie depuis les années 1500, <cite>quand un peintre anonyme assembla ensemble des morceaux de texte</cite> pour réaliser un livre spécimen de polices de texte.<br>Il n'a pas fait que survivre cinq siècles, mais s'est aussi <strike>adapté à la bureautique</strike> informatique, sans que son contenu n'en soit modifié.<br>Il a été popularisé dans les années 1960 grâce à la vente de feuilles Letraset contenant des passages du Lorem Ipsum, et, plus récemment, par son inclusion dans des applications de mise en page de texte, comme Aldus PageMaker.<br><br>
  </body>
</html>
```

Pour intégrer une nouvelle police d'écriture distante (hébergée sur le web), il vous faut d'abord l'adresse url contenant la bonne police.

Editons notre nouveau fichier : **texte2.css** :

```
texte2.css

/* utilisation d'une police css3 à l'aide d'une url */
@font-face {
  font-family: 'Cabin';
  font-style: normal;
  font-weight: 400;
  src: local('Cabin Regular'), local('Cabin-Regular'), url(https://fonts.gstatic.com/s/cabin/v8/yQOMOX-hR0-6LTD879t-PQ.woff2) format('woff2');
  unicode-range: U+0000-00FF, U+0131, U+0152-0153, U+02C6, U+02DA, U+02DC, U+2000-206F, U+2074, U+20AC, U+2212, U+2215, U+E0FF,
  U+EFFD, U+F000;
}

body{
  font-family: "Cabin";
}
h1{
  text-align: center; /* alignement du texte */
  color: #336699; /* couleur */
  letter-spacing: 3px; /* espacement entre les lettres */
  font-size: 50px; /* taille */
}
```

Dévryptons ensemble le code css ci-dessus :

- **@font-face** est une règle utilisée dans la dernière version du langage CSS (la version 3), c'est la raison pour laquelle vous entendrez souvent parler de police css3

La police Cabin utilisée dans l'exemple ci-dessus est hébergée par Google, l'avantage c'est que c'est plus simple à écrire, l'inconvénient c'est que cela peut être plus long à charger (dans la mesure de quelques centièmes de seconde) puisqu'on se connecte à un autre site.

Le Résultat

Mon Titre

Autre Titre

Le Lorem Ipsum **est simplement du faux texte** employé dans la composition et la mise en page avant impression.

Le Lorem Ipsum est le faux texte standard de l'imprimerie depuis les années 1500, quand un peintre anonyme assembla ensemble des morceaux de texte pour réaliser un livre spécimen de polices de texte.

Il n'a pas fait que survivre cinq siècles, mais s'est aussi adapté à la bureautique informatique, sans que son contenu n'en soit modifié.

Il a été popularisé dans les années 1960 grâce à la vente de feuilles Letraset contenant des passages du Lorem Ipsum, et, plus récemment, par son inclusion dans des applications de mise en page de texte, comme Aldus PageMaker.

On sait depuis longtemps que travailler avec du texte lisible et contenant du sens est source de distractions, et empêche de se concentrer sur la mise en page elle-même. L'avantage du Lorem Ipsum sur un texte générique comme 'Du texte. Du texte.' est qu'il possède une distribution de lettres plus ou moins normale, et en tout cas comparable avec celle du français standard. De nombreuses suites logicielles de mise en page ou éditeurs de sites Web ont fait du Lorem Ipsum leur faux texte par défaut, et une recherche pour 'Lorem Ipsum' vous conduira vers de nombreux sites qui n'en sont encore qu'à leur phase de construction. Plusieurs versions sont apparues avec le temps, parfois par accident, souvent intentionnellement (histoire d'y rajouter de petits clins d'œil, voire des phrases embarrassantes).

La Légende

Le texte ...

Pour observer votre page Html, vous pouvez cliquer sur Exécution > Lancer dans FireFox ou effectuer un clic droit sur votre fichier, ouvrir avec > firefox.

Nous apercevons l'utilisation de la police Cabin de Google

Autre exemple avec l'import d'une police distante (hébergée sur le web)

texte2.html

```
<!Doctype html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Police css3</title>
    <link rel="stylesheet" href="texte2.css">
    <link href="http://fonts.googleapis.com/css?family=Roboto:400,900italic,900,700italic,700,500italic,500,400italic,300italic,300,100italic,100" rel="stylesheet" type="text/css">
  </head>
  <body>
    <h1>Mon Titre</h1><!-- Titre principal -->
    <h2>Autre Titre</h2> <!-- SousTitre -->
    <p>Le Lorem Ipsum <strong>est simplement du faux texte</strong> employé dans la composition et la mise en page avant impression.<br>Le Lorem Ipsum est le faux texte standard de l'imprimerie depuis les années 1500, <cite>quand un peintre anonyme assembla ensemble des morceaux de texte</cite> pour réaliser un livre spécimen de polices de texte.<br>Il n'a pas fait que survivre cinq siècles, mais s'est aussi <strike>adapté à la bureautique</strike> informatique, sans que son contenu n'en soit modifié.<br>Il a été popularisé dans les années 1960 grâce à la vente de feuilles Letraset contenant des passages du Lorem Ipsum, et, plus récemment, par son inclusion dans des applications de mise en page de texte, comme Aldus PageMaker.<br><br>
```

On sait depuis longtemps que travailler avec du texte lisible et contenant du sens est source de distractions, et empêche de se concentrer sur la mise en page elle-même. L'avantage du Lorem Ipsum sur un texte générique comme 'Du texte. Du texte.' est qu'il possède une distribution de lettres plus ou moins normale, et en tout cas comparable avec celle du français standard. De nombreuses suites logicielles de mise en page ou éditeurs de sites Web ont fait du Lorem Ipsum leur faux texte par défaut, et une recherche pour 'Lorem Ipsum' vous conduira vers de nombreux sites qui n'en sont encore qu'à leur phase de construction. Plusieurs versions sont apparues avec le temps, parfois par accident, souvent intentionnellement (histoire d'y rajouter de petits clins d'œil, voire des phrases embarrassantes). </p>

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

Dans ce contexte, nous n'avons pas besoin d'utiliser la règle @font-face car elle est déjà présente dans le fichier CSS que nous chargeons.

texte2.css

```
html, body {
```

```
margin: 0;  
font-family: 'Roboto', sans-serif;  
font-size: 0.9em;  
}
```

Le Résultat

Mon Titre

Autre Titre

Le Lorem Ipsum est simplement du faux texte employé dans la composition et la mise en page avant impression. Le Lorem Ipsum est le faux texte standard de l'imprimerie depuis les années 1500, quand un peintre anonyme assembla ensemble des fragments de livres pour en faire un livre à destination d'un imprimeur. Il n'a pas fait que survivre cinq siècles, mais s'est aussi adapté à la bureautique informatique, sans que son contenu n'en soit modifié. Il a été popularisé dans les années 1960 grâce à la vente de feuilles Letraset contenant des passages du Lorem Ipsum, et, plus récemment, par les logiciels d'alignement et d'espacement des textes comme Aldus PageMaker.

On sait depuis longtemps que travailler avec du texte lisible et contenant du sens est source de distractions, et empêche de se concentrer sur les lettres plus ou moins normale, et en tout cas comparable avec celle du français standard. De nombreuses suites logicielles de mise

Nous apercevons l'utilisation de la police Roboto de Google

31

Importer une police d'écriture localement

Nous allons créer deux nouveaux fichiers :  `texte3.html` et  `texte3.css`

Ces deux nouveaux fichiers seront placés dans notre dossier  /HtmlCss/02-Structure/.

Ce qui donnera : /Html/Css/02-Structure/texte3.html et /Html/Css/02-Structure/texte3.css

Voici le contenu de texte3.html :

texte3.html

```
<!Doctype html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Police css3</title>
    <link rel="stylesheet" href="texte3.css">
  </head>
  <body>
    <h1>Mon Titre</h1><!-- Titre principal -->
    <h2>Autre Titre</h2> <!-- SousTitre -->
    <p>Le Lorem Ipsum <strong>est simplement du faux texte</strong> employé dans la composition et la mise en page avant impression.<br>Le Lorem Ipsum est le faux texte standard de l'imprimerie depuis les années 1500, <cite>quand un peintre anonyme assembla ensemble des morceaux de texte</cite> pour réaliser un livre spécimen de polices de texte.<br>Il n'a pas fait que survivre cinq siècles, mais s'est aussi <strike>adapté à la bureautique</strike> informatique, sans que son contenu n'en soit modifié.<br>Il a été popularisé dans les années 1960 grâce à la vente de feuilles Letraset contenant des passages du Lorem Ipsum, et, plus récemment, par son inclusion dans des applications de mise en page de texte, comme Aldus PageMaker.<br><br>
```

On sait depuis longtemps que travailler avec du texte lisible et contenant du sens est source de distractions, et empêche de se concentrer sur la mise en page elle-même. L'avantage du Lorem Ipsum sur un texte générique comme 'Du texte. Du texte. Du texte.' est qu'il possède une distribution de lettres plus ou moins normale, et en tout cas comparable avec celle du français standard. De nombreuses suites logicielles de mise en page ou éditeurs de sites Web ont fait du Lorem Ipsum leur faux texte par défaut, et une recherche pour 'Lorem Ipsum' vous conduira vers de nombreux sites qui n'en sont encore qu'à leur phase de construction. Plusieurs versions sont apparues avec le temps, parfois par accident, souvent intentionnellement (histoire d'y rajouter de petits clins d'œil, voire des phrases embarrassantes). </p>

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

Décryptons ensemble les données saisies :

- **h1** Titre Principal.
 - **h2** Sous Titre.
 - **p** la balise P permet de déclarer un paragraphe pour y introduire du texte.
 - **br** passage à la ligne

- **strong** permet de mettre en gras une partie d'un texte.
- **strike** permet de barrer une partie d'un texte (utile pour les soldes sur les sites ecommerce lorsque les prix sont en promotion).
- **cite** permet d'introduire une citation (italique).

Pour les titres, nous pouvons nous servir des balises suivantes : **h1, h2, h3, h4, h5, h6** (du plus grand au plus petit).

Pour observer votre page, vous pouvez cliquer sur le fichier Html puis vous rendre dans Exécution > Lancer dans FireFox ou effectuer un clic droit sur votre fichier, ouvrir avec > firefox.

Dans le navigateur, cela devrait vous donner le résultat suivant :

Le Résultat

Mon Titre

Autre Titre

Le Lorem Ipsum est **simplement du faux texte** employé dans la composition et la mise en page avant impression.

Le Lorem Ipsum est le faux texte standard de l'imprimerie depuis les années 1500, quand un peintre anonyme assembla ensemble des morceaux de texte pour réaliser un livre spécimen de polices de texte.

Il n'a pas fait que survivre cinq siècles, mais s'est aussi adapté à la bureautique informatique, sans que son contenu n'en soit modifié.

Il a été popularisé dans les années 1960 grâce à la vente de feuilles Letraset contenant des passages du Lorem Ipsum, et, plus récemment, par son inclusion dans des applications de mise en page de texte, comme Aldus PageMaker.

On sait depuis longtemps que travailler avec du texte lisible et contenant du sens est source de distractions, et empêche de se concentrer sur la mise en page elle-même. L'avantage du Lorem Ipsum sur un texte générique comme 'Du texte. Du texte. Du texte.' est qu'il possède une distribution de lettres plus ou moins normale, et en tout cas comparable avec celle du français standard. De nombreuses suites logicielles de mise en page ou éditeurs de sites Web ont fait du Lorem Ipsum leur faux texte par défaut, et une recherche pour 'Lorem Ipsum' vous conduira vers de nombreux sites qui n'en sont encore qu'à leur phase de construction. Plusieurs versions sont apparues avec le temps, parfois par accident, souvent intentionnellement (histoire d'y rajouter de petits clins d'œil, voire des phrases embarrassantes).

La Légende

Le texte ...

Pour le moment cette police d'écriture est classique, créons un nouveau fichier : **texte3.css**.

Ecrivons le fichier **texte3.css** ensemble :

```
texte3.css

@font-face {
    font-family: 'folksbold';
    src: url('police/Folks-Bold-webfont.eot');
    src: url('police/Folks-Bold-webfont.eot?#iefix') format('embedded-opentype'),
         url('police/Folks-Bold-webfont.woff') format('woff'),
         url('police/Folks-Bold-webfont.ttf') format('truetype'),
         url('police/Folks-Bold-webfont.svg#folksbold') format('svg');
    font-weight: normal;
    font-style: normal;
}
body{
    font-family: 'folksbold';
}
```

Déryptons ensemble le code css ci-dessus :

- **@font-face** est une règle utilisée dans la dernière version du langage CSS (la version 3), c'est la raison pour laquelle vous entendrez souvent parlé de police css3
- **font-family** est une propriété permettant de préciser la police d'écriture à utiliser.
- **text-align** est une propriété permettant d'aligner du texte.

- `color` est une propriété permettant de colorer du texte.
- `letter-spacing` est une propriété permettant d'espacer les lettres dans un mot ou une phrase.
- `font-size` est une propriété permettant de modifier la taille du texte.

La police `folksbold` fonctionne avec des fichiers locaux (dans votre ordinateur), vous pourrez les télécharger à l'adresse suivante : [Télécharger la police folds bold](#)



➤ Informations

Pour l'exemple, il est nécessaire de télécharger les polices et les placer dans un dossier nommé "police".

Avec le fichier CSS, voici le résultat dans le navigateur :

Le Résultat

Mon Titre

Autre Titre

Le Lorem Ipsum est simplement du faux texte employé dans la composition et la mise en page avant impression. Le Lorem Ipsum est le faux texte standard de l'imprimerie depuis les années 1500, quand un peintre anonyme assembla ensemble des morceaux de texte pour réaliser un livre spécimen de polices de texte. Il n'a pas fait que survivre cinq siècles, mais s'est aussi adapté à la bureautique informatique, sans que son contenu n'en soit modifié. Il a été popularisé dans les années 1960 grâce à la vente de feuilles Letraset contenant des passages du Lorem Ipsum, et, plus récemment, par son inclusion dans des applications de mise en page de texte, comme Aldus PageMaker.

On sait depuis longtemps que travailler avec du texte lisible et contenant du sens est source de distractions, et empêche de se concentrer sur la mise en page elle-même. L'avantage du Lorem Ipsum sur un texte générique comme 'Du texte. Du texte. Du texte.' est qu'il possède une distribution de lettres plus ou moins normale, et en tout cas comparable avec celle du français standard. De nombreuses suites logicielles de mise en page ou éditeurs de sites Web ont fait du Lorem Ipsum leur faux texte par défaut, et une recherche pour 'Lorem Ipsum' vous conduira vers de nombreux sites qui n'en sont encore qu'à leur phase de construction. Plusieurs versions sont apparues avec le temps, parfois par accident, souvent intentionnellement (histoire d'y rajouter de petits clins d'œil, voire des phrases embarrassantes).

Pour observer votre page Html, vous pouvez cliquer sur Exécution > Lancer dans FireFox ou effectuer un clic droit sur votre fichier, ouvrir avec > firefox.

Nous apercevons l'utilisation de la police Folks Bold de Google

Les sélecteurs

32

Les différents selecteurs

Les selecteurs en CSS permettent « d'attraper » des éléments déclarés dans le fichier Html.

Nous pouvons en lister 3 principaux :

1. ➤ Les selecteurs de balises
2. ➤ Les selecteurs de classe
3. ➤ Les selecteurs d'id

Cela permet de mettre en forme une partie de votre page web en appelant directement la balise concernée.

Exemple Html :

```
<div>
    Div permet de créer un emplacement dans un site web.
</div>
```

Div permet de créer un emplacement dans un site web.

On utilise le mot div pour "division". Un site n'est qu'une série de zones divisées.

Exemple Css (avec selecteur de balise) :

```
div{
    color: green;
}
```

Le Résultat

Div permet de créer un emplacement dans un site web.

En sélectionnant la balise div (et en appliquant la propriété color), le texte à l'intérieur de la div (zone) sera écrit en couleur verte.

Ce code CSS permet de dire « va me chercher la ou les div dans le fichier html et écrit le texte à l'intérieur en couleur verte ».

Pour cela il y a 2 conditions :

- il faut naturellement que la balise div existe dans le fichier html
- il faut impérativement que les deux fichiers (html et css) soit reliés entre eux, grâce à la balise link.

Autre exemple, imaginons que nous ayons plusieurs zones (ce qui est le cas dans tous les sites web).

Observons le fichier Html :

```
<div>
    Zone n°1
</div>
<div>
    Zone n°2
</div>
```

Et le fichier Css :

```
div{
    color: green;
}
```

Le Résultat

Zone n°1

Zone n°2

Voyez-vous la problématique ? Lorsque nous sélectionnons "div{}" dans le fichier CSS, les deux div sont attrapées et sont mises en forme (couleur verte).

Cependant il arrive régulièrement que nous souhaitons appliquer du style à une zone en particulier de manière localisée.

Pour plus de personnalisation, nous allons donc nous servir des sélecteurs de classes.

Exemple avec le fichier Html :

```
<div class="mazoneperso">
    Zone n°1
</div>
<div>
    Zone n°2
</div>
```

Dans ce fichier Html, une des deux zones (div) possède un nom permettant de la représenter.

Et le fichier Css :

```
.mazoneperso{  
color: green;  
}
```

Dans ce fichier CSS, je ne selectionne pas toutes les zones (div) mais seulement celles qui possèdent le nom "mazoneperso".

Pour selectionner une classe, j'utilise un point ".mazoneperso{}".

Le Résultat

Zone n°1

Zone n°2

De cette manière, je suis certain de cibler la bonne zone pour appliquer la mise en forme.

35

Selecteur d'id

Dans le principe, un id est très similaire à une classe puisque l'objectif est de donner un nom à une zone pour mieux l'identifier.

Exemple avec un fichier Html :

```
<div id="zoneprincipale">  
Zone n°1  
</div>  
<div>  
Zone n°2  
</div>
```

Dans ce fichier Html, la première des deux zones (div) possède un nom permettant de l'identifier.

Et voici le fichier Css qui l'accompagne :

```
#zoneprincipale{  
color: green;  
}
```

Le Résultat

Zone n°1

Zone n°2

Avouez que ce n'est pas compliqué, l'objectif d'un ID est donc le même que celui d'une CLASS.

Techniquement, je selectionnerais une CLASS avec un point . et un ID avec le symbole dieze #

Quelle est la différence entre une CLASS et un ID ?

Le W3C dit que nous pouvons utiliser une même classe plusieurs fois et que nous devons utiliser qu'une seule fois le même id dans un fichier Html.

Dans les faits, nous pouvons utiliser plusieurs fois le même id et plusieurs fois la même classe, cela fonctionne très bien.

Pour ne pas froisser le W3C :D et avoir une chance de valider notre code, nous suivrons leur recommandation (un id sera utilisé qu'une seule fois).

Voici un code d'exemple valide W3C :

```
<!Doctype html>  
<html>  
<head>  
 <meta charset="utf-8">  
 <title>Mon Titre</title>  
</head>  
<body>  
 <div class="test"></div>  
 <div class="test"></div> <!-- je peux ré-utiliser la classe test une deuxième fois -->  
 <div id="test2"></div>  
</body>  
</html>
```

Par conséquent, Les Intégrateurs / Développeur Front de métier utilise les ID pour identifier les zones principales d'un site web et les classes pour tout le reste.

Nous utilisons plus souvent des classes dans un code-source.

36

Selecteur d'éléments

Exemple avec un fichier Html :

```
<p> Texte n°1 </p>
<p> Texte n°2 </p>
<p> Texte n°3 </p>
```

Et, le fichier Css :

```
p{ color: red; }
p:first-child { color: green; }
p:last-child { color: blue; }
```

Le Résultat

Texte n°1

Texte n°2

Texte n°3

- p permet de cibler tous les éléments p
- first-child permet de cibler le premier élément p
- last-child permet de cibler le dernier élément p

37

Les règles d'écriture

Il arrive parfois qu'une même classe CSS puisse être utilisée et appelée à plusieurs endroits de votre fichier Html, exemple :

```
<div class="mazoneperso">
  Zone n°1
</div>
<div>
  Zone n°2
</div>
<span class="mazoneperso">
  Zone n°3
</span>
```

Dans ce cas, nous adapterons le fichier CSS, exemple :

```
div.mazoneperso{
  color: green;
}
```

Cela veut dire « va me chercher la div qui possède la classe mazoneperso », ce qui éliminera le span (puisque ce n'est pas une div !).

Le Résultat

Zone n°1

Zone n°2

Zone n°3

Il est également possible de préciser la hiérarchie qui vous emmène jusqu'à un élément en particulier.

Exemple Html :

```
<section>
  <article>Article 1</article>
  <p>Texte 1</p>
  <div>Zone 1</div>
</section>
<div>
  Zone n°2
</div>
<div>
  Zone n°3
</div>
```

Exemple Css :

```
section div{
  color: red;
}
```

Le Résultat

Article 1

Texte 1

Zone 1

Zone n°2

Zone n°3

Dans le fichier CSS, l'espace entre les deux balises permet de dire « va me chercher la div qui se trouve à l'intérieur de la section », ce qui touchera la zone 1 et éliminera les zones 2 et 3 qui sont en dehors de la section.

Il est également possible de regrouper des styles de mises en forme, exemple Html :

```
<header>
  Haut de site
</header>
<section>
  Milieu de site
</section>
<footer>
  Bas de site
</footer>
```

Exemple Css :

```
header, footer{
  color : purple;
}
```

Le Résultat

Haut de site

Milieu de site

Bas de site

Dans le fichier CSS, la virgule entre les deux balises permet de dire « applique ce style sur la balise header ET footer », ce qui touchera le contenu des deux zones (header et footer) et éliminera la zone section.

Il est tout à fait possible de donner un style global pour une zone et un style spécifique pour une partie de texte.

Exemple Html :

```
<div class="mazone">Voici le <span class="montexte">texte</span> de ma Zone</div>
```

Exemple Css :

```
.mazone{
    color : red;
}
.montexte{
    color : blue;
}
```

Le Résultat

Voici le texte de ma Zone



➤ Bon à savoir

Une classe CSS peut contenir un chiffre, par exemple : *maclasse2*

Une classe CSS ne peut pas commencer par un chiffre, exemple : *2maclasse*

Une classe CSS est sensible à la casse, ainsi ce code ne fonctionnera pas :

Exemple Html :

```
<div class="Maclasser">Voici une zone avec Maclasser</div>
```

Exemple Css :

```
.maclasse{
    color : red;
}
```

Le Résultat

Voici une zone avec Maclasser

En effet, la classe a été nommée avec un M majuscule au niveau du code Html et en minuscule du côté CSS.

Dans le nom d'une classe CSS, nous pouvons mettre des lettres minuscules ou majuscules de l'alphabet sans accent.

Une classe peut aussi contenir tous les chiffres de 0 à 9 et les caractères (underscore) "_" et (tiret) "-".

Dans tous les cas, il faut rester cohérent entre le nom que vous donnerez côté Html et celui que vous utiliserez côté CSS.

Liste des selecteurs CSS :

Selecteurs Généraux			
Selecteur	Exemple CSS	Exemple HTML	Description
balise	div{}	<div> mazone1 </div> <div> mazone2 </div>	Selectionne toutes les balises <div>
.class	.mazone{}	<div class="mazone"> mazone1 </div> <div> mazone2 </div>	Selectionne tous les éléments possédant la class="mazone".
#id	#zoneprincipale{}	<div id="zoneprincipale"> mazoneprincipale1 </div>	Selectionne l'élément possédant l'id="zoneprincipale"

		<div> mazone2 </div>	
*	*{}	<header> haut de site </header> <div> zone </div>	Selectionne tous les éléments de la page
balise balise balise > balise	div p{} div > p{}	<div> <p> mon texte </p> </div>	L'espace permet de selectionner toutes les balises <p> à l'intérieur des balises <div> Ce chevron permet de selectionner toutes les balises p ayant comme parent la balise div.
balise, balise	div, p{}	<div> mazone </div> <p> mon texte </p>	La virgule permet de selectionner toutes les balises <div> et toutes les balises <p>
balise + balise	div + p{}	<div> mazone </div> <p> mon texte </p>	Le + permet de selectionner toutes les balises <p> placées immédiatement après une balise <div> (sans autres éléments entre les deux)
balise ~ balise	div ~ p{}	<div> mazone </div> <p> mon texte </p>	Selectionne les balises <p> précédées par une balise <div>
balise.class	span.mazone{}	 mazone1 	Selectionne tous les éléments span possédant la class="mazone".

Selecteurs d'attributs			
Selecteur	Exemple CSS	Exemple HTML	Description
[attribut]	[title]{}	 Le lien 	Selectionne tous les éléments ayant l'attribut title
[attribut=valeur]	[type="text"]{}	<input type="text" name="monchamp">	Selectionne tous les éléments ayant l'attribut type="text"
[attribut\$=valeur]	img[src\$=".png"]{}		Selectionne toutes les balises <a> ayant la terminaison ".png"
[attribut^=valeur]	a[href^="http"]{}	google	Selectionne toutes les balises <a> commençant par "http"
[attribut~=valeur]	[title~=Paris]{}		Selectionne tous les éléments ayant dans leur titre "Paris". (Attention, à vérifier).

Selecteurs Complémentaires - pseudo éléments			
Selecteur	Exemple CSS	Exemple HTML	Description
:hover	a:hover{}	Mon Lien	Selectionne les liens lors du survol pour leur appliquer un style spécifique
:active	a:active{}	Mon Lien	Selectionne les liens actifs
:link	a:link{}	Mon Lien	Selectionne tous les liens qui n'ont pas été visités
:visited	a:visited{}	Mon Lien	Permet d'appliquer une mise en forme sur les liens ayant déjà été visités.
:not(selecteur)	div:not(.mazone2){}	<div class="mazone1">Ma zone perso 1</div> <div class="mazone2">Ma zone perso 2 </div> <div class="mazone3">Ma zone perso 3 </div>	Selectionne toutes les zones div autres que la zone div.mazone2{}
:first-child	p:first-child{}	<p>Mon premier paragraphe</p> <p>Mon second paragraphe</p>	Selectionne la première balise <p> du corps de la page (body) ou les premières balises p à l'intérieur d'une autre balise parent
:last-child	p:last-child{}	<p>Mon paragraphe</p> <p>Mon dernier paragraphe</p>	Selectionne la dernière balise <p> du corps de la page (body) ou les dernières balises p à l'intérieur d'une autre balise parent
:nth-child(n)	p:nth-child(2){}	<p>Texte n°1</p> <p>Texte n°2</p>	Selectionne toutes les balises <p> étant positionnées en 2e élément dans son parent ou dans le corps de page général (body)
:first-letter	p:first-letter{}	<p>Ce texte comporte une première lettre</p>	Selectionne la 1ère lettre du texte contenue dans tous les éléments <p>
:first-line	p:first-line{}	<p>Ce texte comporte plusieurs	Selectionne chaque 1ère ligne de chaque balise <p>

nom de la classe	paramètres	lignes</p>	Selectionne chaque ligne de chaque balise spec
:after	h1:after{}	<h1>Mon Titre</h1>	Permet d'insérer du contenu après un élément <p>
:before	h1:before{}	<h1>Mon Titre</h1>	Permet d'insérer du contenu avant un élément <p>
::selection	::selection{}	-	permet de mettre en forme la partie du site sélectionné et mis en surbrillance par l'internaute (comme lors d'un copier/coller)
:target	#zone1 :target{}	Mon Lien	Permet d'appliquer un style sur l'élément actif présent dans l'url (#zone1) suite à un clic de l'internaute
:required	input:required{}	<input type="text" required="required">	Permet de mettre en forme sur les balises input ayant l'attribut "required".

Firebug

42

Comment télécharger FireBug ?

FireBug est indispensable pour tout Intégrateur / Développeur Front !

Pour télécharger FireBug, vous devez vous munir du navigateur firefox et [télécharger l'extension firebug](#) en cliquant sur le bouton vert "Ajouter à Firefox".

Firebug est à l'origine une extension pour Firefox afin d'aider à la création de page web.

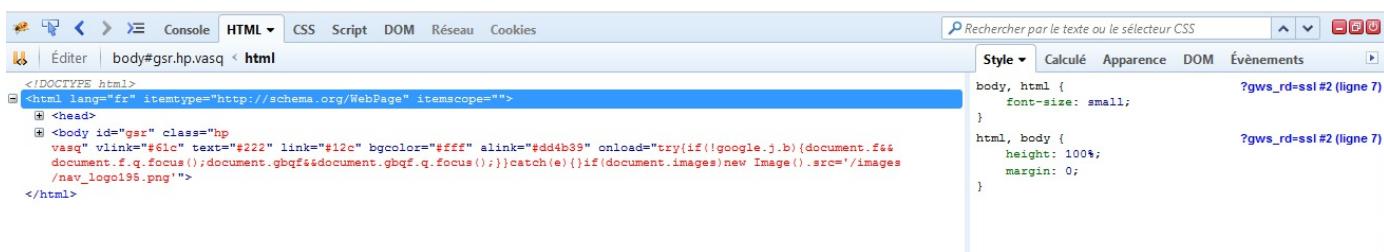
Vous pourrez trouver son équivalence sur Google Chrome.

Les autres navigateurs possèdent généralement leur propre extension.

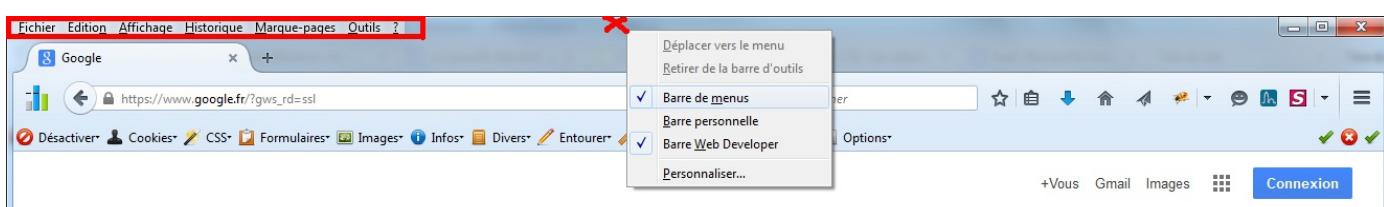
43

Comment utiliser FireBug ?

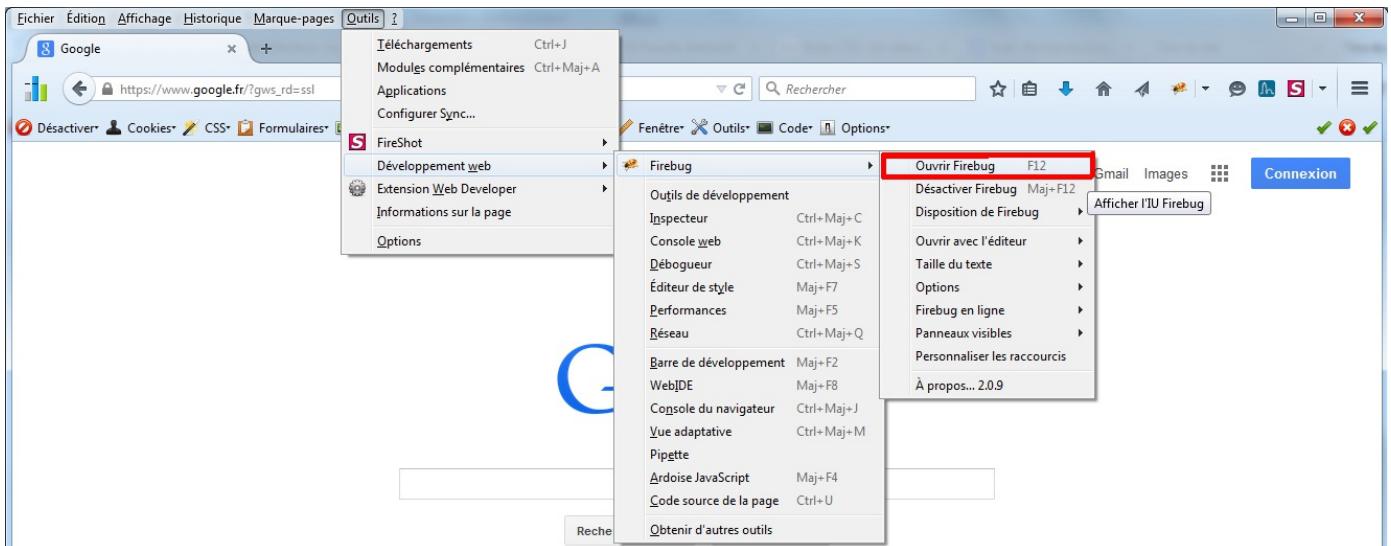
Pour ouvrir FireBug, il faut d'abord vous rendre dans votre navigateur FireFox et appuyer sur la touche **F12** de votre clavier. Cela devrait ouvrir un panneau (en bas de page) comme ceci :



Si ce n'est pas le cas, vous pouvez effectuer un clic droit dans la zone du haut de firefox pour faire apparaître la barre de menu, comme ceci :



Et ensuite vous rendre dans la partie Outils > Développement Web > Firebug > Ouvrir Firebug :



Exemple avec le logo de Google

Lorsque votre panneau apparaît, nous allons cliquer sur le sélecteur en bas à gauche :

The screenshot shows the Google homepage with the classic blue, red, and green logo. A large red arrow points from the left side of the screen towards the logo. At the bottom of the browser window, the Firebug toolbar is open, showing various tabs like 'Console', 'Script', 'DOM', 'Réseau', and 'Cookies'. The 'HTML' tab is currently active. A red box highlights the selector icon (a small square with a diagonal line) located on the far left of the toolbar.

Et nous allons cliquer une fois sur le logo Google pour se positionner dessus (sans recharger la page).

Ensuite, nous allons effectuer un clic droit dans la partie CSS :

Le code CSS de la page

Le code HTML de la page

The screenshot shows the Firebug interface over a Google search results page. The CSS panel on the right contains the following code:

```

element.style {
    background: url("/images/nav/nav_bg_w.png") no-repeat scroll 0 0 / 0;
    height: 95px;
    width: 269px;
}

```

The HTML panel on the left shows the DOM structure of the Google page.

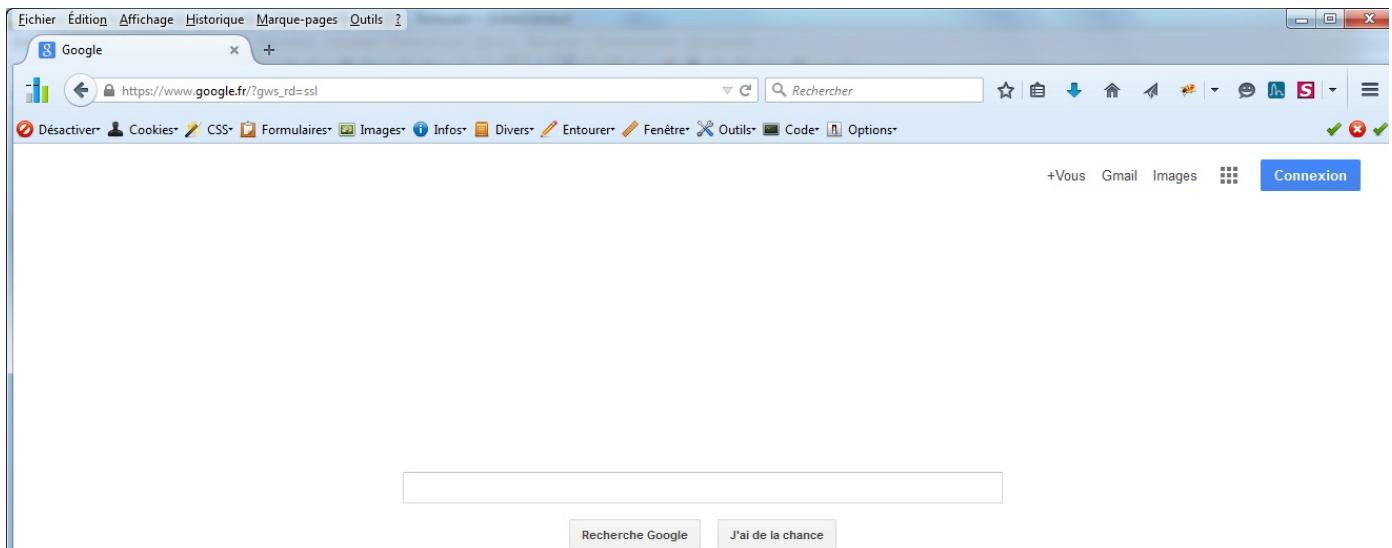
Nous allons choisir "Nouvelle Propriété" et écrire :

```
display: none;
```

Et voilà le travail, avec la propriété `display: none;` qui permet de masquer une partie de l'affichage, nous avons réussi à retirer le logo de Google !

Bien entendu, cela sera rétabli dès que l'on rechargea la page ! Les modifications ne sont pas enregistrées par FireBug (cela permet seulement d'avoir un aperçu des modifications que nous pouvons apporter).

De plus, ce n'est pas notre site web et Google ne serait pas très content si la modification était réellement effective :p !



Le code HTML de la page

The screenshot shows the Firebug developer toolbar open in a browser window. The left pane displays the HTML structure of the Google homepage, with a red arrow pointing to it from the text 'Le code HTML de la page'. The right pane shows the CSS styles applied to the page, with a red arrow pointing to it from the text 'Le code CSS de la page'. The CSS pane includes a search bar at the top and tabs for Style, Calculé, Apparence, DOM, and Événements.

Exemple avec la couleur des liens de Google

Nous allons effectuer une recherche sur Google et sélectionner en cliquant l'un des liens qui apparaît en bleu.

Google actualités

Web Actualités Shopping Vidéos Maps Plus Outils de recherche Connexion

Environ 218 000 000 résultats (0,19 secondes)

Google Actualités
news.google.fr/ ▾
Offre une compilation d'articles publiés par un grand nombre de sources d'actualités francophones. Il est possible d'accéder aux informations des régions francophones dans le ...

Yahoo Actualités - Toute l'actualité en France et dans le ...
https://fr.news.yahoo.com/ ▾
Suivez l'actualité en France et dans le monde sur Yahoo Actualités. Consultez et recherchez dans la source d'information la plus complète du web.

Le Parisien: Actualités en direct et info en continu
www.leparisien.fr/ ▾
L'actualité du quotidien de l'Ile-de-France triée par département et par rubrique. Service "Vivre ma Ville" pour un accès aux infos locales, aux sorties et aux sports.

Actualités en direct, info en continu - 20minutes.fr
www.20minutes.fr/actus ▾
COREE DU SUD. VIDEO. Corée du Sud: Tristesse et colère toujours vives, un an après le naufrage du ferry... 09:08. 0; 3; 4; 0; 0; 7. FOOTBALL. PSG-Barça ...

Le Point – Actualité Politique, Monde, France, Économie ...
www.lepoint.fr/ ▾

Le code CSS de la page

Le code HTML de la page

The screenshot shows a browser window with a search results page from Google. Below the main content, the developer tools are open, specifically the 'HTML' tab. A red arrow points from the text 'Le code HTML de la page' to the left side of the developer tools, where the HTML tree is displayed. Another red arrow points from the text 'Le code CSS de la page' to the right side, where the CSS styles are listed. The HTML tree highlights a specific link to 'Google Actualités'. The CSS panel lists several rules, with annotations like '?gws_rd=...ualités #2 (ligne 7)' next to some of them.

```

#res h3.r {
    display: block;
    overflow: hidden;
    text-overflow: ellipsis;
    white-space: nowrap;
}
#res h3 {
    font-size: 18px;
}
.r {
    font-size: medium;
}

```

Ensuite, nous allons ajouter une nouvelle propriété :

actualités - Recherche Goo... +

https://www.google.fr/?gws_rd=ssl#q=actualités

Rechercher

Désactiver Cookies CSS Formulaires Images Infos Divers Entourer Fenêtre Outils Code Options

Google actualités

Web Actualités Shopping Vidéos Maps Plus Outils de recherche

Environ 218 000 000 résultats (0,19 secondes)

Google Actualités
news.google.fr/

Offre une compilation d'articles publiés par un grand nombre de sources d'actualités francophones. Il est possible d'accéder aux informations des régions francophones dans le ...

Yahoo Actualités - Toute l'actualité en France et dans le ...
https://fr.news.yahoo.com/

Suivez l'actualité en France et dans le monde sur Yahoo Actualités. Consultez et recherchez dans la source d'information la plus complète du web.

Le Parisien: Actualités en direct et info en continu
www.leparisien.fr/

L'actualité du quotidien de l'Ile-de-France triée par département et par rubrique. Service "Vivre ma Ville" pour un accès aux infos locales, aux sorties et aux sports.

Actualités en direct, info en continu - 20minutes.fr
www.20minutes.fr/actus

COREE DU SUD. VIDEO. Corée du Sud: Tristesse et colère toujours vives, un an après le naufrage du ferry... 09:08. 0; 3; 4; 0; 0; 7. FOOTBALL. PSG-Barça ...

Le Point – Actualité Politique, Monde, France, Économie ...
www.lepoint.fr/

Le code CSS de la page

Le code HTML de la page

Style Calculé Apparence

Copier la déclaration de propriété Copier le nom de propriété Copier la valeur de propriété Éditer le style de l'élément... Ajouter une règle... Nouvelle propriété... Éditer "text-overflow"... Supprimer "text-overflow" Ajout Désactiver "text-overflow" Actualiser Utiliser en ligne de commande Inspecter dans le panneau DC Inspecter dans le panneau CS

```
#res h3.r {
    display: block;
    overflow: hidden;
    text-overflow: ellipsis;
    white-space: nowrap;
}
#res h3 {
    font-size: 18px;
}
.r {
    font-size: medium;
}
```

Selectionner la partie <a> dans la partie Html et saisir dans la partie CSS color : red ;

actualités - Recherche Goo... +

https://www.google.fr/?gws_rd=ssl#q=actualités

Rechercher

Désactiver Cookies CSS Formulaires Images Infos Divers Entourer Fenêtre Outils Code Options

Google actualités

Web Actualités Shopping Vidéos Maps Plus Outils de recherche

Environ 218 000 000 résultats (0,19 secondes)

Google Actualités
news.google.fr/ ▾
Offre une compilation d'articles publiés par un grand nombre de sources d'actualités francophones. Il est possible d'accéder aux informations des régions francophones dans le ...

Yahoo Actualités - Toute l'actualité en France et dans le ...
https://fr.news.yahoo.com/ ▾
Suivez l'actualité en France et dans le monde sur Yahoo Actualités. Consultez et recherchez dans la source d'information la plus complète du web.

Le Parisien: Actualités en direct et info en continu
www.leparisien.fr/ ▾
L'actualité du quotidien de l'Ile-de-France triée par département et par rubrique. Service "Vivre ma Ville" pour un accès aux infos locales, aux sorties et aux sports.

Actualités en direct, info en continu - 20minutes.fr
www.20minutes.fr/actus ▾
COREE DU SUD. VIDEO. Corée du Sud: Tristesse et colère toujours vives, un an après le naufrage du ferry... 09:08. 0; 3; 4; 0; 7. FOOTBALL. PSG-Barça ...

Le Point – Actualité Politique, Monde, France, Économie ...
www.lepoint.fr/ ▾

Le code CSS de la page

Le code HTML de la page

Les liens apparaissent en rouge !

Exemple avec la taille d'une zone

Ecrivons ce code dans un fichier HTML nommé firebug.html :

```
firebug.html
```

```
<!Doctype html>
<html>
<head>
<meta charset="utf-8">
<title>Entrainement sur l'utilisation de Firebug</title>
<link rel="stylesheet" href="firebug.css">
</head>
<body>
<div>Ma zone</div>
</body>
</html>
```

Ecrivons ce code dans un fichier CSS nommé firebug.css :

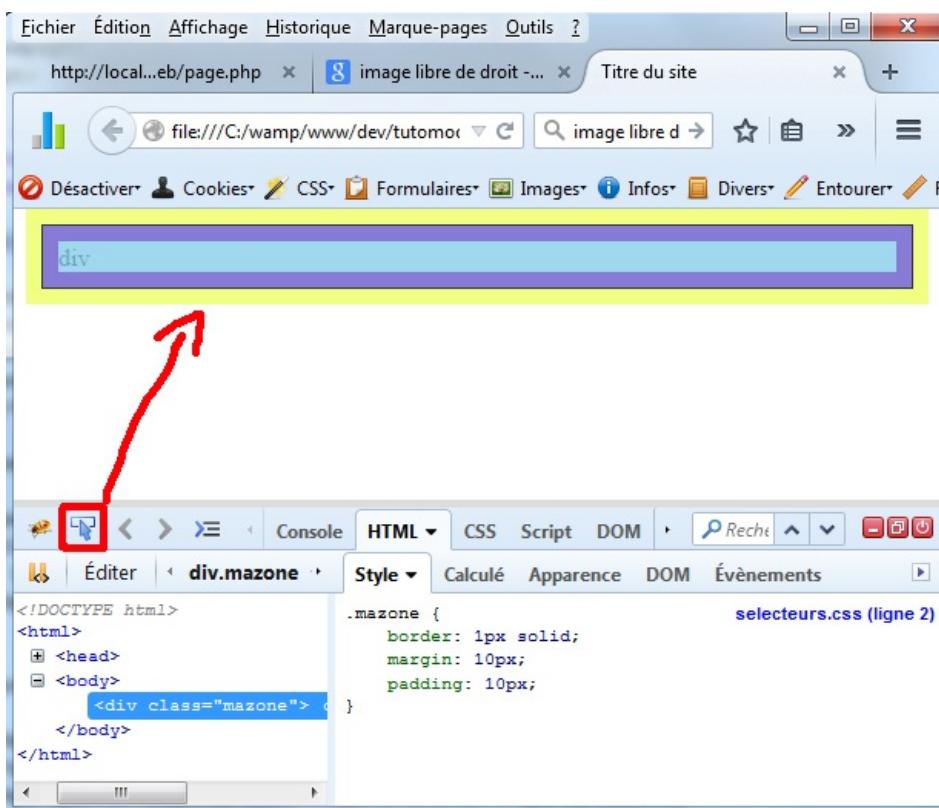
```
firebug.css
```

```
div{
background: orange;
width: 500px;
height: 500px;
}
```

Suivez ces étapes :

1. Afficher votre page web dans le navigateur
2. Attraper votre zone orange à l'aide de firebug

3. Dans la partie de droite cliquer sur "500px"
4. Augmenter la taille ou réduire la taille en pixels et voir le changement en direct à l'écran



Avec ces différents exemples nous savons que Firebug est un outil indispensable pouvant nous aider dans la création d'un site web notamment pour ajuster les proportions, positionner et tailler les différentes zones.

Les modifications apportées par firebug ne sont pas enregistrées. Cela nous permet d'avoir un aperçu précis en direct.

Si l'aperçu vous convient, vous devrez copier/coller les lignes de code de firebug dans la feuille de style css.

Styles Css

44

Styles CSS

Il existe plusieurs moyens d'écrire du code CSS :

1. ➤ Dans un fichier CSS
2. ➤ Dans un fichier HTML
3. ➤ Dans l'attribut d'un fichier HTML

Style css dans un attribut Html

Exemple Html :

```
<!Doctype html>
<html>
<head>
<meta charset="utf-8">
<title>Ma Page</title>
</head>
<body>
<p style="color: green; text-decoration: underline; font-weight: bold;">Mon premier paragraphe de texte !</p>
</body>
</html>
```

Style css directement dans un fichier Html

Exemple Html :

```
<!Doctype html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Ma Page</title>
    <style>
      .montexte { color: green; text-decoration: underline; font-weight: bold; }
    </style>
  </head>
  <body>

    <p class="montexte">Mon premier paragraphe de texte !</p>

  </body>
</html>
```

Le seul avantage de ces deux techniques réside dans le fait qu'il est possible de tout écrire au sein du même fichier.

L'inconvénient c'est que ce style ne pourra pas être utilisé ailleurs, à moins de le réécrire à nouveau, ce qui peut être long et imprécis.

Nous privilégions donc une feuille de style CSS à part entière en plus de notre fichier HTML.

Style dans un fichier CSS externalisé

Exemple Html :

```
<!Doctype html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Ma Page</title>
    <link rel="stylesheet" href="style.css">
  </head>
  <body>

    <p class="montexte">Mon premier paragraphe de texte !</p>

  </body>
</html>
```

Autre Exemple Html

```
<!Doctype html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Ma Page</title>
    <style type="text/css">@import url('style.css');</style>
  </head>
  <body>

    <p class="montexte">Mon premier paragraphe de texte !</p>

  </body>
</html>
```

Et le fichier CSS :

```
.montexte{ color: green; text-decoration: underline; font-weight: bold; }
```

45

Gérer les affichages : Display Inline & Block

Nous allons prendre un cas d'étude permettant de centrer du texte sur la page.

Comprendre la différence entre la propriété display inline et display block

Exemple Html :

display.html

```
<!Doctype html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Ma Page</title>
    <link rel="stylesheet" href="displaycss">
  </head>
```

```
<body>
  <h1> h1 </h1>
  <p> p </p>
  <span> span </span>
  <div> div </div>
  <a href=""> a </a>
</body>
</html>
```

Exemple Css :

display.css

```
h1{ text-align: center; }
p{ text-align: center; }
span{ text-align: center; }
div{ text-align: center; }
a{ text-align: center; }
```

La propriété `text-align` permet de gérer l'alignement d'un texte

Le Résultat

h1

p

span

div

a

Sur ce résultat, nous pouvons constater que chaque contenu est bien centré, sauf le span et le lien a !

Pourquoi est-ce que le contenu de mon span et mon lien (a) ne se centrent pas ?

Pour le comprendre il va falloir ajouter des bordures :

Exemple Css :

display.css

```
h1{ text-align: center; border: 1px solid; }
p{ text-align: center; border: 1px solid; }
span{ text-align: center; border: 1px solid; }
div{ text-align: center; border: 1px solid; }
a{ text-align: center; border: 1px solid; }
```

La propriété `border` permet d'ajouter une bordure - 1px correspond à l'épaisseur du trait - solid correspond à un trait plein.

Le Résultat

h1

p

span

div

a

Nous pouvons remarquer que le span et le a ne font pas toute la largeur de la page !

En effet, ils sont prévus pour occuper uniquement la place dont il ont besoin, on dit qu'il s'agit d'éléments de type inline !

C'est la raison pour laquelle nous ne pouvons pas centrer le texte, puisqu'il est déjà compressé à l'intérieur du span (et du lien a).

Si nous souhaitons les centrer, nous pouvons les passer en type block :

display.css

```
h1{ text-align: center; border: 1px solid; }
p{ text-align: center; border: 1px solid; }
span{ text-align: center; border: 1px solid; display: block; }
div{ text-align: center; border: 1px solid; }
a{ text-align: center; border: 1px solid; display: block; }
```

Le Résultat

h1
p
span
div
a

Nos textes à l'intérieur du span et de notre lien (a) sont bien centrés !

Pour les autres éléments : <h1>, <p>, <div>, il n'est pas nécessaire d'appliquer la propriété *display: block;* car il sont déjà en type block !



➤ Résumé

certains éléments sont de type inline et d'autres de type block.

Dans notre cas : h1, p et div sont de type block par défaut. span et a sont de type inline par défaut.

Nous pouvons changer la propriété display d'un élément via le code CSS :

```
display: block;  
display: inline;  
display: inline-block;
```

46

Margin et Padding

Dans un site web, il arrive fréquemment que l'on veuille espacer des éléments entre eux. Pour cela, il est courant d'utiliser les propriétés margin et padding.

Différence entre MARGIN et PADDING

Exemple Html

marginpadding.html
<!Doctype html> <html> <head> <meta charset="utf-8"> <title>Ma Page</title> <link rel="stylesheet" href="marginpadding.css"> </head> <body> <div class="maZoneA"> div </div> <div class="maZoneB"> div </div> </body> </html>

Exemple Css

marginpadding.css
.maZoneA{ border: 1px solid; background: blue; /* MARGIN : marges extérieures */ margin: 10px 9px 8px 7px; /* 10px en haut - 9px à droite - 8px en bas - 7px à gauche */ margin: 10px 7px; /* 10px en haut et en bas - 7px à gauche et à droite */ margin: 10px 9px 8px; /* 10px en haut - 9px à droite - 8px en bas. */ margin: 10px; /* 10px tout autour (haut, droite, bas, gauche). */ /* Cela peut aussi s'écrire de cette manière : */ margin-top: 10px; /* top en haut */

```

margin-right: 10px; /* right à droite */
margin-bottom: 10px; /* bottom en bas */
margin-left: 10px; /* left à gauche */

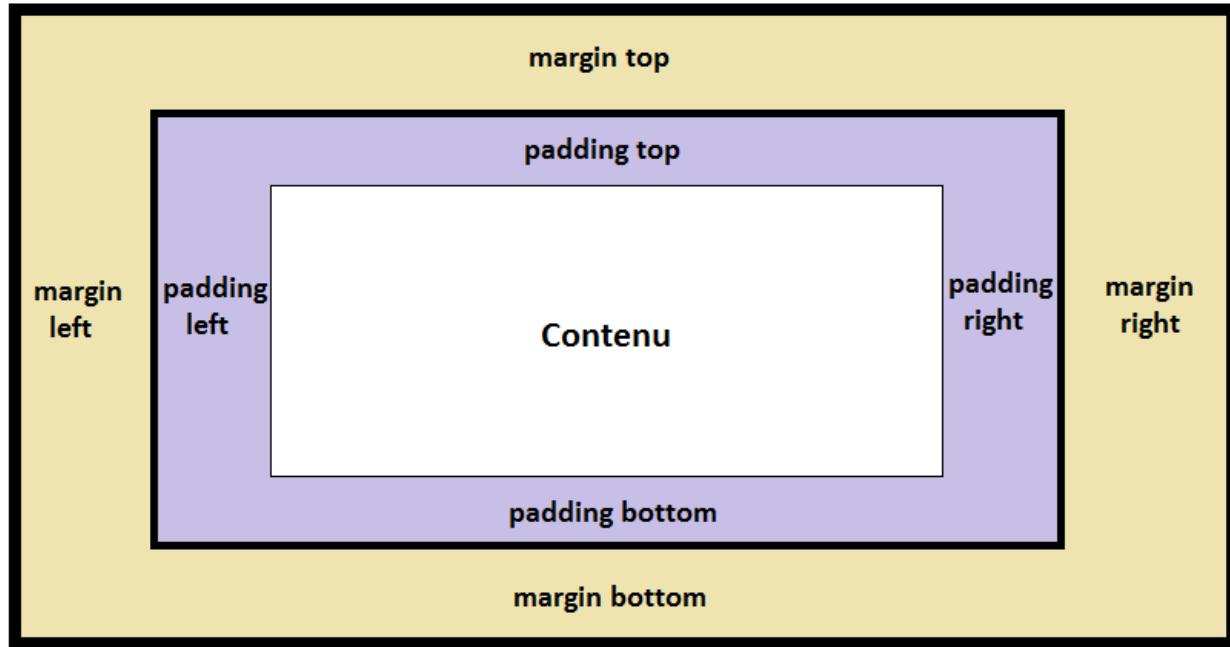
/* Même chose pour PADDING : marges intérieures */

padding: 10px 9px 8px 7px; /* 10px en haut - 9px à droite - 8px en bas - 7px à gauche */
padding: 10px 7px; /* 10px en haut et en bas - 7px à gauche et à droite */
padding: 10px 9px 8px; /* 10px en haut - 9px à droite - 8px en bas. */
padding: 10px; /* 10px à l'intérieur (haut, droite, bas, gauche). */
padding-top: 10px; /* top en haut */
padding-right: 10px; /* right à droite */
padding-bottom: 10px; /* bottom en bas */
padding-left: 10px; /* left à gauche */
}

.maZoneB{
border: 1px solid;
background: red;
}

```

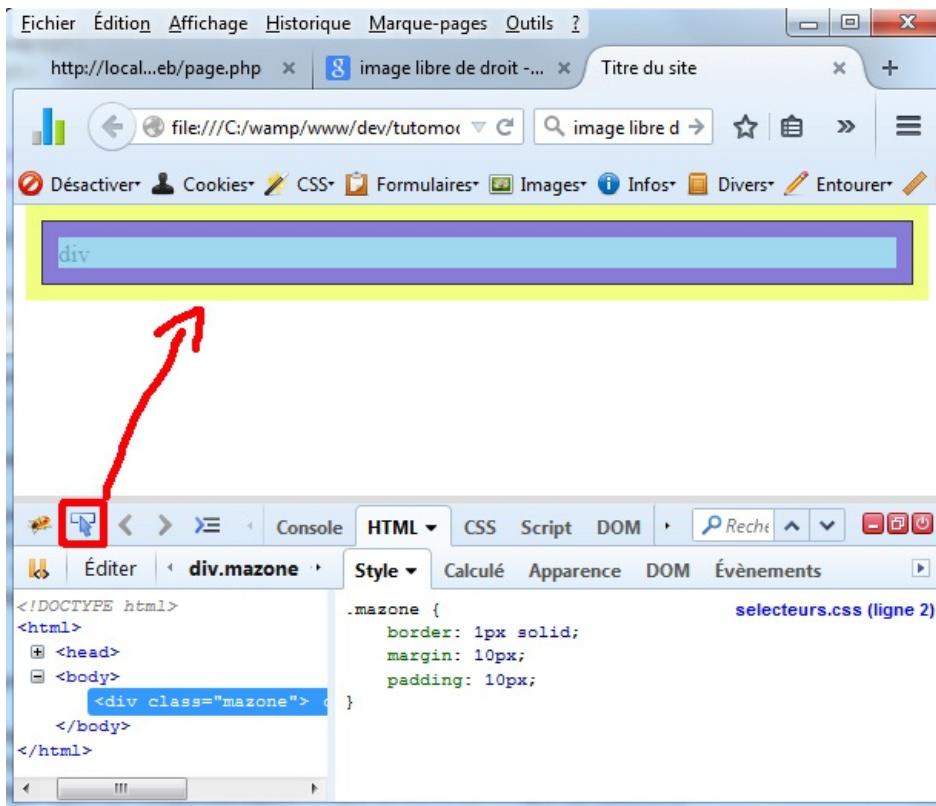
Toutes les explications se trouvent en commentaire directement dans le code, mais pour bien comprendre, rien ne vaut un bon schéma :



Le padding (espace intérieur d'une zone) apparaît en violet.

Le margin (espace extérieur d'une zone) apparaît en jaune.

Nous pouvons constater cela avec l'aide de Firebug et modifier directement les valeurs avec l'aperçu en direct :



Faites le test !

L'héritage

47

Qu'est-ce que l'héritage en CSS ?

Lorsque vous déclarez certaines balises, il arrive que ces éléments héritent d'un style prédéfini particulier.

Exemple Html :

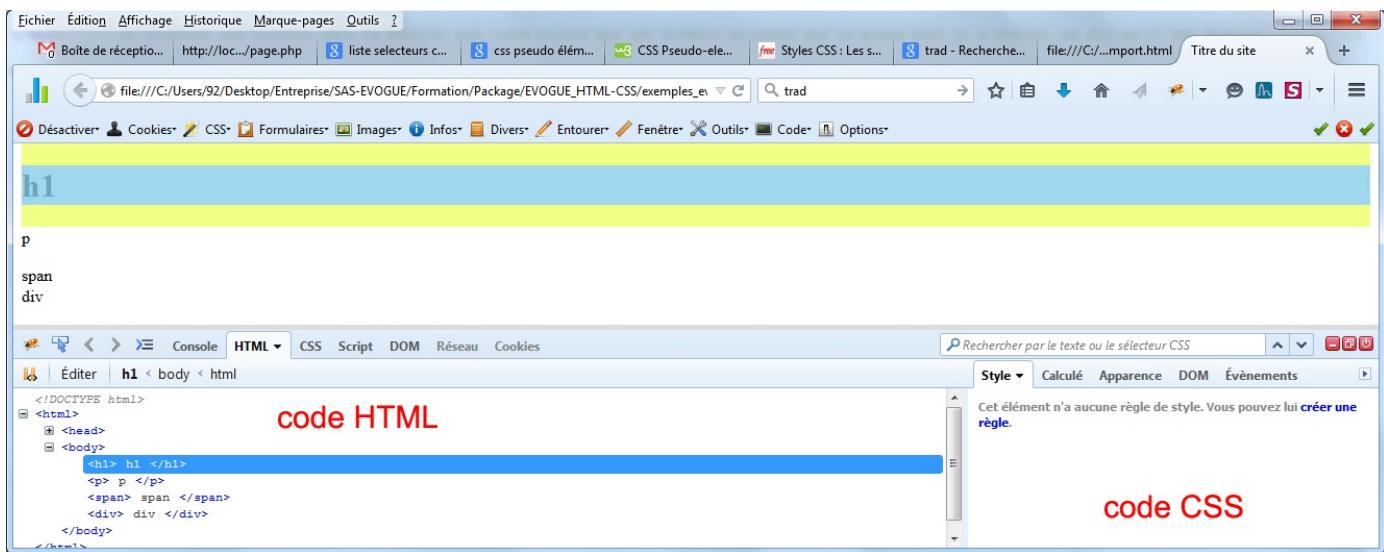
```
heritage.html

<!Doctype html>
<html>
<head>
<meta charset="utf-8">
<title>Ma Page</title>
<link rel="stylesheet" href="heritage.css">
</head>
<body>

<h1> h1 </h1>
<p> p </p>
<span> span </span>
<div> div </div>

</body>
</html>
```

Voici le résultat dans le navigateur :



Vous ne remarquez rien ?

Par exemple, en sélectionnant le H1 avec l'aide de FireBug, on se rend compte que celui-ci possède et hérite d'une marge et d'une certaine taille de caractères alors que nous n'avons rien demandé de particulier en code CSS.

On nous impose des styles par défaut pour certaines balises, mais ces styles ne correspondent pas forcément à ce que l'on souhaite !

Bien entendu, nous pouvons par exemple redéfinir le style de notre h1 :

```
heritage.css
```

```
h1{
    text-align: center; /* alignement */
    font-size: 20px; /* taille du texte */
    font-weight: bold; /* gras */
    margin: 10px 9px 8px 7px; /* marges ext. */
    padding: 20px; /* marges int. */
    border: 1px solid; /* bordure */
}
```

Cela peut être contraignant de toujours revenir sur les styles hérités par défaut.

C'est la raison pour laquelle il est souvent utile d'introduire un **RESET CSS** de manière à ré-initialiser tous les styles à 0.

48

Reset CSS

Pour effectuer un **RESET CSS** et ré-initialiser les styles, deux solutions sont possibles :

1ère solution avec l'appel de la plupart des balises existantes devant être remise à 0 :

```
html, body, div, span, applet, object, iframe, h1, h2, h3, h4, h5, h6, p, blockquote, pre, a, abbr, acronym, address, big, cite, code, del, dfn, em, img, ins, kbd, q, s, samp, small, strike, strong, sub, sup, tt, var, b, u, i, center, dl, dt, dd, ol, ul, li, fieldset, form, label, legend, table, caption, tbody, tr, th, td, article, aside, canvas, details, embed, figure, figcaption, footer, header, hgroup, menu, nav, output, ruby, section, summary, time, mark, audio, video {
    margin: 0;
    padding: 0;
    border: 0;
    font-size: 100%;
    font: inherit;
    vertical-align: baseline;
}
/* Pour les anciens navigateurs, cela permet de cibler les balises Html5 et de les rendre "block" */
article, aside, details, figcaption, figure, footer, header, hgroup, menu, nav, section {
    display: block;
}
body {
    line-height: 1;
}
ol, ul {
    list-style: none;
}
blockquote, q {
    quotes: none;
}
blockquote:before, blockquote:after, q:before, q:after {
    content: '';
    content: none;
```

```
}

table {
    border-spacing: 0;
    border-collapse: collapse;
}
```

2ème solution, un raccourci consiste à appeler toutes les balises par l'intermédiaire du symbole "*" :

```
* {
    margin: 0;
    padding: 0;
    border: 0;
    font-size: 100%;
    font: inherit;
    vertical-align: baseline;
}
```

Le 2e exemple est plus rapide à écrire, mais plus long à exécuter pour le navigateur en terme de performance.

Si l'on souhaite appliquer le reset partout sauf sur une balise, cela ne sera pas possible avec le symbole "*" dans le 2e cas, tandis que dans le 1er cas nous pouvons retirer la balise que nous ne souhaitons pas concerner par le RESET.

Les images

49

Afficher une image dans une page web html ?

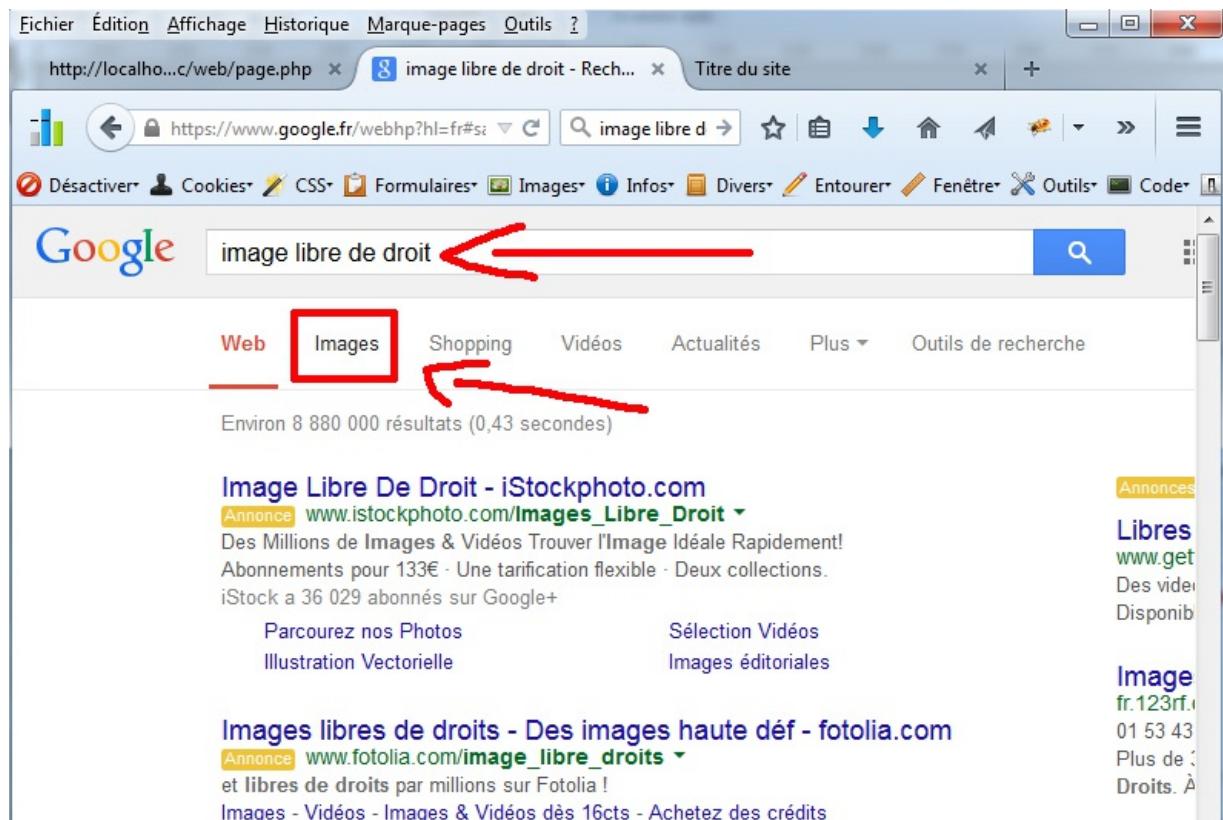
Nous allons créer un premier fichier nommé : image.html

image.html 16/04/2015 13:34 Firefox HTML Doc...

Pour afficher une image dans une page web, vous devez posséder un fichier image, de préférence l'extension doit être : .jpg, .png, .gif

Si vous n'avez pas d'image, vous ne pourrez rien afficher. ce qui paraît plutôt logique.

Pour récupérer une image, nous pouvons demander à notre ami Google de nous trouver cela :



Cliquer sur l'une des images :

The screenshot shows the Google Images search results for the query "image libre de droit". The interface includes a navigation bar with "Web", "Images" (which is highlighted in red), "Shopping", "Vidéos", "Actualités", "Plus", and "Outils de recherche". Below the search bar, there are three main categories of images: "Image Libre de Droit Gratuite" (free), "Image Libre de Droit Paysage" (landscape), and "Image Libre". The "Image Libre" category contains several images, including a large rock in the ocean, which is highlighted with a red border. Below this section, there are more image thumbnails, such as a person holding balloons and a city skyline.

Ensuite effectuer un clic droit > Enregistrer l'image sous.

This screenshot shows a right-click context menu over the image of the rock in the ocean. The menu items include: "Afficher l'image", "Copier l'image", "Copier l'adresse de l'image", "Enregistrer l'image sous...", "Partager l'image", "Envoyer l'image par courriel...", "Choisir l'image comme fond d'écran", "Informations sur l'image", "Examiner l'élément", "Web Developer", "FireShot", and "Inspecter l'élément avec Firebug".

Une fois que vous avez enregistré l'image, glissez la dans le même dossier que votre fichier html (juste à côté) :

	image.html	16/04/2015 13:34	Firefox HTML Doc...
	image.jpg	16/04/2015 13:34	Image JPEG

Si le nom de l'image est un peu compliqué, il est préférable de renommer le fichier image.

Exemple Html

```
image.html
?
<!Doctype html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Mon Image</title>
  </head>
  <body>
    
  </body>
</html>
```

Vous pouvez tester ce code, vous verrez qu'il fonctionne et affiche votre image.

Pour redimensionner et gérer la taille de votre image, l'idéal serait de passer par photoshop (logiciel de retouche d'image), sinon une option existe directement en code Html :

```

?
```

A vous de tester cela dans votre navigateur ! Exécution > Lancer dans Firefox.

i ➤ Votre image ne s'affiche pas ?

- Est-ce que votre fichier `image.html` est bien enregistré (`ctrl+s`) ?
- Est-ce que votre fichier `image.jpg` est bien placé au même endroit (dans le même dossier) que le fichier `image.html` ?
- Est-ce que le nom de votre fichier image est correct ?
- L'extension de votre fichier est bien ".jpg" et non pas ".jpeg" ou ".png" ou ".gif" ?

Le Résultat



Décryptons ensemble le code Html :

- la balise `` permet d'insérer une image en Html
- L'attribut `src=""` permet de préciser le chemin de l'image
- L'attribut `title=""` permet de préciser un titre pour l'image.
Vous ne l'avez pas vu s'afficher ? passez donc votre souris sur l'image !
- L'attribut `alt=""` permet de préciser un texte de remplacement. Ce texte s'affichera si l'image est manquante et en plus de contribuer à un bon référencement, cela aide les personnes malvoyantes, il faut donc toujours penser à ajouter cet attribut !
- L'attribut `width=""` permet de préciser la taille en largeur de votre image

- L'attribut `height=""` permet de préciser la taille en hauteur de votre image



➤ Résumé

Nous n'avons pas besoin de CSS si on souhaite afficher une image dans une page web mais nous pouvons utiliser du code CSS pour améliorer la mise en forme de l'image.

50

Afficher une image de fond dans une page web ?

Nous allons créer un premier fichier nommé : `fond.html`

Exemple Html :

```
fond.html
?
<!Doctype html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Mon Fond</title>
    <link rel="stylesheet" href="fond.css">
  </head>
  <body>

  </body>
</html>
```

Exemple Css :

```
fond.css
?
body{
background: #000 url(fond.jpg) no-repeat;
background-size: cover;
}
```

Le Résultat



Décryptons le code :

- `body` permet de sélectionner tout le corps du document.
- `background` permet d'appliquer un fond, nous choisissons une couleur noire `#000` + une image `fond.jpg` + `no-repeat` nous demandons à ce que cette image ne soit pas répétée dans le cas où elle ne recouvre pas toute la zone du navigateur.
- `background-size` est une propriété css3 permettant de couvrir tout l'écran, il s'agit d'un fond extensible, peut importe la taille et la résolution d'écran utilisées.

Faites le test !

Qu'elle est la différence entre intégrer une image avec la balise Html `` et appliquer une image en fond via la propriété `background` en Css ?

- Lorsqu'on utilise la propriété `background`, nous pourrons écrire par dessus
- Si nous n'avons pas besoin d'écrire par dessus une image, dans la majorité des cas nous utiliserons la balise ``

51

Gérer une opacité sur une image de fond ?

Pour gérer une opacité sur une image de fond, nous allons créer un premier fichier nommé : `fond2.html`

Pour afficher une image en fond d'une page web, vous devez posséder un fichier image, de préférence l'extension doit être : `.jpg`, `.png`, `.gif`

Exemple Html :

```
fond2.html
? ?

<!Doctype html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Mon Fond</title>
    <link rel="stylesheet" href="fond2.css">
  </head>
  <body>
    <div class="contenu">
      <p>Du texte...</p>
    </div>
  </body>
</html>
```

Exemple Css :

```
fond2.css
? ?

body{
background: #000000 url(fond1.jpg) no-repeat; /* image de fond */
background-size: cover; /* couverture du fond */
}
.contenu{
background-color: rgba(255, 255, 255, 0.3); /* atténuation, nous gardons 30% de la couleur blanche */
width: 800px; /* largeur */
height: 500px; /* hauteur */
margin: 200px auto 0 auto; /* marges ext*/
padding: 10px; /* marges int */
}
.contenu p{
position : relative; /* positionnement */
top: 50%; /* écartement de 50% du haut de la page */
text-align: center; /* alignement du texte */
}
```

Le Résultat



Décryptons ensemble le code :

- **body** permet de sélectionner tout le corps du document.
- **background** permet d'appliquer un fond, nous choisissons une couleur noire #000 + une image **fond.jpg** + **no-repeat** nous demandons à ce que cette image ne soit pas répétée dans le cas où elle ne recouvre pas toute la zone du navigateur.
- **background-size** est une propriété css3 permettant de couvrir tout l'écran, il s'agit d'un fond extensible, peut importe la taille et la résolution d'écran utilisées.
- **.contenu** permet de sélectionner une zone du document.
- **background-color: rgba(255, 255, 255, 0.3);** (255, 255, 255 correspond au code couleur blanc), le 0.3 permet d'atténuer et de garder seulement 30% de sa couleur.
Un autre moyen consiste à utiliser la propriété CSS **opacity** je vous la déconseille dans la mesure où cela atténuerait également la couleur du texte (et non pas que le fond).
- **width: 800px;** permet de fixer une largeur de 800 pixels.
- **height: 500px;** permet de fixer une hauteur de 500 pixels.
- **margin: 200px auto 0 auto;** permet de fixer les marges extérieures : 200pixels du haut, 0 en bas, automatique à gauche et à droite (permettant d'être centré horizontalement).
- **padding: 10px;** permet de fixer les marges intérieures : 10px s'applique tout autour (haut, droite, bas, gauche) dans le contenu de la zone
- **.contenu p** permet de sélectionner le paragraphe p à l'intérieur de la zone
- **position: relative;** permet de gérer le positionnement
- **top: 50%;** permet de s'écartier de 50% du haut de page
- **text-align: center;** permet de gérer l'alignement du texte

Exemple Html :

```
fond3.html
```

```
<!Doctype html>
<html>
<head>
<meta charset="utf-8">
<title>Mon Fond</title>
<link rel="stylesheet" href="fond3.css">
</head>
<body>
<div class="contenu">
<p>Du texte...</p>
</div>
</body>
</html>
```

Exemple Css :

```
fond3.css
```

```
.contenu{
background: #f6eadd; /* fond de couleur uni pour les anciens navigateurs */
background: linear-gradient( #f6eadd, #f9dac3); /* fond dégradé pour les nouveaux navigateurs qui prennent en charge cette valeur linear-gradient */
width: 800px; /* largeur */
height: 500px; /* hauteur */
margin: 0 auto; /* marges extérieures */
}

/* Autres possibilités :
/* background: linear-gradient( to left, red, blue); -- direction du dégradé */
/* background: -moz-linear-gradient(top, #f6eadd 0%, #f9dac3 100%); -- dégradé pour Firefox */
/* background: -webkit-gradient(linear, left top, left bottom, color-stop(0%,#f6eadd), color-stop(100%,#f9dac3)); -- dégradé pour GoogleChrome et Safari */
/* background: -o-linear-gradient(top, #f6eadd 0%,#f9dac3 100%); -- dégradé pour opera */
/* background: -ms-linear-gradient(top, #f6eadd 0%,#f9dac3 100%); -- dégradé pour InternetExplorer */
/* filter: progid:DXImageTransform.Microsoft.gradient( startColorstr="#f6eadd", endColorstr="#f9dac3",GradientType=0 ); -- dégradé pour IE6-9 */
```

Le Résultat

Nous remarquons ci-dessus le dégradé dans les couleurs orange !

53

Effet de transition

Voici un exemple de transition sur les images :

```
image2.html
```

```
<!Doctype html>
<html>
<head>
<meta charset="utf-8">
<title>Images</title>
<link rel="stylesheet" href="image2.css">
</head>
<body>
<div>



</div>
</body>
</html>
```

```
image2.css
```

```
div{ margin: 100px 0 0 0; } /* marges */
div img{
height: 100px; /* hauteur */
width: 100px; /* largeur */
border: 5px solid #ccc; /* bordure */
```

```

float: left; /* effet flottant (côte à côte) */
margin: 15px; /* marges */
transition: margin 0.5s; /* durée de transition lorsque l'on survole l'élément (hover) */
}
div img:hover {
margin-top: 2px; /* on modifie la marge de 2px vers le haut (au lieu de 15px initial) lors du survol */
}

```

Le Résultat (passer votre souris sur l'image)



Décryptons le code :

- `div{margin: 100px 0 0 0;}` permet d'ajouter 100pixels de marges
- `div img{ height: 100px;}` permet d'ajouter une hauteur aux images
- `div img{ width: 100px;}` permet d'ajouter une largeur aux images
- `div img{ border: 5px solid #ccc;}` permet d'ajouter une bordure aux images
- `div img{ float: left;}` permet de donner un effet "flottant" côté à côté aux images
- `div img{ margin: 15px;}` permet d'ajouter de la marge extérieure autour des images
- `div img{ transition: margin 0.5s;}` permet de donner un effet "progressif" dans la transition des margins sur les images
- `div img:hover{ margin-top: 2px;}` permet de donner un effet différent lors du survol sur l'image avec le curseur de la souris

Effet grossissant

Voici un exemple de transition sur les images :

image3.html

```

<!Doctype html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Image</title>
    <link rel="stylesheet" href="image3.css">
  </head>
  <body>
    <div>
      
      
      
    </div>
  </body>
</html>

```

image3.css

```

img{
  height: 100px; /* hauteur */
  width: 200px; /* largeur */
  margin: 15px 0; /* marges ext. */
  transition: 1s; /* effet progressif de transition : 1 seconde */
}
img:hover { /* la taille est augmentée lors du survol et l'image décalée par le margin */
  height: 130px;
  width: 300px;
}

```

Le Résultat (passer votre souris sur l'image)



Décryptons le code :

- `img{ height: 100px; }` permet d'ajouter une hauteur aux images
- `img{ width: 200px; }` permet d'ajouter une largeur aux images
- `img{ margin: 15px 0; }` permet d'ajouter de la marge extérieure en haut et en bas des images
- `img{ transition: transition: 1s; }` permet de donner un effet "progressif" dans la transition de la taille sur les images : 1 seconde
- `div img:hover{ height: 130px; }` permet de donner un effet différent lors du survol sur l'image avec le curseur de la souris
- `div img:hover{ width: 400px; }` permet de donner un effet différent lors du survol sur l'image avec le curseur de la souris

Effet de transition - Image - 4

Voici un exemple de transition sur les images :

```
image4.html
?
<!Doctype html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Image</title>
    <link rel="stylesheet" href="image4.css">
  </head>
  <body>
    <div>
      <p>Du texte d'accompagnement</p>
    </div>
  </body>
</html>
```

```
image4.css
?
div {
  height: 133px; /* hauteur */
  width: 730px; /* largeur */
  line-height: 0px; /* alignement dans la hauteur (du texte) */
  color: transparent; /* couleur transparente */
  font-size: 50px; /* taille du texte */
  text-transform: uppercase; /* transformation du texte en majuscule */
  transition: 0.5s; /* durée de transition */
}
div:hover {
  line-height: 210px; /* alignement dans la hauteur (du texte) */
  color: #575858; /* couleur */
}
div img{
  float: left; /* effet flottant (côte à côte) */
  margin: 0 15px; /* marge */
}
```

Le Résultat



Texte d'accompagnement

Décryptons le code :

- `div{ height: 133px; }` permet d'ajouter une hauteur
 - `div{ width: 730px; }` permet d'ajouter une largeur
 - `div{ line-height: 0px; }` alignement vertical (du texte)
 - `div{ color: transparent; }` couleur transparente
 - `div{ font-size: 50px; }` taille du texte
 - `div{ text-transform: uppercase; }` transformation du texte en majuscule
 - `div{ transition: 0.5s; }` durée de transition
-
- `div:hover{ line-height: 210px; }` Lors du survol, alignement dans la hauteur (du texte)
 - `div:hover{ color: #575858; }` Lors du survol, changement de couleur
-
- `div img{ float: left; }` effet flottant (côte à côte)
 - `div img{ margin: 0 15px; }` marges extérieures

Effet de rotation

Voici un exemple de transition sur les images :

```
image5.html
<!Doctype html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Image</title>
    <link rel="stylesheet" href="image5.css">
  </head>
  <body>
    <div>
      
      
      
    </div>
  </body>
</html>
```

```
image5.css
div img {
  margin: 20px; /* marges */
  border: 5px solid #eee; /* bordure */
  box-shadow: 4px 4px 4px rgba(0,0,0,0.2); /* ombre */
  transition: 0.5s; /* transition all : sur toutes les propriétés */
}
div img:hover {
```

```
transform: rotate(-7deg); /* rotation 7 degrés */  
}  
div img.rotation360:hover {  
    transform: rotate(-360deg); /* rotation 360 degrés */  
}
```

Le Résultat



Décryptons le code :

- `div img { margin: 20px; }` marges extérieures
- `div img { border: 5px solid #eee; }` bordure
- `div img { box-shadow: 4px 4px 4px rgba(0,0,0,0.2); }` ombre
- `div img { transition: 0.5s; }` effet de transition
- `div img:hover{ transform: rotate(-7deg); }` rotation 7 degrés
- `div img.rotation360:hover{ transform: rotate(-360deg); }` rotation 360 degrés

Les positionnements

54

Centrer son site sur la page

Dans cet exemple, nous allons apprendre à centrer une zone sur une page (1 site est avant tout une zone centrée sur une page, comprenant d'autres zones).

Pour cela, nous allons créer deux nouveaux fichiers : `centre.html` & `centre.css`

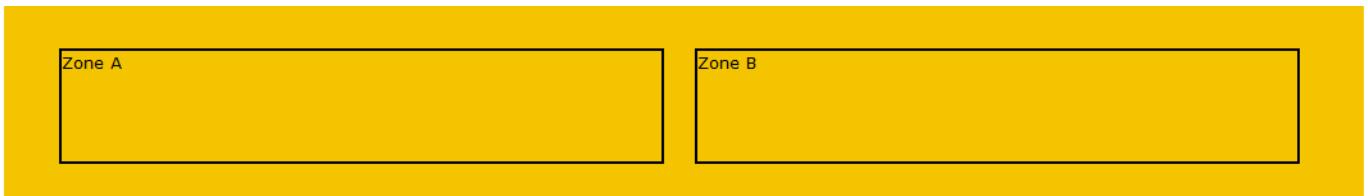
centre.html

```
<!Doctype html>  
<html>  
    <head>  
        <meta charset="utf-8">  
        <title>Centre</title>  
        <link rel="stylesheet" href="centre.css">  
    </head>  
    <body>  
        <div id="conteneur">  
            <div class="a">Zone A</div>  
            <div class="b">Zone B</div>  
        </div>  
    </body>  
</html>
```

centre.css

```
#conteneur{  
width: 960px; /* largeur */  
background: #F4C400; /* fond */  
margin: 0 auto; /* marges */  
padding: 30px; /* espacement interne du conteneur */  
}  
.a, .b{  
height: 150px; /* hauteur */  
width: 450px; /* largeur */  
border: 2px solid; /* bordure */  
margin: 0 10px; /* espacement horizontal entre les blocs */  
display: inline-block; /* inline block rend le vertical align possible. */  
vertical-align: middle; /* alignement vertical des zones a et b par rapport à #conteneur.  
}
```

Le Résultat



Décryptons le code :

- `#conteneur{ width: 960px; } largeur`
- `#conteneur{ background: #F4C400; } fond`
- `#conteneur{ margin: 0 auto; } marges`
- `#conteneur{ padding: 30px; } espace interne du conteneur`
- `.a, .b{ height: 150px; } hauteur`
- `.a, .b{ width: 450px; } largeur`
- `.a, .b{ border: 2px solid; } bordure`
- `.a, .b{ display: inline-block; } dans ce contexte, inline-block rend le vertical align possible`
- `.a, .b{ vertical-align: middle; } alignement vertical des zones a et b par rapport à #conteneur.`

Qu'est-ce qu'un conteneur (anglais : container, parfois écrit wrap (enveloppe)) ?

Un conteneur permet de contenir !

C'est une pratique régulièrement utilisée pour centrer son site sur une page web. Et, puisqu'on applique un centrage sur le conteneur, par répercussion, tout ce qui se trouve à l'intérieur sera également centré.

55

Pixels ou Pourcentage ?

Voyons à présent la différence dans les unités de mesure entre les pixels et les pourcentages

Pour cela, nous allons créer deux nouveaux fichiers : `pourcentage.html` & `pourcentage.css`

```
pourcentage.html
?
<!Doctype html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Pourcentage</title>
    <link rel="stylesheet" href="pourcentage.css">
  </head>
  <body>
    <div class="taillefixe">Fixe</div>
    <div class="taillevariable">Variable</div>
  </body>
</html>
```

```
pourcentage.css
?
.taillefixe{ width: 1000px; border: 2px solid #cb3d3d; }
.taillevariable{ width: 100%; border: 2px solid #336699; }
```

Le Résultat

Zone A

Zone B

Faites le test !

Décryptons le résultat :

- Dans un cas, la zone 1 (div) fixe rouge prend 1000 pixels de largeur qui est une taille fixe.
- Dans l'autre cas, la zone 2 (div) variable bleue prend 100% de la largeur, cette largeur est donc variable selon le redimensionnement de la fenêtre qui la contient.

En [responsive design](#) (site adapté aux différentes résolutions et écrans : smartphone, tablette, pc), il peut être intéressant de créer son site en pourcentage ou en pixels en appliquant des cas différents selon les tailles de résolutions d'écran utilisés.

56

La position absolute

Nous allons étudier la position absolute.

Pour cela, nous allons créer deux nouveaux fichiers : **absolute.html** & **absolute.css**

absolute.html

```
<!Doctype html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Absolute</title>
    <link rel="stylesheet" href="absolute.css">
  </head>
  <body>
    <p class="vert"> Bonjour ! </p>
  </body>
</html>
```

absolute.css

```
p{ margin: 0; }
.vert{ color: #21c165; position: absolute; top: 50%; left: 0px; font-weight: bold; }
```

Le Résultat

Bonjour !

Faites le test !

Décryptons le code :

La position absolute redéfinit l'emplacement de la zone par rapport à la fenêtre.

Ainsi en utilisant **top: 50%; left: 0px;** Nous écartons notre zone de 50% du haut de site et de 0 pixel du bord gauche de la fenêtre.

Exercice : Compte tenu de ces informations, pouvez-vous placer cette zone en bas à droite de l'écran ?

Correction :

```
p{ margin: 0; }
.vert{ color: #21c165; position: absolute; bottom: 0; right: 0; font-weight: bold; }
```

57

La position relative

Nous allons étudier la position relative. Pour cela, nous allons créer deux nouveaux fichiers : **relative.html** & **relative.css**

```
relative.html
```

```
<!Doctype html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Relative</title>
    <link rel="stylesheet" href="relative.css">
  </head>
  <body>
    <div id="zone">
      <p> Bonjour tout le monde </p>
      <p class="vert"> Salut! </p>
    </div>
  </body>
</html>
```

```
relative.css
```

```
p{ margin: 0; }
.vert{ color: #21c165; position: relative; top: 1px; left: 30px; }
#zone{ border: 2px solid #336699; }
```

Le Résultat

Bonjour tout le monde

Salut !

Décryptons le code :

Lorsque la position relative est appliquée sur un élément, cet élément se place ou s'écarte par rapport à lui même, il s'écarte par rapport à sa position initiale (et non pas par rapport à la div dans laquelle l'élément est contenu, ni le bord de la fenêtre).

Ainsi en utilisant `top: 1px; left: 30px;` Nous l'écartons de 1 pixel du haut par rapport à sa position initiale et de 30 pixels de la gauche par rapport à sa position initiale. Faites des tests !

58

La position fixed

La position fixed est particulièrement pratique et à la mode sur certains sites. Le plus connu d'entre eux : Facebook.

Vous ne l'avez jamais remarqué ? Lorsque vous naviguez dans les actualités et que vous rentrez dans la profondeur des pages, le menu reste constamment affiché en haut de site.

Pour étudier ce cas, nous allons créer deux nouveaux fichiers : **fixed.html** & **fixed.css**

```
fixed.html
```

```
<!Doctype html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Fixed</title>
    <link rel="stylesheet" href="fixed.css">
  </head>
  <body>
    <nav>Menu</nav>
    <div id="contenu">
      *** SAISIR (ou copier/coller :) 100 LIGNES DE TEXTE ICI, CELA EST IMPERATIF POUR CET EXEMPLE ***
    </div>
  </body>
</html>
```

```
fixed.css
```

```
nav{
background: #dea5a5; /* fond */
position: fixed; /* Le menu a une position fixe lors de la descente dans la page */
text-align: center; /* alignement du texte centré */
width: 100%; /* largeur 100% de la fenêtre */
top: 0; /* aligné à 0px du haut */
```

```

left: 0; /* aligné à 0px de la gauche */
}
#contenu{
border: 2px solid #336699; /* bordure */
margin: 30px 0 0 0; /* marges haut droite bas gauche */
}

```

Le Résultat

The screenshot shows a sidebar menu with a red header labeled "Menu". The main content area is white and contains text about the history of Lorem Ipsum, mentioning its roots in a Latin classical work from 45 BC and its popularity during the Renaissance. It also notes the standard extract used since the 16th century.

Faites le test !

Décryptons le code :

La position fixed permet de maintenir une zone toujours affichée !

Ainsi en utilisant `position: fixed; top: 0; left: 0;` cela permet de « figer » la zone en haut de site en la plaçant à 0 pixel du haut et à 0 pixel de la gauche. Le contenu est quant à lui écarté de 30 pixels du haut pour éviter qu'il vienne en "collision" et en dessous de notre zone menu.

59

La propriété float

Nous allons étudier les positions float avec `float left`, `float right`, `float center` ainsi que la propriété clear avec `clear left`, `clear right` et `clear both`.

La propriété float left est particulièrement utile pour placer deux zones côté à côté. (ce qui arrive fréquemment dans un site web).

Pour cela, nous allons créer deux nouveaux fichiers : `float.html` & `float.css`

float.html

```

<!Doctype html>
<html>
<head>
<meta charset="utf-8">
<title>Float</title>
<link rel="stylesheet" href="float.css">
</head>
<body>
<div class="d1"> 1 </div>
<div class="d2"> 2 </div>
<div class="d3"> 3 </div>
<div class="d4"> 4 </div>
<div class="clear"></div>
<div class="d5"> 5 </div>
<div class="d6"> 6 </div>
<div class="d7"> 7 </div>
<div class="clear"></div>
<div class="d8"> 8 </div>
<div class="d9"> 9 </div>
<div class="clear"></div>
<div class="d10"> 10 </div>
</body>
</html>

```

float.css

```

.d1 ,.d2 ,.d3 ,.d4 ,.d5 ,.d6 ,.d7 ,.d8 ,.d9 ,.d10 {
float: left;
width: 170px;
height: 170px;
text-align: center;
border: 1px solid #000000;
}
.clear{
clear: both;
}

```

Le Résultat

1	2	3	4
5	6	7	
8	9		
10			

Faites le test !

Décryptons le code :

La propriété `float left` permet de créer un flux qui va permettre d'empiler les zones côté à côté.

FLUX FLOTTANT

`float: left;`

`float: left;`

Tout ce qui doit être "à côté de" doit posséder la propriété `float: left;`

Nous attribuons donc la propriété `float: left;` à toutes les zones ayant besoin d'être à côté l'une de l'autre.

Une fois qu'un flux est ouvert, nous devons l'arrêter avec la propriété `clear: both;`, cela permet (en quelque sorte) de repasser à la ligne

Notez bien que nous ferons toujours appel à la propriété `clear both après un float left` pour arrêter le flux, cela nous évitera d'avoir des dysfonctionnements d'affichage.



➤ Question : pourquoi on ne met pas un coup `float left` et l'autre coup `float right` ?

Lorsqu'on utilise la propriété `float`, nous ouvrons un flux, nous devons préciser si nous l'ouvrons de la gauche vers la droite (`float: left;`) ou de la droite vers la gauche.

C'est moins courant mais nous pouvons avoir une zone en `float left` et une autre en `float right` aux deux extrémités, cela serait tout à fait valable.

Exemple avec `float : left;`:

```
float-left.html
??
<!Doctype html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Float</title>
    <link rel="stylesheet" href="float-left.css">
  </head>
  <body>
    <div class="d1"> 1 </div>
    <div class="d2"> 2 </div>
    <div class="clear"></div>
  </body>
```

```
</html>
```

float-left.css

```
.d1,.d2{  
    float: left;  
    width: 170px;  
    height: 170px;  
    text-align: center;  
    border: 1px solid #000000;  
}  
.clear{  
    clear: both;  
}
```

Le Résultat

1	2
---	---

Ici les éléments s'affichent correctement, côté à côté et de la gauche vers la droite.

Exemple avec float : right;

float-right.html

```
<!Doctype html>  
<html>  
    <head>  
        <meta charset="utf-8">  
        <title>Float</title>  
        <link rel="stylesheet" href="float-right.css">  
    </head>  
    <body>  
        <div class="d1"> 1 </div>  
        <div class="d2"> 2 </div>  
        <div class="clear"></div>  
    </body>  
</html>
```

float-right.css

```
.d1,.d2{  
    float: right;  
    width: 170px;  
    height: 170px;  
    text-align: center;  
    border: 1px solid #000000;  
}  
.clear{  
    clear: both;  
}
```

Le Résultat

2	1
---	---

Ici les éléments s'affichent correctement, côté à côté mais cette fois-ci de la droite vers la gauche.

Et oui, on voit nettement que les cadres sont le plus à droite de l'écran.

Exemple avec float : left; et float : right;

float-left-right.html

```
<!Doctype html>  
<html>  
    <head>  
        <meta charset="utf-8">  
        <title>Float</title>  
        <link rel="stylesheet" href="float-left-right.css">  
    </head>  
    <body>  
        <div class="d1"> 1 </div>  
        <div class="d2"> 2 </div>  
        <div class="clear"></div>  
    </body>  
</html>
```

float-left-right.css

```
.d1{  
    float: left;  
    width: 170px;  
    height: 170px;
```

```
text-align: center;
border: 1px solid #000000;
}
.dz{
float: right;
width: 170px;
height: 170px;
text-align: center;
border: 1px solid #000000;
}
.clear{
clear: both;
}
```

Le Résultat

1

2

Dans ce résultat, on voit très nettement la différence ! la 1ère zone se trouve le plus à gauche possible et la 2e se trouve la plus à droite de l'écran.

A moins de vouloir cet affichage pour des besoins particuliers, nous garderons l'utilisation du float left (ou du float right) sur plusieurs éléments d'affiliés.

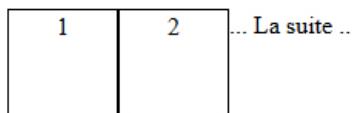
Exemple avec et sans la propriété clear: both;



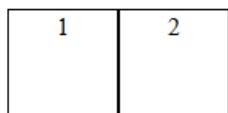
➤ Question

je n'ai pas bien compris l'utilité de la propriété clear both, a quoi ça sert ?

Sans la propriété clear : both ;



Avec la propriété clear : both ;



... La suite ...

On voit clairement que la propriété `clear: both;` permet de stopper un flux et de passer à la ligne, il est donc vraiment recommandé de l'utiliser pour éviter des dysfonctionnements d'affichage.

Zoning

60

Démonstration : Zoning et modèle de mise en page

Si l'on souhaite créer un site web, il va falloir apprendre à créer des zones (un zoning).

Avant d'intégrer des images ou encore de saisir du texte, nous avons besoin de définir des emplacements pour un positionnement précis de nos informations.

Pour cela, nous allons créer deux nouveaux fichiers : `modele1.html` & `modele1.css`

Pour mieux comprendre, prenons un cas d'étude, imaginons que nous souhaitons viser ce rendu graphique :

Zone du Haut

Zone de Gauche

Zone du Milieu

Voici le code correspondant.

```
<!Doctype html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Mon Modele</title>
    <link rel="stylesheet" href="modele1.css">
  </head>
  <body>
    <div id="conteneur">
      <header>
        <p> Zone du Haut</p>
      </header>
      <div class="gauche">
        <p> Zone de Gauche</p>
      </div>
      <div class="milieu">
        <p> Zone du Milieu</p>
      </div>
      <div class="clear"></div> <!-- on arrete l'effet flottant -->
    </div>
  </body>
</html>
```

modele1.html

```
#conteneur{
background: #336699; width: 800px;
}
header{
background: #ff5555; width: 800px; height: 70px;
}
.gauche{
background: #5585ff; width: 150px; height: 530px; float: left;
}
.milieu{
background: #55D87C; width: 650px; height: 530px; float: left;
}
.clear{
clear: both;
}
```

modele1.css

Maintenant que vous avez pris des forces, nous allons nous entraîner avec plusieurs exercices.

Si vous comprenez le code c'est déjà pas mal mais il ne suffit pas de savoir le lire, il faut également savoir le créer.

Exercice, reproduire cette image :

Conseil : aidez-vous de l'extension firebug



Voici la correction :

```
model2.html

<!Doctype html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Mon Modèle</title>
    <link rel="stylesheet" href="modele2.css">
  </head>
  <body>
    <div id="conteneur">
      <header>
        <p> Zone du Haut</p>
      </header>
      <div class="gauche">
        <p> Zone de Gauche</p>
      </div>
      <div class="milieu">
        <p> Zone du Milieu</p>
      </div>
      <div class="droite">
        <p> Zone de Droite</p>
      </div>
      <div class="clear"></div>
    </div>
  </body>
</html>
```

modele2.css

```
#conteneur{  
background: #336699; width: 800px;  
}  
header{  
background: #ff5555; width: 800px; height: 70px;  
}  
.gauche{  
background: #5585ff; width: 150px; height: 530px; float: left;  
}  
.milieu{  
background: #55D87C; width: 500px; height: 530px; float: left;  
}  
.droite{  
background: #9f6f3f; width: 150px; height: 530px; float: left;  
}  
.clear{  
clear: both;  
}
```

62

Exercice : Zoning et modèle de mise en page - 3

Exercice, reproduire cette image :



Voici la correction :

modele3.html

```
<!Doctype html>  
<html>  
  <head>  
    <meta charset="utf-8">  
    <title>Mon Modèle</title>  
    <link rel="stylesheet" href="modele3.css">  
  </head>  
  <body>  
    <div id="conteneur">  
      <div class="gauche">gauche</div>  
      <div class="droite">droite</div>  
      <div class="clear"></div>  
      <div>Bas</div>  
    </div>  
  </body>  
</html>
```

modele3.css

```
#conteneur{ width: 1000px; margin: 0 auto; }  
.gauche{ background: #40A497; height: 400px; width: 500px; float: left; }
```

```
.droite{ background: #9A40A4; height: 400px; width: 500px; float: left; }
footer{ background: #F0C000; height: 80px; width: 1000px; }
.clear{ clear: both; }
```

63

Exercice : Zoning et modèle de mise en page - 4

Exercice, reproduire cette image :



Voici la correction :

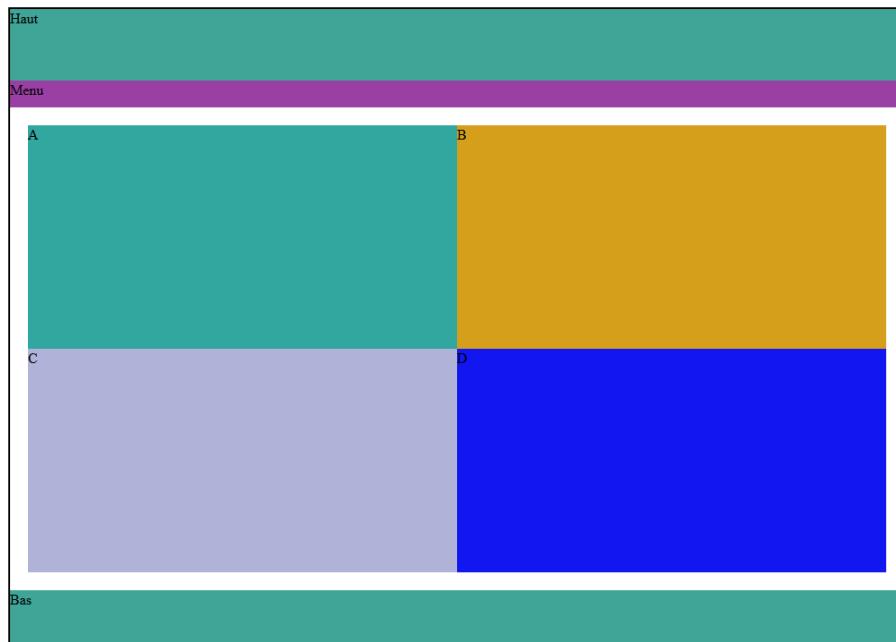
modele4.html

```
<!Doctype html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Mon Modele</title>
    <link rel="stylesheet" href="modele4.css">
  </head>
  <body>
    <div id="conteneur">
      <header>Haut</header>
      <nav>Menu</nav>
      <section>Milieu</section>
      <footer>Bas</footer>
    </div>
  </body>
</html>
```

modele4.css

```
#conteneur{ width: 1000px; margin: 0 auto; border: 2px solid; }
header{ background: #40A497; height: 80px; }
nav{ background: #9A40A4; height: 30px; }
section{ background: #F0C000; height: 500px; margin: 20px; }
footer{ background: #40A497; height: 60px; }
.clear{ clear: both; }
```

Exercice, reproduire cette image :



Voici la correction :

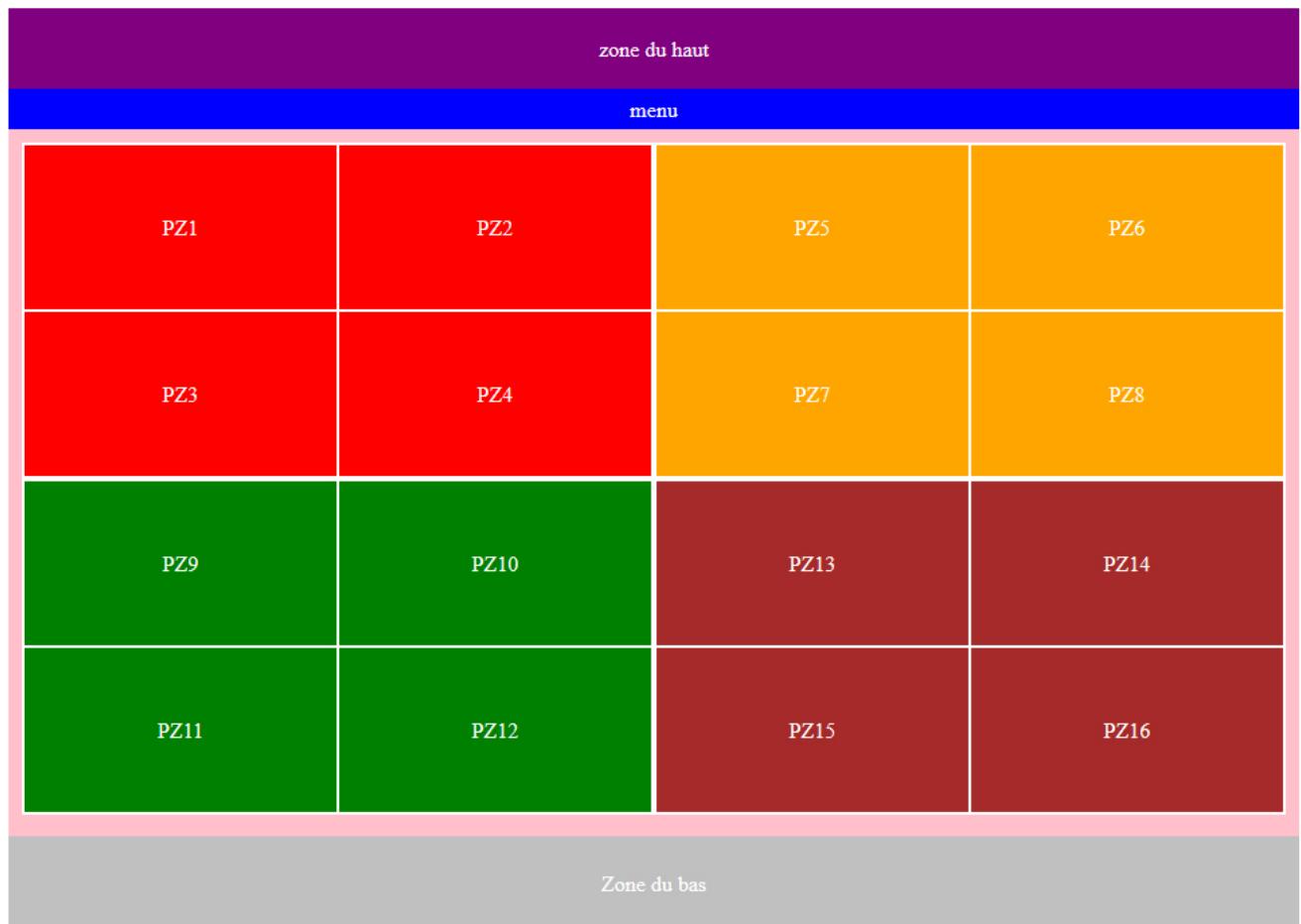
```
model5.html
```

```
<!Doctype html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Mon Modele</title>
    <link rel="stylesheet" href="modele5.css">
  </head>
  <body>
    <div id="conteneur">
      <header>Haut</header>
      <nav>Menu</nav>
      <section>
        <div class="a">A</div>
        <div class="b">B</div>
        <div class="c">C</div>
        <div class="d">D</div>
      </section>
      <footer>Bas</footer>
    </div>
  </body>
</html>
```

```
modele5.css
```

```
#conteneur{ width: 1000px; margin: 0 auto; border: 2px solid; }
header{ background: #40A497; height: 80px; }
nav{ background: #9A40A4; height: 30px; }
section{ background: #F0C000; height: 500px; margin: 20px; }
.a{ background: #32a7a0; height: 250px; width: 480px; float: left; }
.b{ background: #d69f1b; height: 250px; width: 480px; float: left; }
.c{ background: #b1b2d8; height: 250px; width: 480px; float: left; }
.d{ background: #1217f2; height: 250px; width: 480px; float: left; }
footer{ background: #40A497; height: 60px; }
.clear{ clear: both; }
```

Exercice, reproduire cette image :



Voici la correction :

modele6.html

```
<!Doctype html>
<html>
<head>
  <meta charset="utf-8">
  <title>Mon Modele</title>
  <link rel="stylesheet" href="modele6.css">
</head>
<body>
  <div id="conteneur">
    <header>
      <p>zone du haut</p>
    </header>
    <nav>menu</nav>
    <section>
      <div class="GZ_couleur_rouge">
        <div class="PZ">PZ1</div>
        <div class="PZ">PZ2</div>
        <div class="PZ">PZ3</div>
        <div class="PZ">PZ4</div>
      </div>
      <div class="GZ_couleur_jaune">
        <div class="PZ">PZ5</div>
        <div class="PZ">PZ6</div>
        <div class="PZ">PZ7</div>
        <div class="PZ">PZ8</div>
      </div>
      <div class="GZ_couleur_verte">
        <div class="PZ">PZ9</div>
        <div class="PZ">PZ10</div>
        <div class="PZ">PZ11</div>
        <div class="PZ">PZ12</div>
      </div>
      <div class="GZ_couleur_marron">
        <div class="PZ">PZ13</div>
        <div class="PZ">PZ14</div>
        <div class="PZ">PZ15</div>
        <div class="PZ">PZ16</div>
      </div>
    </section>
  </div>
</body>
```

```

<p>Zone du bas</p>
</footer>
</div>
</body>
</html>

```

modele6.css

```

#conteneur{ width: 960px; background: pink; color: white; margin: 0 auto; }
header { background: purple; height: 60px; line-height: 60px; text-align: center; }
nav { background: blue; height: 30px; line-height: 30px; text-align: center; }
section{ height: 500px; background: gray; margin: 10px; }
.GZ { border: 1px solid; float: left; height: 248px; width: 468px; }
.PZ { border: 1px solid; float: left; height: 122px; line-height: 122px; text-align: center; width: 232px; }
.couleur_rouge{ background: red; }
.couleur_jaune{ background: orange; }
.couleur_verte{ background: green; }
.couleur_marron{background: brown; }
footer { background: silver; height: 70px; line-height: 70px; text-align: center; }

```

66

Exercice : Zoning et modèle de mise en page - 7

Exercice, reproduire cette image :



Voici la correction :

modele7.html

```

<!Doctype html>
<html>
<head>
<meta charset="utf-8">
<title>Mon Modele</title>
<link rel="stylesheet" href="modele7.css">
</head>
<body>
<div id="conteneur">
<header>
<p> Zone du Haut</p>
</header>
<div class="gauche">
<p> Zone de Gauche</p>
</div>
<div class="milieu">
<div class="z_milieu">
Z1
</div>
<div class="z_milieu">
Z2
</div>
<div class="z_milieu">
Z3
</div>
<div class="z_milieu">
Z4
</div>
<div class="z_milieu">
Z5
</div>
</div>
<div class="droite">
<p> Zone du Droite</p>
</div>
<div class="clear"></div>
<div class="bas">
<p>Zone du bas</p>
</div>
<div class="clear"></div>
</div>
</body>
</html>

```

modele7.css

```

html, body, div, span, applet, object, iframe, h1, h2, h2, h4, h5, h6, p, blockquote, pre, a, abbr, acronym, address, big, cite, code, del, dfn, em, img, ins, kbd, q, s, samp, small, strike, strong, sub, sup, tt, var, b, u, i, center, dl, dt, dd, ol, ul, li, fieldset, form, label, legend, table, caption, tbody, tfoot, thead, tr, th, td, article, aside, canvas, details, embed, figure, figcaption, footer, header, hgroup, menu, nav, output, ruby, section, summary, time, mark, audio, video {
margin: 0; padding: 0; border: 0; font-size: 100%; font: inherit; vertical-align: baseline; text-align: center; color: white; }
#conteneur { background: none repeat scroll 0 0 #336699; height: 600px; margin: 0 auto; width: 800px; }
header{ background: #ff5555; width: 800px; height: 70px; line-height: 70px; }

```

```

.gauche{ background: #5586ff; width: 150px; height: 530px; float: left; line-height: 530px; }
.droite{ background: #5586ff; width: 150px; height: 530px; float: left; line-height: 530px; }
.milieu { background: #55d87c; float: left; height: 450px; line-height: 530px; width: 500px; }
.z_milieu { border: 1px solid; float: left; height: 448px; width: 98px; }
.bas { background: gold; bottom: 80px; height: 80px; line-height: 80px; left: 150px; position: relative; width: 500px; }
.clear{ clear: both; }

```

Les tableaux

67

Les Tableaux

Les tableaux permettent de représenter des données tabulaires, des statistiques, des listes d'informations en BackOffice, et des newsletters.

Nous ne construirons pas tout un site web avec des tableaux mais pour des raisons historiques de compatibilité les [newsletters / emailing](#) (page web envoyé par email) sont construits à l'aide de tableaux.

Pour créer une table il faut réfléchir en ligne et en cellule (de gauche à droite) et non pas en colonne (de haut en bas).

Voici un premier exemple : un bulleting de notes.

table1.html

```

<!Doctype html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Ma Table</title>
    <link rel="stylesheet" href="table1.css">
  </head>
  <body>
    <table border="1">
      <tr> <!-- TR : nouvelle ligne -->
        <th>Matière</th><!-- TH : nouvelle cellule d'entête -->
        <th>Moyenne</th>
        <th>Coef</th>
        <th>Formateur</th>
        <th>Commentaire</th>
      </tr> <!-- Fin de ligne -->

      <tr> <!-- nouvelle ligne -->
        <td>Html</td> <!-- TD : nouvelle cellule -->
        <td>13</td>
        <td>1</td>
        <td>LT</td>
        <td>Ok !</td>
      </tr><!-- Fin de ligne -->

      <tr> <!-- nouvelle ligne -->
        <td>Référencement</td>
        <td>10</td>
        <td>1</td>
        <td>XX</td>
        <td>Moyen :-(</td>
      </tr><!-- Fin de ligne -->

      <tr> <!-- nouvelle ligne -->
        <td>Droit</td>
        <td>12</td>
        <td>1</td>
        <td>XX</td>
        <td>Pas mal !</td>
      </tr><!-- Fin de ligne -->

      <tr> <!-- nouvelle ligne -->
        <td>Marketing</td>
        <td>12</td>
        <td>1</td>
        <td>XX</td>
        <td>Pas mal !</td>
      </tr><!-- Fin de ligne -->

      <tr> <!-- nouvelle ligne -->
        <td>Pub</td>
        <td>12</td>
        <td>1</td>
        <td>XX</td>
        <td>Pas mal !</td>
      </tr><!-- Fin de ligne -->
      <tr>
        <td>Général</td>
        <td colspan="4">Avis Favorable !!!</td>
      </tr>
    </table> <!-- Fin de table -->

```

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

Décryptons le code Html :

- TABLE est un tableau
- TR est une nouvelle ligne
- TH est une cellule d'entête
- TD est une cellule

table1.css

```
td, th{
  text-align: center;
}
tr.changementCouleur:hover{
  background: #f43c80;
}
```

Décryptons le code Css :

- td, th{} permet de cibler et sélectionner toutes les balises td et th. Nous alignons le texte au centre de la cellule grâce à la propriété text-align: center;
 - tr.changementCouleur:hover{} permet de cibler toutes les balises <TR> possédant la classe "changementCouleur" en effet survolé (:hover) afin d'appliquer un style différent lors du passage de la souris sur l'une des lignes du tableau.
- Essayer de passer votre souris sur le tableau pour mieux le comprendre !

Le Résultat

Matière	Moyenne	Coef	Formateur	Commentaire
Intégration : Html / Css	15	1	XX	Ok !
Référencement	10	1	XX	Moyen :-)
Marketing	13	1	XX	Pas mal !
Pub	12	1	XX	Pas mal !
Général	Avis Favorable !!!			

Voici un autre exemple présentant des statistiques :

table2.html

```
<!Doctype html>
<html>
<head>
  <meta charset="utf-8">
  <title>Ma Table</title>
  <link rel="stylesheet" href="table2.css">
</head>
<body>
  <table>
    <tr> <!-- nouvelle ligne -->
      <th colspan="4">Statistiques</th> <!-- colspan fusionne les cellules horizontal -->
    </tr>

    <tr> <!-- nouvelle ligne -->
      <th>Pays</th> <!-- nouvelle cellule d'entête (en gras) -->
      <td>France</td> <!-- nouvelle cellule -->
      <td>Italie</td>
      <td>Espagne</td>
    </tr>

    <tr>
      <th>Nombre d'habitants</th> <!-- nouvelle cellule d'entête (en gras) -->
      <td>65 000 000</td>
      <td>60 157 214</td>
      <td>46 661 950</td>
    </tr>
    <tr>
      <th>Capitale</th> <!-- nouvelle cellule d'entête (en gras) -->
      <td>Paris</td>
      <td>Rome</td>
      <td>Madrid</td>
    </tr>
  </table>
</body>
</html>
```

```

</tr>
</table> <!-- Fin de table -->
</body>
</html>

```

Décryptons le code Html :

- TABLE est un tableau
- TR est une nouvelle ligne
- TH est une cellule d'entête
- TD est une cellule
- COLSPAN permet de fusionner des cellules dans le sens horizontal
- ROWSPAN permet de fusionner des cellules dans le sens vertical

table2.css

```

table{
border-collapse: collapse; /* trait plat */
width: 700px; /* largeur */
height: 100px; /* hauteur */
margin: 0 auto; /* marges */
}
th{
background: #f40c80; /* couleur de fond */
}
td, th{
padding: 7px; /* espacement intérieur */
border: 1px solid; /* bordure */
text-align: center; /* alignement du texte au centre */
}

```

Décryptons le code Css :

- table{} permet de cibler le tableau et de lui appliquer certaines propriétés : border-collapse permet d'aplatir le trait qui dessine le tableau, width c'est la largeur, height pour hauteur et margin 0 auto pour centrer le tableau dans la page web.
- th{} permet de cibler les cellules d'entête pour leur appliquer une couleur de fond particulière.
- td, th{} permet d'attraper toutes les cellules du tableau pour leur appliquer padding: 7px; de la marge intérieure, border: 1px solid une bordure et text-align: center; un alignement du texte au centre.

Le Résultat

Statistiques			
Pays	France	Italie	Espagne
Nombre d'habitants	65 000 000	60 157 214	46 661 950
Capitale	Paris	Rome	Madrid

Voici un autre exemple présentant une salle de cours (pour l'entraînement) :

table3.html

```

<!Doctype html>
<html>
<head>
<meta charset="utf-8">
<title>Ma Table</title>
<link rel="stylesheet" href="table3.css">
</head>
<body>
<table>
<caption>Plan de Classe</caption>
<tr>
<td colspan="2" class="vide"> </td>
<td colspan="2" class="vide"> Prof </td>
<td colspan="2" class="vide"> </td>

```

```

</tr>
<tr>
  <td class="vide" colspan="6" > </td>
</tr>
<tr>
  <td class="homme" > 1 </td>
  <td class="vide" > </td>
  <td class="femme" > 2 </td>
  <td class="homme" > 3 </td>
  <td class="vide" > </td>
  <td class="femme" > 4 </td>
</tr>
<tr>
  <td> 5</td>
  <td class="vide" > </td>
  <td> 6</td>
  <td> 7</td>
  <td class="vide" > </td>
  <td> 8 </td>
</tr>
<tr>
  <td> 9 </td>
  <td class="vide" > </td>
  <td> 10 </td>
  <td> 11 </td>
  <td class="vide" > </td>
  <td> 12 </td>
</tr>
<tr>
  <td> 13 </td>
  <td class="vide" > </td>
  <td> 14 </td>
  <td> 15 </td>
  <td class="vide" > </td>
  <td> 16 </td>
</tr>
<tr>
  <td> 17 </td>
  <td class="vide" > </td>
  <td> 18 </td>
  <td> 19 </td>
  <td class="vide" > </td>
  <td> 20 </td>
</tr>
</table>
</body>
</html>

```

Decryptons le code Html :

- TABLE est un tableau
- TR est une nouvelle ligne
- TH est une cellule d'entête
- TD est une cellule
- COLSPAN permet de fusionner des cellules dans le sens horizontal
- ROWSPAN permet de fusionner des cellules dans le sens vertical

table3.css

```

table{
  border: 1px solid;
}
td{
  width: 200px;
  height: 50px;
  text-align: center;
  border: 1px solid;
}
caption{
  font-weight: bold;
}
.vide{ border: 0; width: 100px; }
.homme{ background: #62A9F0 ; }
.femme{ background: #E52867 ; }

```

Decryptons le code Css :

- table{} permet de cibler le tableau et de lui appliquer certaines propriétés :border permet d'ajouter un trait pour dessiner le tableau.
- td{} permet de cibler les cellules pour leur appliquer une largeur width, hauteur height, alignement du texte text-align, une bordure border
- caption{} permet de cibler le titre du tableau pour appliquer du gras font-weight: bold;.

- `.vide{}` permet de donner un effet "vide" à certaines cellules pour dessiner les rangs de notre salle de classe.
- `.homme{}` permet de donner une couleur bleue à la place des Hommes dans la classe.
- `.femme{}` permet de donner une couleur rose à la place des Femmes dans la classe.

Le Résultat

Plan de Classe		
1	2	3
5	6	7
9	10	11
13	14	15
17	18	19
		4
		8
		12
		16
		20

Les listes

68

Les listes

Les listes en html peuvent être pratiques pour présenter des données. et en plus, c'est un chapitre facile ! :-).

Voici un premier exemple d'une liste :

```
liste1.html

<!Doctype html>
<html>
<head>
<meta charset="utf-8">
<title>Ma Page</title>
<link rel="stylesheet" href="liste4.css">
</head>
<body>

<ul> <!-- debut de la liste -->
<li>France</li> <!-- élément de la liste -->
<li>Italie</li> <!-- élément de la liste -->
<li>Espagne</li> <!-- élément de la liste -->
</ul> <!-- fin de la liste -->

<hr>
<!-- Il est possible d'imbriquer une liste dans une autre -->
<ul>
<li>Pays
<ul>
<li>France</li>
<li>Italie</li>
<li>Espagne</li>
</ul>
</li>
</ul>

</body>
</html>
```

Le Résultat

- France

- Italie
- Espagne
- Pays
 - France
 - Italie
 - Espagne

 permet de déclarer une nouvelle liste non ordonnée.

 représente les éléments à l'intérieur de la liste.

Comme vous pouvez le constater, il est possible d'imbriquer une liste dans une autre.

Exemple avec une liste ordonnée :

liste2.html

```
<!Doctype html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Ma Page</title>
    <link rel="stylesheet" href="liste4.css">
  </head>
  <body>

    <ol><!-- liste ordonnée -->
      <li>Html/Css</li>
      <li>Sql/Php</li>
      <li>Java/J2ee</li>
    </ol>

  </body>
</html>
```

Le Résultat

1. Html/Css
2. Sql/Php
3. Java/J2ee

La balise permet d'obtenir une liste numérotée.

Exemple avec une liste de termes :

liste3.html

```
<!Doctype html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Ma Page</title>
    <link rel="stylesheet" href="liste4.css">
  </head>
  <body>

    <dl><!-- Definition List -->
      <dt>F</dt> <!-- Definition Term -->
      <dd>France</dd> <!-- Definition Description -->
      <dd>Finland</dd> <!-- Definition Description -->
    </dl>

  </body>
</html>
```

Le Résultat

F

France
Finland

La balise <dl> permet de déclarer une liste de définitions.

La balise <dt> permet de déclarer un terme à expliquer.

La balise <dd> permet de fournir la description d'un terme.

Exemple d'une liste avec CSS

```
liste4.html
```

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Ma Page</title>
    <link rel="stylesheet" href="liste4.css">
  </head>
  <body>
    <ul class="fruit"> <!-- debut de la liste -->
      <li>Orange</li> <!-- élément de la liste -->
      <li>Fraise</li> <!-- élément de la liste -->
      <li>Poire</li> <!-- élément de la liste -->
    </ul> <!-- fin de la liste -->
  </body>
</html>
```

Décryptons le code Html :

`` permet de déclarer une nouvelle liste non ordonnée.

`` représente les éléments à l'intérieur de la liste.

Le Résultat (sans css)

- Orange
- Fraise
- Poire

Ajoutons un peu de CSS ensemble (pour remplacer la puce d'origine) :

```
liste4.css
```

```
ul.fruit{
  list-style-type: none; /* none, circle, square, disc (par défaut) */
}
ul.fruit li{
  float: left;
  margin: 15px;
  background: url(img/fleche.gif) left center no-repeat;
  padding: 0 0 0 17px;
}
```

Pour que cet exemple fonctionne, il vous faudra une image nommée fleche.gif.

Décryptons les selecteurs du code Css :

```
.fruit{
  propriété : valeur;
}
/* traduction : "va me chercher la classe fruit". */
ul.fruit{
  propriété : valeur;
}
/* traduction : "va me chercher le ul qui possède la classe fruit". */

/* Dans le cas où il y est une autre zone qui utilise la classe fruit, exemple : <div class="fruit"> a b c </div>
La 2e version ul.fruit ira chercher uniquement la classe fruit sur les ul tandis que la 1ère version ira chercher toutes les classes fruits (ul et div confondues). */

/* - Ce n'est pas pareil que : */
ul .fruit{
  propriété : valeur;
}
/* traduction : "va me chercher la classe fruit qui se trouve dans le ul". (il n'y en a pas). */
```

Décryptons les propriétés du code Css :

`list-style-type` permet de changer la puce des éléments de la liste.

`float` permet d'ajouter un effet flottant aux éléments (côte à côte).

`margin` permet d'ajouter de la marge (pour écarter les éléments entre eux).

`background` permet de spécifier un fond. cela permet de remplacer la puce d'origine par une image (dans notre cas il s'agit d'une flèche). Procurez vous une image et ajustez le nom si nécessaire.

`padding` permet d'ajouter de la marge intérieure (dans notre cas : à gauche pour écarter l'image de la flèche avec le texte).

Le Résultat (avec css)

► Orange ► Fraise ► Poire

Nous retrouvons notre flèche sur le résultat !

69

Les listes avec des effets de transitions

A présent, vous connaissez les listes classiques, mais savez-vous que l'on peut modifier leur affichage en profondeur ? En effet avec l'arrivée des dernières propriétés CSS3, nous allons pouvoir générer des affichages totalement différents !

Pour cet exemple, nous allons déclarer une liste permettant de présenter plusieurs offres pour un abonnement d'un service/outil sur le web.

Nous allons prendre une structure type de page web et y déclarer des listes :

```
list5.html

<!Doctype html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Liste avec Css3</title>
    <link rel="stylesheet" href="list5.css">
  </head>
  <body>

    <ul>
      <li>Projets</li>
      <li>Utilisateurs</li>
      <li>Stockage</li>
      <li>Base de données</li>
      <li>Gestion des droits</li>
    </ul>

    <ul>
      <li>Pack 1</li>
      <li>199 € / mois</li>
      <li>illimité</li>
      <li>125</li>
      <li>illimité</li>
      <li>oui</li>
      <li>oui</li>
    </ul>

    <ul>
      <li>Pack 2</li>
      <li>139 € / mois</li>
      <li>50</li>
      <li>90</li>
      <li>illimité</li>
      <li>oui</li>
      <li>non</li>
    </ul>

    <ul>
      <li>Pack 3</li>
      <li>59 € / mois</li>
      <li>10</li>
      <li>20</li>
      <li>2 GO</li>
      <li>oui</li>
      <li>non</li>
    </ul>

  </body>
</html>
```

Le Résultat (provisoire)

- Projets
- Utilisateurs
- Stockage
- Base de données

- Gestion des droits

- Pack 1

- 199 € / mois

- illimité

- 125

- illimité

- oui

- oui

- Pack 2

- 139 € / mois

- 50

- 90

- illimité

- oui

- non

- Pack 3

- 59 € / mois

- 10

- 20

- 2 GO

- oui

- non

Nous allons disposer la première liste dans une zone à gauche et le reste dans une zone sur la droite.

```
<div class="cote-gauche">
  <!-- le 1er ul -->
</div>

<div class="cote-droit">
  <!-- les 3 derniers ul -->
</div>
```

Ce qui nous donne :

liste5.html

```
<!Doctype html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Liste avec Css3</title>
    <link rel="stylesheet" href="liste5.css">
  </head>
  <body>
    <div class="cote-gauche">
      <ul>
        <li>Projets</li>
        <li>Utilisateurs</li>
        <li>Stockage</li>
        <li>Base de données</li>
        <li>Gestion des droits</li>
      </ul>
    </div>

    <div class="cote-droit">
      <ul>
        <li>Pack 1</li>
        <li>199 € / mois</li>
        <li>illimité</li>
        <li>125</li>
        <li>illimité</li>
        <li>oui</li>
        <li>oui</li>
      </ul>

      <ul>
        <li>Pack 2</li>
        <li>139 € / mois</li>
        <li>50</li>
        <li>90</li>
        <li>illimité</li>
        <li>oui</li>
        <li>non</li>
      </ul>

      <ul>
```

```

<li>Pack 3</li>
<li>59 € / mois</li>
<li>10</li>
<li>20</li>
<li>2 GO</li>
<li>oui</li>
<li>non</li>
</ul>
</div>

</body>
</html>

```

Ensute pour la partie CSS, nous allons redéfinir les styles des listes :

```

liste5.css

body{
font-family: verdana;
font-size: 13px;
}
ul{
list-style: none;
float: left;
margin: 0;
padding: 0;
}

```

Décryptons les propriétés du code Css :

body permet de selectionner toute la page.
font-family permet de modifier la police d'écriture.
font-size permet de modifier la taille d'écriture du texte.

ul permet de selectionner toutes les listes.
list-style permet de redéfinir les puces des listes.
float permet de donner un effet flottant (côte à côté).
margin permet de modifier les marges extérieures. Nous les remettons à zéro pour éviter et ainsi casser l'héritage.
padding permet de modifier les marges intérieures. Nous les remettons à zéro pour éviter et ainsi casser l'héritage.

Le Résultat (provisoire)

	Pack 1	Pack 2	Pack 3
Utilisateurs	199 € / mois	139 € / mois	59 € / mois
Stockage	illimité	50	10
Base de données	125	90	20
Gestion des droits	illimité	2 GO	
	oui	oui	oui
	oui	non	non

Ajoutons quelques classes CSS dans notre fichier html afin de pouvoir mieux les selectionner côté CSS et ainsi appliquer des effets de transformation :
Ci dessous, nous ajouterons les classes CSS "transformation" et "liste1".

```

liste5.html

<!Doctype html>
<html>
<head>
<meta charset="utf-8">
<title>Liste avec Css3</title>
<link rel="stylesheet" href="liste5.css">
</head>
<body>

<div class="cote-gauche">
<ul>
<li>Projets</li>
<li>Utilisateurs</li>
<li>Stockage</li>
<li>Base de données</li>
<li>Gestion des droits</li>
</ul>
</div>
<div class="cote-droit">
<div class="transformation">
<ul class="liste1">
<li>Pack 1</li>
<li>199 € / mois</li>
<li>illimité</li>

```

```

<li>125</li>
<li>illimité</li>
<li>oui</li>
<li>oui</li>
</ul>
</div>
<div class="transformation">
<ul>
<li>Pack 2</li>
<li>139 € / mois</li>
<li>50</li>
<li>90</li>
<li>illimité</li>
<li>oui</li>
<li>non</li>
</ul>
</div>
<div class="transformation">
<ul>
<li>Pack 3</li>
<li>59 € / mois</li>
<li>10</li>
<li>20</li>
<li>2 GO</li>
<li>oui</li>
<li>non</li>
</ul>
</div>
</div>

</body>
</html>

```

Et voici le code CSS associé :

```

liste5.css

body{
font-family: verdana;
font-size: 13px;
}
ul{
list-style: none;
float: left;
margin: 0;
padding: 0;
}
.cote-gauche ul li:first-child{
border-top: 1px solid #c0c0c0;
}
.cote-gauche ul li{
border-bottom: 1px solid #c0c0c0;
border-left: 1px solid #c0c0c0;
padding: 10px;
width: 150px;
}

.cote-droit .transformation ul li:last-child{
border-bottom: 1px solid #c0c0c0;
}

.cote-droit .transformation ul li{
border-top: 1px solid #c0c0c0;
border-right: 1px solid #c0c0c0;
padding: 10px;
width: 150px;
text-align: center;
}
.cote-droit .transformation ul li:first-child{
border-left: 1px solid #c0c0c0;
background: #5d5d5d;
color: #fff;
font-weight: bold;
}
.cote-droit .transformation ul:hover{
background: linear-gradient(#d5d5d5, #8a8a8a);
cursor: pointer;
transform:scale(1.3);
transition-duration: 1s;
}
.cote-droit .transformation ul:hover li:first-child{
background: #000;
}
.cote-gauche ul{
margin: 76px 0 0 0;
}
ul.liste1 li{
border-left: 1px solid #c0c0c0;
}
#contenu{
margin: 70px 0 0 50px;
}
```

Décryptons les selecteurs du code Css :

body
permet de selectionner toute la page.

ul
permet de selectionner toutes les listes.

.cote-gauche ul li:first-child
permet de selectionner le premier (first-child) élément (li) de la liste (ul) de la div ayant pour classe (.cote-gauche).

.cote-gauche ul li
permet de selectionner tous les éléments (li) de la liste (ul) de la div ayant pour classe (.cote-gauche).

.cote-droit .transformation ul li:last-child
permet de selectionner le dernier (last-child) élément (li) de la liste (ul) de la div ayant pour classe (.transformation), elle même contenue dans la div ayant pour classe (.cote-droit).

.cote-droit .transformation ul li
permet de selectionner tous les éléments (li) de la liste (ul) de la div ayant pour classe (.cote-droit).

.cote-droit .transformation ul li:first-child
permet de selectionner le premier (first-child) élément (li) de la liste (ul) de la div ayant pour classe (.transformation), elle même contenue dans la div ayant pour classe (.cote-droit).

.cote-droit .transformation ul:hover
permet de selectionner les ul au moment où la souris de l'internaute les survole (hover) à l'intérieur de la div ayant pour classe (.transformation), elle même contenue dans la div ayant pour classe (.cote-droit).

.cote-gauche ul
permet de selectionner toutes les listes ul de la div ayant pour classe (.cote-gauche)

ul.liste1 li
permet de selectionner tous les éléments (li) de la liste (ul) ayant pour classe (liste1)

Décryptons les propriétés du code Css :

font-family permet de modifier la police d'écriture.
font-size permet de modifier la taille d'écriture du texte.

list-style permet de redéfinir les puces des listes.
float permet de donner un effet flottant (côte à côté).
margin permet de modifier les marges extérieures. Nous les remettons à zéro pour éviter et ainsi casser l'héritage.
padding permet de modifier les marges intérieures. Nous les remettons à zéro pour éviter et ainsi casser l'héritage.

border-top permet d'appliquer une bordure en haut.

border-bottom permet d'appliquer une bordure en bas.
border-left permet d'appliquer une bordure à gauche.
border-right permet d'appliquer une bordure à droite.
padding permet de créer de la marge intérieure.
width permet de définir une largeur.

text-align permet de gérer l'alignement du texte.

background permet de définir une couleur de fond.

color permet de définir une couleur de texte.

font-weight permet de définir la font du texte (gras).

background: linear-gradient permet de définir un fond dégradé entre deux couleurs.

cursor permet de définir un curseur de souris différent.

transform permet la transformation de la liste. scale(1.3); permet de l'augmenter de 30 %.

transition-duration permet de fixer la durée de transition.

Et voici le résultat

Le Résultat (final)

	Pack 1	Pack 2	Pack 3
Projets	199 € / mois	139 € / mois	59 € / mois
Utilisateurs	illimité	50	10
Stockage	125	90	20
Base de données	illimité	illimité	2 GO
Gestion des droits	oui	oui	oui
	oui	non	non

Liens et Ancres

70

Les liens de navigation

Ce qui fait la richesse du web, c'est aussi les liens que l'on peut y trouver, on peut se promener de page en page ou de site en site très facilement avec les liens hypertext !

Nous allons créer deux nouveaux fichiers : lien1.html et lien2.css

Ces deux nouveaux fichiers seront placés dans notre dossier /HtmlCss/09-Lien/.

Ce qui donnera : /HtmlCss/09-Lien/lien1.html et /HtmlCss/09-Lien/lien2.html

Voici le contenu de lien1.html :

```
lien1.html

<!Doctype html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Ma Page 1</title>
    <link rel="stylesheet" href="lien1.css">
  </head>
  <body>

    <h1>Voici la page n°1</h1>
    <a href="lien2.html"> cliquez ici pour vous rendre sur la page 2 </a><br><!-- lien vers une page intérieur -->
    <a href="http://www.tf1.fr/news" title="Se rendre sur le site">TF1 actualités</a> le site officiel de TF1 <br> <!-- lien vers une page extérieur et
    title permet d'obtenir une description -->
    <a href="http://www.google.com/" target="_blank">Google</a> le célèbre moteur de recherche<br> <!-- target="_ blank" pour ouverture dans un
```

```
nouvel onglet -->
<a href="cv.doc"> Télécharger mon CV </a><br> <!-- pour le téléchargement de document -->
<a href="cv.pdf"> Voir mon CV </a><br> <!-- pour l'ouverture d'un fichier pdf -->
<a href="http://www.lepointfr"></a><br> <!-- Lien sur une image -->
<a href="" onClick="window.print()">Imprimer la page</a>

</body>
</html>
```

Décryptons le code Html du fichier lien1.html.

La balise `a` permet de créer un lien hypertexte et l'attribut `href` permet de définir la destination du lien.

Comment faire un lien ?

```
<a href="lien2.html"> cliquez ici pour vous rendre sur la page 2 </a>
```

Ce lien permet de vous emmener sur le fichier qui se nomme lien2.html (attention, l'orthographe doit être bonne et les deux fichiers doivent se trouver dans le même dossier).

```
<a href="http://www.tf1.fr/news" title="Se rendre sur le site">TF1 actualités</a> le site officiel de TF1
```

Ce lien permet de vous emmener sur le site de TF1 et vous verrez apparaître une info bulle (title) lors du survol sur le lien avec votre souris.

71

Un lien ouvrant un nouvel onglet

```
<a href="http://www.google.com/" target="_blank">Google</a> le célèbre moteur de recherche
```

Ce lien permet de vous emmener sur le site de Google sans vous faire quitter la page en cours. La page de destination s'ouvrira dans un nouvel onglet (`target=_blank`). Si vous faites pointer des liens vers des sites extérieurs sur votre propre site, l'utilisation de cet attribut est pratique pour ne pas perdre votre trafic (les visiteurs ne perdront pas la navigation de votre site en cliquant dessus).

72

Un lien proposant un téléchargement

```
<a href="cv.doc"> Télécharger mon CV </a>
```

Ce lien permet de proposer le téléchargement d'un fichier (à condition de créer le fichier cv.doc et de l'enregistrer dans le même dossier).

```
<a href="cv.pdf"> Voir mon CV </a>
```

Ce lien ouvre un document PDF directement dans la page web (à condition de créer le fichier cv.pdf et de l'enregistrer dans le même dossier).

```
<a href="cv.pdf" download> Télécharger mon CV </a>
```

L'attribut `download` introduit directement à l'intérieur du lien `a` permet de proposer le téléchargement afin que le fichier soit enregistrer par l'internaute.

73

Un lien sur une image

```
<a href="http://www.lepoint.fr/"></a>
```

?

Ce lien ouvre permet de naviguer ailleurs en cliquant sur une image (et non pas du texte).

74

Un lien pour imprimer la page

```
<a href="" onClick="window.print()">Imprimer la page</a>
```

?

Ce lien permet de proposer l'impression de la page.

75

Exercice : Créer un lien retour

Objectif de l'exercice : essayer de faire un lien retour de la page lien2.html vers la page lien1.html !

Correction :

Voici le contenu de lien2.html :

lien2.html

```
<!Doctype html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Ma Page 1</title>
    <link rel="stylesheet" href="lien2.css">
  </head>
  <body>
    <a href="lien1.html"> cliquez ici pour vous rendre sur la page 1 </a>
  </body>
</html>
```

Le Résultat

[cliquez ici pour vous rendre sur la page 2](#)

[TF1 actualités](#) le site officiel de TF1

[Google](#) le célèbre moteur de recherche

[Télécharger mon CV](#)

[Voir mon CV](#)



[Imprimer la page](#)

En plus des liens permettant la navigation d'une page web vers une autre, nous pouvons également créer des liens internes permettant de nous déplacer à l'intérieur d'une même page lorsque les contenus sont denses.

Les ances sont particulièrement utiles sur les sites web intégrant beaucoup de contenu (comme wikipedia), cela permet de se déplacer rapidement d'une partie à une autre.

Nous allons créer un nouveau fichier :  ancre.html

Ce nouveau fichier sera placé dans notre dossier  /HtmlCss/09-Lien/.

Ce qui donnera : /HtmlCss/09-Lien/ancre.html

Voyons un exemple ensemble :

```
ancre.html

<!Doctype html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Mes Ancre</title>
    <link rel="stylesheet" href="ancre.css">
  </head>
  <body>

    <h1>Sommaire</h1>
    <a href="#sport">Sport</a><br>
    <a href="#politique">Politique</a><br>
    <a href="#culture">Culture</a><br>
    <hr>
    <h2 id="sport">Sport</h2>
    <p>
      Article sur le sport
      <!-- Prévoir de copier/coller 100 ligne de texte ! -->
    </p>
    <hr>
    <h2 id="politique">Politique</h2>
    <p>
      Article sur la Politique
      <!-- Prévoir de copier/coller 100 ligne de texte ! -->
    </p>
    <hr>
    <h2 id="culture">Culture</h2>
    <p>
      Article sur la Culture
      <!-- Prévoir de copier/coller 100 ligne de texte ! -->
    </p>
  </body>
</html>
```

Il est impératif que vous ajoutiez du texte dans ce fichier pour voir la valeur ajoutée des ances, cela est estimable uniquement lorsque la page web possède une scrollbar

Décryptons le code Html du fichier ancre.html.

Comment faire une ancre html ?

Comme vous le savez, la balise `a` permet de créer un lien hypertext et l'attribut `href` permet de définir la destination du lien.

Dans notre cas, nous allons inscrire un dieze # pour préciser que nous ne souhaitons pas sortir de la page actuel et que nous voulons nous rendre sur le point d'ancre nommé "sport".

Voici l'ancre :

```
<a href="#sport">Sport</a>
```

Le point d'ancre se déclare avec un id (comme pour les id css).

Voici le point d'ancre :

```
<h2 id="html_css1.77"><span class="petitnumero">77</span><span class="isocele"></span> <span class="contenu-h2">Sport</span></h2>
```

Dans notre cas, notre point d'ancre se fait sur une balise h2, mais cela aurait pu se faire sur un autre élément à condition d'y ajouter l'id sport.



➤ Attention

pour qu'une ancre fonctionne, il faut donner le même nom à votre ancre qu'à votre point d'ancrage.
Le Html est sensible à la casse, il faudra relier "#politique" à "politique" et non pas "#politique" à "Politique".

Exercice : En bas de page, faites un lien "retour en haut de page" !

Correction :

```
ancre.html
?
<!Doctype html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Mes Ancres</title>
    <link rel="stylesheet" href="ancre.css">
  </head>
  <body>

    <h1 id="sommaire">Sommaire</h1>
    <a href="#sport">Sport</a><br>
    <a href="#politique">Politique</a><br>
    <a href="#culture">Culture</a><br>
    <hr>
    <h2 id="sport">Sport</h2>
    <p>
      Article sur le sport
      <!-- Prévoir de copier/coller 100 ligne de texte ! -->
    </p>
    <hr>
    <h2 id="politique">Politique</h2>
    <p>
      Article sur la Politique
      <!-- Prévoir de copier/coller 100 ligne de texte ! -->
    </p>
    <hr>
    <h2 id="culture">Culture</h2>
    <p>
      Article sur la Culture
      <!-- Prévoir de copier/coller 100 ligne de texte ! -->
    </p>
    <a href="#sommaire">Remonter en haut de page</a>
  </body>
</html>
```

Le Résultat

Sommaire

[Sport](#)
[Politique](#)
[Culture](#)

Sport

Article sur le sport

Politique

Article sur la Politique

Culture

Article sur la Culture

[Remonter en haut de page](#)

Menu de navigation

Nous allons créer un nouveau fichier :  menu1.html

Ce nouveau fichier sera placé dans notre dossier  /HtmlCss/09-Lien/.

Ce qui donnera : /HtmlCss/09-Lien/menu1.html

```
menu1.html

<!Doctype html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Mon Menu</title>
    <link rel="stylesheet" href="menu1.css">
  </head>
  <body>

    <nav>
      <a href="">Accueil</a>
      <a href="">Actualité</a>
      <a href="">Boutique</a>
      <a href="">Panier</a>
      <a href="">Qui sommes nous ?</a>
    </nav>

  </body>
</html>
```

Nous utilisons la balise `<nav>` (dédié à la navigation) avec des liens `<a>` à l'intérieur.

```
menu1.css

nav a{ /* tous les liens a à l'intérieur de la balise nav */
  float: left; /* effet flottant */
  text-decoration: none; /* pas d'héritage avec effet souligné */
  color: #fffff; /* couleur du texte */
  background: #000000; /* couleur de fond */
  height: 36px; /* hauteur */
  line-height: 33px; /* interligne */
  padding: 0 10px; /* marges intérieur */
  font-family: sans-serif; /* police d'écriture */
  font-size: 13px; /* taille du texte */
  transition: .5s; /* temps de transition lors du hover : une demi seconde */
}
nav a:hover{ /* tous liens lors du survol dans la balise nav */
  background: #f43c80; /* couleur de fond */
}
```

Décryptons les selecteurs et les propriétés du code Css :

`nav a` permet de sélectionner les liens `<a>` à l'intérieur de la balise `<nav>`.

`float: left;` permet de donner un effet flottant côté à côté.

`text-decoration: none;` permet de retirer l'héritage de l'effet souligné des liens.

`color: #fffff;` permet de colorer le texte.

`background: #000000;` permet de définir une couleur de fond.

`height: 36px;` permet de donner une hauteur.

`line-height: 33px;` permet de gérer l'interligne.

`padding: 0 10px;` permet de fixer la marge intérieure.

`font-family: sans-serif;` permet de modifier la police d'écriture.

`font-size: 13px;` permet de modifier la taille du texte.

`transition: .5s;` permet de gérer le temps de transition.

`nav a:hover` permet de sélectionner les liens `<a>` à l'intérieur de la balise `<nav>` lorsque la souris survole les liens du menu.

`background: #f43c80;` permet de définir une couleur de fond différente lorsque la souris survole les liens du menu.

Et voici le résultat :



Exercice : Essayer de faire en sorte lors du survol que soit appliqué un fond bleu sur le 1er lien du menu.

Correction :

menu1.css

```
nav a:hover:first-child{ background: #336699; }
```

Nous utilisons le pseudo élément `first-child` mais nous aurions également pu mettre une classe CSS spécifique au lien sur lequel nous souhaitons appliquer un style particulier.

Et après la correction, voici le résultat :

Résultat

```
Accueil Actualité Boutique Panier Qui sommes nous ?
```

Autre exemple de menu de navigation

Voici un autre menu composé de différents liens.

Nous allons créer un nouveau fichier : menu2.html

Ce nouveau fichier sera placé dans notre dossier /HtmlCss/09-Lien/.

Ce qui donnera : /HtmlCss/09-Lien/menu2.html

menu2.html

```
<!Doctype html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Mon Menu</title>
    <link rel="stylesheet" href="menu2.css">
  </head>
  <body>

    <nav>
      <a href="">Accueil</a>
      <a href="">Actualité</a>
      <a href="">Boutique</a>
      <a href="">Panier</a>
      <a href="">Qui sommes nous ?</a>
    </nav>

  </body>
</html>
```

Pour rappel, nous utilisons la balise `<nav>` dédié à la navigation.

Les balises `<a>` permettent de créer des liens.

menu2.css

```
nav a{
  float: left; /* effet flottant */
  text-decoration: none; /* pas d'héritage avec effet souligné */
  color: #fffff; /* couleur du texte */
  background: url(rubrique_off.png); /* image de fond */
  height: 36px; /* hauteur */
  line-height: 33px; /* interligne */
  padding: 0 10px; /* marges intérieur */
  font-family: sans-serif; /* police d'écriture */
  font-size: 13px; /* taille du texte */
}
nav a:hover{
  background: url(rubrique_on.png); /* image de fond */
}
```

Décryptons les selecteurs et les propriétés du code Css :

`nav a` permet de sélectionner les liens `<a>` à l'intérieur de la balise `<nav>`.

`float: left;` permet de donner un effet flottant côté à côté.

`text-decoration: none;` permet de retirer l'héritage de l'effet souligné des liens.

`color: #fffff;` permet de colorer le texte.

`background: url(rubrique_off.png);` permet de définir une image de fond.

`height: 36px;` permet de donner une hauteur.

```

line-height: 33px; permet de gérer l'interligne.
padding: 0 10px; permet de fixer la marge intérieure.
font-family: sans-serif; permet de modifier la police d'écriture.
font-size: 13px; permet de modifier la taille du texte.
transition: .5s; permet de gérer le temps de transition.

nav a:hover permet de sélectionner les liens <a> à l'intérieur de la balise <nav> lorsque la souris survole les liens du menu.
background: url(rubrique_on.png) permet de définir une image de fond différente lorsque la souris survole les liens du menu.

```

Résultat



Autre exemple de menu de navigation (dans un cadre)

Pour cet exemple, nous allons créer un menu dans un cadre, ce genre de menu secondaire peut être susceptible de se trouver à tout endroit sur un site web.

Nous allons créer un nouveau fichier : menu3.html

Ce nouveau fichier sera placé dans notre dossier /HtmlCss/09-Lien/.

Ce qui donnera : /HtmlCss/09-Lien/menu3.html

```

menu3.html

<!Doctype html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Mon Menu</title>
    <link rel="stylesheet" href="menu3.css">
  </head>
  <body>

    <div class="cadre">
      <span class="cadre">Titre du Cadre</span>
      <a href="#">Lien 1</a>
      <a href="#">Lien 2</a>
      <a href="#">Lien 3</a>
    </div>

  </body>
</html>

```

```

menu3.css

.cadre{
background: url(cadre-degrade.png) repeat-x left bottom #FFFFFF; /* image de fond */
border: 1px solid #CACCCD; /* bordure */
border-radius: 10px; /* bordure arrondie */
width: 200px; /* largeur */
}
.cadre span{
border-bottom: 1px solid #CACCCD; /* bordure du bas */
font-size: 15px; /* taille du texte */
margin: 0; /* marges ext */
padding: 5px 0; /* marges int */
text-align: center; /* alignement du texte */
display : block; /* permet de prendre toute la place */
}
.cadre a{
display: block; /* affichage block (pleine largeur)
border-bottom: 1px solid #CACCCD; /* bordure du bas */
color: #02A3D9; /* couleur du texte */
padding: 5px; /* marges int */
text-decoration: none; /* permet de retirer l'héritage de l'effet souligné des liens */
}
.cadre a:hover{
background: #fffff; /* couleur de fond */
}
.cadre a:last-child{
border: 0; /* pas de bordure pour le dernier lien */
}
.cadre a:hover:last-child{
border-radius: 0 0 10px 10px; /* arrondie lors du survol pour éviter débordement lors de l'application du hover */
}
```

Décryptons les selecteurs et les propriétés du code Css :

```
.cadre permet de selectionner la balise<div> contenant la classe cadre.  
background: url(rubrique_off.png); permet de définir une image de fond.  
border: 1px solid #CACCCD; permet de définir une bordure.  
border-radius: 10px; permet d'obtenir une bordure arrondie.  
width: 200px; permet de définir la largeur du cadre.  
.cadre span permet de selectionner la balise<span> à l'intérieur de la classe cadre. Cela représente le sous-titre.  
border-bottom: 1px solid #CACCCD; permet d'obtenir une bordure en bas.  
font-size: 15px; fixe la taille du texte.  
margin: 0; permet de définir les marges extérieures.  
padding: 5px 0; permet de définir les marges intérieures.  
text-align: center; permet de préciser l'alignement du texte.  
display : block; permet de prendre toute la place.  
.cadre a permet de selectionner la balise<a> à l'intérieur de la classe cadre. Cela représente tous les liens.  
display: block; Nous affichons les liens sous forme de bloc afin qu'ils prennent toute la place en largeur.  
border-bottom: 1px solid #CACCCD; permet d'obtenir une bordure en bas.  
color: #02A3D9; nous permet de préciser la couleur du texte  
padding: 5px; permet de définir les marges intérieures.  
text-decoration: none; permet de retirer l'héritage de l'effet souligné des liens.  
.cadre a:hover permet de selectionner la balise<a> à l'intérieur de la classe cadre au moment du survol (hover). Cela représente les liens survolés.  
background: #ffffff; permet de définir une couleur de fond (ainsi, en survolant les liens, la couleur blanche s'appliquera).  
.cadre a:last-child permet de selectionner la dernière balise<a> à l'intérieur de la classe cadre. Cela représente le dernier lien de la liste.  
border: 0; pas de bordure pour le dernier lien.  
.cadre a:hover:last-child permet de selectionner la dernière balise<a> à l'intérieur de la classe cadre au moment du survol (hover). Cela représente le dernier lien survolé dans la liste.  
border-radius: 0 0 10px 10px; permet de définir une bordure arrondie en bas à gauche et à droite au moment du survol sur le dernier lien.
```

Résultat

Titre du Cadre
Lien 1
Lien 2
Lien 3

79

Un menu déroulant horizontal

Comment créer un menu déroulant horizontal avec les langages Html et Css, tout d'abord, est-ce possible ?

Et oui il est possible de créer un menu déroulant (contenant des sous-rubriques) sans l'aide de JavaScript / Jquery.

Pour cela, nous allons créer deux nouveaux fichiers :  menu-horizontale.html et menu-horizontale.css

Ce nouveau fichier sera placé dans notre dossier  /HtmlCss/09-Lien/.

Ce qui donnera : /HtmlCss/09-Lien/menu-horizontale.html

```
menu-horizontale.html  
  
<!Doctype html>  
<html>  
  <head>  
    <meta charset="utf-8">  
    <title>Mon Menu</title>  
    <link rel="stylesheet" href="menu-horizontale.css">  
  </head>
```

```

<body>
  <ul class="niveau1">
    <li><a href="">Accueil</a></li>
    <li class="sousmenu"><a href="">Formation</a>
      <ul class="niveau2">
        <li><a href="">Etudiants</a></li>
        <li><a href="">Emploi du temps</a></li>
        <li><a href="">Recherche de stage</a></li>
        <li><a href="">Documents</a></li>
      </ul>
    </li>
    <li><a href="">Informations</a></li>
    <li class="sousmenu">
      <a href="">Cours</a>
      <ul class="niveau2">
        <li><a href="">Internet</a></li>
        <li><a href="">HTML/CSS</a></li>
        <li><a href="">CMS</a></li>
        <li><a href="">Stats</a></li>
        <li><a href="">Réseaux sociaux</a></li>
        <li><a href="">Référencement</a></li>
        <li><a href="">Marketing</a></li>
        <li><a href="">Cahier des charges</a></li>
        <li><a href="">Photoshop</a></li>
      </ul>
    </li>
    <li><a href="">mon CV</a></li>
  </ul>
</body>
</html>

```

Nous emboîterons les `` des sous menus à l'intérieur d'autres ``.

Vous remarquerez également la présence des classes CSS comme "sousmenu" pour identifier les `` (élément de liste) contenant d'autres sous-listes.

Nous ajoutons également les classes CSS "niveau1" et "niveau2" permettant de cibler les `` (conteneur de listes) de 1er niveau (rubriques principales) et de second niveau (sous rubriques).

Voici le code CSS associé :

menu-horizontale.css

```
nav ul li { float:left; list-style: none; }
```

Décryptons les selecteurs et les propriétés du code Css :

`nav ul li` permet de sélectionner la balise `` à l'intérieur de la balise ``, elle-même à l'intérieur de la balise `<nav>`.

`float: left;` permet de donner un effet flottant côté à côté.

`list-style: none;` permet de retirer les puces de la liste.

Ce qui nous conduit au résultat provisoire suivant :

Résultat (provisoire)

The screenshot shows a horizontal navigation bar with the following structure:

- Accueil
- Formation
- Informations
- Etudiants
- Emploi du temps
- Recherche de stage
- Documents
- Cours
- Internet
- HTML/CSS
- CMS
- Stats
- Réseaux sociaux
- Référencement
- Marketing
- Cahier des charges
- Photoshop
- mon CV

Ensuite, nous allons avoir besoin de cacher les sous rubriques déroulantes, puisque le principe d'un menu déroulant s'est dérouler !

menu-horizontale.css

```
nav ul li { float:left; list-style: none; }
nav ul ul { display:none; width:150px; }
```

Cette fois ci, nous sélectionnons les balises `` de second niveau à l'intérieur des balises `` de premier niveau. Cela revient à dire "va me chercher les UL à l'intérieur des autres UL" ou encore "Selectionne les UL contenus à l'intérieur d'autres UL".

`display: none;` permet de masquer les sous-listes.

`width: 150px;` permet de modifier la largeur de la sous liste (en vue de leur affichage).

Ce qui nous conduit au résultat provisoire suivant :

Résultat (provisoire)

[Accueil](#)[Formation](#)[Informations](#)[Cours](#)[mon CV](#)

menu-hizontale.css

```
nav ul li { float:left; list-style: none; }
nav ul ul {display:none; width:150px;}
nav li {background:#CCCCCC;}
```

Nous allons sélectionner les éléments de listes représentés par la balise `` à l'intérieur de la balise `<nav>`.

Background permet de leur appliquer une couleur de fond (ceci est facultatif).

Résultat (provisoire)

[Accueil](#)[Formation](#)[Informations](#)[Cours](#)[mon CV](#)

menu-hizontale.css

```
nav ul li { float:left; list-style: none; }
nav ul ul {display:none; width:150px;}
nav li {background:#CCCCCC;}
nav li a {text-decoration: none; color: #000; padding: 4px 0 4px; display:block; width:150px;}
```

Nous allons sélectionner les liens (balise `<a>`) à l'intérieur de la balise ``, elle même contenue dans la balise `<nav>`.

`text-decoration: none;` Nous retirons l'effet souligné.

`color: #000;` Nous modifions la couleur du texte en blanc.

`padding: 4px 0 4px;` Nous ajoutons de la marge interieure de 4px en haut, 0 à droite, et 4px à gauche.

`display:block;` Nous en faisons des éléments "blocks".

`width:150px;` Nous fixons la largeur à 150 pixels.

Résultat (provisoire)

Accueil Formation Informations Cours mon CV

menu-hizontale.css

```
nav ul li { float:left; list-style: none; }
nav ul ul {display:none; width:150px;}
nav li {background:#CCCCCC;}
nav li a {text-decoration: none; color: #000; padding: 4px 0 4px; display:block; width:150px;}
nav ul {padding: 0; margin:0px; text-align:center;}
```

Nous ajoutons une ligne permettant de retirer la marge extérieure (margin) et intérieure (padding), cela sera plus pratique en l'absence d'un reset CSS.

Nous alignons le texte au centre (`text-align`).

Il faut à présent faire ressortir les sous rubriques au moment du survol (hover) :

menu-hizontale.css

```
nav ul li { float:left; list-style: none; }
nav ul ul {display:none; width:150px;}
nav li {background:#CCCCCC;}
nav li a {text-decoration: none; color: #000; padding: 4px 0 4px; display:block; width:150px;}
nav ul {padding: 0; margin:0px; text-align:center;}
nav ul.niveau1 li.sousmenu:hover ul.niveau2{display:block;}
```

Nous allons sélectionner les balises `` ayant la classe `.niveau2` à l'intérieur de la balise `` en survol ayant la classe `.sousmenu`, elle même contenue dans la balise `` à l'intérieur de la balise `<nav>`.

En d'autres termes : nous ciblons les `` de niveau 2 lorsque les `` contenu dans les `` de niveau1 sont survolés.

C'est précisément ce qui permettra l'ouverture du menu déroulant !

Résultat (provisoire)

Accueil	Formation	Informations	Cours	mon CV
---------	-----------	--------------	-------	--------

Le code CSS final :

menu-horizontale.css

```
nav ul li { float:left; list-style: none; }
nav ul ul {display:none; width:150px;}
nav li {background:#CCCCCC}
nav li a {text-decoration: none; color: #000; padding: 4px 0 4px; display:block; width:150px;}
nav ul {padding: 0; margin:0px; text-align:center;}
nav ul.niveau1 li.sousmenu:hover ul.niveau2{display:block;}
nav li:hover {background: #E0D;}
nav li.sousmenu:hover {background: #EBB;}
```

Le résultat final :

Résultat (final)

Accueil	Formation	Informations	Cours	mon CV
---------	-----------	--------------	-------	--------

80

Un menu déroulant vertical

Voici une version d'un menu déroulant en vertical et uniquement à l'aide des langages Html et Css :

menu-vertical.html

```
<!Doctype html>
<html>
<head>
  <meta charset="utf-8">
  <title>Mon Menu</title>
  <link rel="stylesheet" href="menu-vertical.css">
</head>
<body>

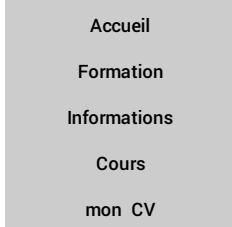
<ul class="niveau1">
  <li><a href="">Accueil</a></li>
  <li class="sousmenu"><a href="">Formation</a>
    <ul class="niveau2">
      <li><a href="">Etudiants</a></li>
      <li><a href="">Emploi du temps</a></li>
      <li><a href="">Recherche de stage</a></li>
      <li><a href="">Documents</a></li>
    </ul>
  </li>
  <li><a href="">Informations</a></li>
  <li class="sousmenu"><a href="">Cours</a>
    <ul class="niveau2">
      <li><a href="">Internet</a></li>
      <li><a href="">HTML/CSS</a></li>
      <li><a href="">CMS</a></li>
      <li><a href="">Stats</a></li>
      <li><a href="">Réseaux sociaux</a></li>
      <li><a href="">Référencement</a></li>
      <li><a href="">Marketing</a></li>
      <li><a href="">Cahier des charges</a></li>
      <li><a href="">Photoshop</a></li>
    </ul>
  </li>
  <li><a href="">mon CV</a></li>
</ul>
```

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

menu-vertical.css

```
nav ul li { float:left; list-style: none; }
nav ul ul {display:none; width:150px;}
nav li {background:#CCCCCC;}
nav li a {text-decoration: none; color: #000; padding: 4px 0 4px; display:block; width:150px;}
nav ul {padding: 0; margin:0px; text-align:center;}
nav ul.niveau1 li.sousmenu:hover ul.niveau2{display:block;}
nav li:hover {background: #EDD;}
nav li.sousmenu:hover {background: #EBB;}
```

Résultat



81

Menu fixé en haut de page

Voici une version d'un menu fixé en haut de page à l'aide de la position fixed en CSS :

menu-fixe.html

```
<!Doctype html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Mon Menu</title>
    <link rel="stylesheet" href="menu-fixe.css">
  </head>
  <body>
    <nav>
      <a href="">Accueil</a>
      <a href="">Actualités</a>
      <a href="">Qui Sommes Nous ?</a>
      <a href="">Contact</a>
    </nav>
    <div id="contenu">
      [Ecrivez 100 lignes de texte]
    </div>
  </body>
</html>
```

Et voici le code CSS qui va avec :

menu-fixe.css

```
nav{
  background: #000; /* fond */
  color: #fff;
  padding: 0.5em 0;
  position: fixed; /* Le menu a une position fixe lors de la descente dans la page */
  text-align: center; /* alignement du texte centré */
  width: 100%; /* largeur 100% de la fenêtre */
  top: 0; /* aligné à 0px du haut */
  left: 0; /* aligné à 0px de la gauche */
}
nav a{
  padding: 0.5em 2em;
  color: #fff;
  text-decoration: none;
  height: 33px;
}
nav a:hover{
  background: #fda500;
}
```

```
#contenu{  
border: 2px solid #336699; /* bordure */  
margin: 40px 0 0 0; /* marges haut droite bas gauche */  
}
```

Résultat

Accueil Actualités Qui Sommes Nous ? Contact

de lettres plus ou moins normale, et en tout cas comparable avec celle du français standard. De nombreuses suites logicielles de mise en page ou éditeurs de sites Web ont fait du Lorem Ipsum leur faux texte par défaut, et une recherche pour 'Lorem Ipsum' vous conduira vers de nombreux sites qui n'en sont encore qu'à leur phase de construction. Plusieurs versions sont apparues avec le temps, parfois par accident, souvent intentionnellement (histoires d'y rajouter de petits clins d'œil, voire des phrases embarrassantes). Contrairement à une opinion répandue, le Lorem Ipsum n'est pas simplement du texte aléatoire. Il trouve ses racines dans une oeuvre de la littérature latine classique datant de 45 av. J.-C., le rendant vieux de 2000 ans. Un professeur du Hampden-Sydney College, en Virginie, s'est intéressé à un des mots latins les plus obscurs, consecetur, extrait d'un passage du Lorem Ipsum, et en étudiant tous les usages de ce mot dans la littérature classique, découvrit la source incontestable du Lorem Ipsum. Il évoquent en fait des sections 1.10.32 et 1.10.33 du "De Finibus Bonorum et Malorum" (Des Sutrénes

Pour faire des tests, il suffit de scroller vers le bas (descendre la barre de défilement vers le bas), vous appercevez alors que le menu reste toujours affiché en haut de page.

Les formulaires

82

Création d'un formulaire

Le langage Html permet de construire un formulaire. Le langage Css permet de mettre en forme un formulaire.

Un langage comme PHP (ou autre) permet de récupérer les saisies d'un formulaire. Le langage SQL permet de sauvegarder les informations dans une base de données.

Le langage JAVASCRIPT permet (par exemple) de contrôler les informations saisies par l'internaute.

Voici un premier formulaire :

```
formulaire1.html  
?<!Doctype html><html><head><meta charset="utf-8"><title>Mon Menu</title><link rel="stylesheet" href="menu-vertical.css"></head><body><form method="post" action="" enctype="multipart/form-data"></form></body></html>
```

Un formulaire se trouve toujours à l'intérieur des balises <form>.

L'attribut `method` permet de préciser la manière dont vont circuler les données, généralement 2 choix s'offrent à nous, GET (dans l'url) et POST (côté serveur).

L'attribut `action` permet de définir la page de destination du formulaire. Si vous souhaitez récupérer les saisies des internautes sur la même page, vous pouvez laisser la valeur vide (comme ci-dessus) sinon il faudra préciser le nom d'une page, par exemple : traitement.php (à condition de créer le fichier correspondant).

L'attribut `enctype` permet de faire véhiculer des fichiers multimédias (fichier pdf, doc word, etc. mais aussi des images et des vidéos). Sans cela, les fichiers multimédias ne passeront pas.

Et maintenant ? Notre formulaire est totalement vide, nous allons déclarer des champs à l'intérieur.

formulaire1.html
?<input type="text" name="nom">

```

<!Doctype html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Mon Menu</title>
    <link rel="stylesheet" href="formulaire1.css">
  </head>
  <body>
    <form method="post" action="" enctype="multipart/form-data">
      <label for="pseudo">Pseudo *</label>
      <input type="text" id="pseudo" name="pseudo" required><br>
    </form>
  </body>
</html>

```

Voici la balise `label` qui nous permet de déclarer une étiquette permettant de préciser le type d'informations dont nous aurons besoin.

C'est la balise `input` comportant l'attribut `type="text"` qui va nous permettre de saisir du texte dans une case.

L'attribut `required` permet de rendre le champ obligatoire.

L'attribut `name` est indispensable pour pouvoir récupérer plus tard la saisie via le langage PHP.

En effet, nous ne verrons pas la différence si nous travaillons uniquement en HTML mais plus tard, au niveau du développement côté PHP il ne sera pas possible de récupérer le texte saisie par l'internaute sans l'attribut `name` et je pense que vous serez d'accord avec moi si je vous dis que l'objectif d'un formulaire c'est bien de récupérer les données postées ! sinon notre formulaire ne servira pas à grand chose :



➤ Attention

Pour qu'une récupération des saisies postées soit possible, il faut toujours prévoir des attributs `name` dans les balises `input` de vos formulaires !

L'attribut `for` de la balise `label` fonctionne avec l'attribut `id` de la balise `input`.

Cette fois-ci l'`id` n'est pas présent pour être sélectionné en CSS mais pour augmenter l'accessibilité (norme accessiweb), en effet une personne mal-voyante pourra cliquer sur le mot "Pseudo" et cela activera le curseur dans la case de saisie `input`.

Par conséquent nous n'oublions pas de relier l'attribut `for` de la balise `label` à l'attribut `id` de la balise `input`.

Voici le résultat actuel :

Résultat

Pseudo *

Pour qu'un formulaire soit conforme, il ne faut pas oublier le bouton `submit` (permettant la validation). sinon comment l'internaute pourrait-il soumettre ses données vers le site web ?

Ajoutons le :

formulaire1.html

```

<!Doctype html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Mon Menu</title>
    <link rel="stylesheet" href="formulaire1.css">
  </head>
  <body>
    <form method="post" action="" enctype="multipart/form-data">
      <label for="pseudo">Pseudo *</label>
      <input type="text" id="pseudo" name="pseudo" required><br>
      <input type="submit" value="valider">
    </form>
  </body>
</html>

```

L'attribut `submit` de la balise `input` permet de valider le formulaire.

L'attribut `value` de la balise `input` permet de préciser le texte à l'intérieur du bouton d'envoi.

De cette manière, on se sent un peu mieux, l'internaute pourra tout à fait communiquer avec nous !

Résultat

Pseudo *

V A L I D

Nous pouvons tout à fait ajouter d'autres champs, exemple :

```
formulaire1.html

<!Doctype html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Mon Menu</title>
    <link rel="stylesheet" href="formulaire1.css">
  </head>
  <body>
    <form method="post" action="" enctype="multipart/form-data">
      <label for="pseudo">Pseudo *</label>
      <input type="text" id="pseudo" name="pseudo" required><br>

      <label for="mdp">Mot de passe</label>
      <input type="password" id="mdp" name="mdp"><br>

      <input type="submit" value="valider">
    </form>
  </body>
</html>
```

L'attribut `type` de la balise `input` permet de préciser le type de case de saisie que l'on souhaite obtenir et en précisant la valeur `password`, le texte saisi par l'internaute sera crypté à l'affichage. Ce n'est pas forcément sécurisé au sens propre du terme, cela permet surtout à l'internaute de saisir un mot de passe sans que quelqu'un placé avec lui derrière l'écran (un ami,...) puisse voir le texte saisi.

Comme vous pouvez le constater, toujours pour des normes accessiweb, la valeur de l'attribut `for` est la même que la valeur de l'attribut `id` afin que l'étiquette et la case de saisie soit reliées (si l'internaute clique sur l'étiquette, cela place le curseur dans la case de saisie).

Et bien entendu l'attribut `name` est TOUJOURS présent, sans cela, aucune récupération serait possible ! Ne l'oubliez pas !

Pour rappel nous évitons de mettre des espaces et des accents dans les valeurs de nos attributs, exemple :

```
<input type="text" name="prénom">
```

préférez l'écriture suivante :

```
<input type="text" name="prenom">
```

```
<input type="text" name="mot de passe">
```

préférez l'écriture suivante :

```
<input type="text" name="mdp">
```

Résultat

Pseudo *

Mot de passe

V A L I D

Ajoutons un autre champ :

```

formulaire1.html

<!Doctype html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Mon Menu</title>
    <link rel="stylesheet" href="formulaire1.css">
  </head>
  <body>
    <form method="post" action="" enctype="multipart/form-data">
      <label for="pseudo">Pseudo *</label>
      <input type="text" id="pseudo" name="pseudo" required><br>

      <label for="mdp">Mot de passe</label>
      <input type="password" id="mdp" name="mdp"><br>

      <label for="prenom">Prenom</label>
      <input type="text" id="prenom" name="prenom" placeholder="Votre prenom"><br>

      <input type="submit" value="valider">
    </form>
  </body>
</html>

```

Cette fois-ci, l'attribut placeholder nous permet d'avoir un texte pré-saisi qui s'efface lorsque l'internaute commencera à écrire à l'intérieur de ce champ.

Cela permet de fournir une indication supplémentaire.

Résultat

Pseudo *

Mot de passe

Prenom

V A L I D

Depuis l'arrivée d'HTML5, les champs de formulaires se sont améliorés, exemple :

```

formulaire1.html

<!Doctype html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Mon Menu</title>
    <link rel="stylesheet" href="formulaire1.css">
  </head>
  <body>
    <form method="post" action="" enctype="multipart/form-data">
      <label for="pseudo">Pseudo *</label>
      <input type="text" id="pseudo" name="pseudo" required><br>

      <label for="mdp">Mot de passe</label>
      <input type="password" id="mdp" name="mdp"><br>

      <label for="prenom">Prenom</label>
      <input type="text" id="prenom" name="prenom" placeholder="Votre prenom"><br>

      <label for="email">Email</label>
      <input type="email" id="email" name="email"><br>

      <input type="submit" value="valider">
    </form>
  </body>
</html>

```

Le type email est présent pour contrôler que le format de l'email soit conforme (avec un arrobase @ et un point .).

Cela ne vérifie pas si l'adresse email existe mais déjà ce premier niveau de vérification permet d'éviter une erreur involontaire de l'internaute.

Faites le test ! Essayez de valider une chaîne de caractères quelconque dans ce champ du style : "123". Vous verrez une infobulle venir vous prévenir que le format de l'adresse email n'est pas correct.

Résultat**Pseudo ***

Mot de passe

Prénom

Email

V A L I D

Ajoutons à présent un champ upload pour joindre une image (avatar de profil) ou bien encore un CV dans un formulaire de recrutement ou une vidéo dans un réseau social...

formulaire1.html

```
<!Doctype html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Mon Menu</title>
    <link rel="stylesheet" href="formulaire1.css">
  </head>
  <body>
    <form method="post" action="" enctype="multipart/form-data">
      <label for="pseudo">Pseudo *</label>
      <input type="text" id="pseudo" name="pseudo" required><br>

      <label for="mdp">Mot de passe</label>
      <input type="password" id="mdp" name="mdp"><br>

      <label for="prenom">Prénom</label>
      <input type="text" id="prenom" name="prenom" placeholder="Votre prénom"><br>

      <label for="email">Email</label>
      <input type="email" id="email" name="email"><br>

      <label for="avatar">Avatar</label>
      <input type="file" id="avatar" name="avatar" multiple><br>

      <input type="submit" value="valider">
    </form>
  </body>
</html>
```

Le type File permet de faire apparaître un bouton d'upload parcourir.

Quant à la valeur multiple, cela permet d'accepter l'upload de plusieurs fichiers en une seule fois.

Résultat**Pseudo ***

Mot de passe

Prénom

Email

Avatar
 Choose File | No file selected
V A L I D

Parfois, ce n'est pas à l'internaute de saisir du texte mais on lui demande de faire un choix parmi plusieurs options, exemple :

formulaire1.html

```
<!Doctype html>
<html>
  <head>
```

```

<meta charset="utf-8">
<title>Mon Menu</title>
<link rel="stylesheet" href="formulaire1.css">
</head>
<body>
<form method="post" action="" enctype="multipart/form-data">
<label for="pseudo">Pseudo *</label>
<input type="text" id="pseudo" name="pseudo" required><br>

<label for="mdp">Mot de passe</label>
<input type="password" id="mdp" name="mdp"><br>

<label for="prenom">Prenom</label>
<input type="text" id="prenom" name="prenom" placeholder="Votre prenom"><br>

<label for="email">Email</label>
<input type="email" id="email" name="email"><br>

<label for="avatar">Avatar</label>
<input type="file" id="avatar" name="avatar" multiple><br>

<label for="Pays">Pays</label>
<select id="Pays" name="pays">
<optgroup label="Europe">
<option value="fr">France</option>
<option value="it">Italie</option>
<option value="es" selected>Espagne</option>
</optgroup>
</select><br>
<input type="submit" value="valider">
</form>
</body>
</html>

```

La balise select permet effectivement d'obtenir un déroulant proposant plusieurs options.

Optgroup n'est pas obligatoire, cela permet de classer et "catégoriser" des options s'il y en a beaucoup.

A l'intérieur d'une option, l'attribut selected permet de garder une option sélectionnée par défaut.

La balise select contient des options. L'internaute doit faire son choix parmi les options présentes (que nous avons définie à l'avance).

A l'intérieur de la balise <select> nous pourrions ajouter l'attribut multiple pour donner la possibilité à l'internaute de choisir plusieurs options (en maintenant la touche **ctrl** de son clavier appuyé).

L'attribut name est toujours présent pour permettre au développeur (côté php) de récupérer l'option qui aura été choisie par l'internaute.

La value présente dans la balise <option> permettra au développeur (côté php) de récupérer une information de notre choix plutôt que le texte contenu dans les options.

C'est à dire qu'avec le code suivant :

```
<option>France</option>
```

Nous récupérerons France dans le traitement PHP.

Alors qu'avec le code suivant :

```
<option value="fr">France</option>
```

Nous récupérerons la valeur fr dans le traitement PHP.

Résultat

Pseudo *

Mot de passe

Prenom

Email

Avatar

 Choose File No file selected

Pays

Espagne

V A L I D

Ajoutons d'autres choix sous forme de cases à cocher :

formulaire1.html

```
<!Doctype html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Mon Menu</title>
    <link rel="stylesheet" href="formulaire1.css">
  </head>
  <body>
    <form method="post" action="" enctype="multipart/form-data">
      <label for="pseudo">Pseudo *</label>
      <input type="text" id="pseudo" name="pseudo" required><br>

      <label for="mdp">Mot de passe</label>
      <input type="password" id="mdp" name="mdp"><br>

      <label for="prenom">Prenom</label>
      <input type="text" id="prenom" name="prenom" placeholder="Votre prenom"><br>

      <label for="email">Email</label>
      <input type="email" id="email" name="email"><br>

      <label for="avatar">Avatar</label>
      <input type="file" id="avatar" name="avatar" multiple><br>

      <label for="Pays">Pays</label>
      <select id="Pays" name="pays">
        <optgroup label="Europe">
          <option value="fr">France</option>
          <option value="it">Italie</option>
          <option value="es" selected>Espagne</option>
        </optgroup>
      </select><br>

      <label for="choix">Quel sont vos fruits préférés ?</label>
      <p><input type="checkbox" name="choix" checked> Orange
      <input type="checkbox" name="choix"> Fraise
      <input type="checkbox" name="choix"> Poire</p>

      <input type="submit" value="valider">
    </form>
  </body>
</html>
```

L'attribut `type` de la balise `input` permet de préciser le type de case de saisie que l'on souhaite obtenir et en précisant la valeur `checkbox`, cela permet d'obtenir une case à cocher.

L'attribut `name` permet de rendre le choix de l'internaute récupérable (en PHP).

L'attribut `checked` permet de cocher une case par défaut.

Résultat

Pseudo *

Mot de passe

Prenom

Email

Avatar

Choose File | No file selected

Pays

Espagne

Quel sont vos fruits préférés ?

Orange Fraise Poire

V A L I D

Dans l'exemple précédent, nous venons de voir que l'internaute pouvait cocher plusieurs cases et ainsi effectuer plusieurs choix.

Dans cet exemple, nous allons conserver les cases à cocher avec la possibilité d'effectuer un choix unique :

```
formulaire1.html
?
<!Doctype html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Mon Menu</title>
    <link rel="stylesheet" href="formulaire1.css">
  </head>
  <body>
    <form method="post" action="" enctype="multipart/form-data">
      <label for="pseudo">Pseudo *</label>
      <input type="text" id="pseudo" name="pseudo" required><br>

      <label for="mdp">Mot de passe</label>
      <input type="password" id="mdp" name="mdp"><br>

      <label for="prenom">Prenom</label>
      <input type="text" id="prenom" name="prenom" placeholder="Votre prenom"><br>

      <label for="email">Email</label>
      <input type="email" id="email" name="email"><br>

      <label for="avatar">Avatar</label>
      <input type="file" id="avatar" name="avatar" multiple><br>

      <label for="Pays">Pays</label>
      <select id="Pays" name="pays">
        <optgroup label="Europe">
          <option value="fr">France</option>
          <option value="it">Italie</option>
          <option value="es" selected>Espagne</option>
        </optgroup>
      </select><br>

      <label for="choix">Quel sont vos fruits préférés ?</label>
      <p><input type="checkbox" name="choix"> Orange
      <input type="checkbox" name="choix"> Fraise
      <input type="checkbox" name="choix"> Poire</p>

      <label for="civilite">Civilite</label>
      <p><input type="radio" name="civilite" value="m" checked>Homme
      <input type="radio" name="civilite" value="f">Femme</p>

      <input type="submit" value="valider">
    </form>
  </body>
</html>
```

En précisant la valeur `radio` dans l'attribut `type` de la balise `input` cela permet d'obtenir une case à cocher unique.

Ainsi l'internaute ne pourra effectuer qu'un seul choix.

L'attribut `name` permet de rendre le choix de l'internaute récupérable (en PHP).

L'attribut `value` permet de récupérer (en PHP) soit M soit F.

L'attribut `checked` permet de cocher une case par défaut.

Résultat

Pseudo *

Mot de passe

Prenom

Email

Avatar

Choose File | No file selected

Pays

Quel sont vos fruits préférés ?

Orange Fraise Poire

Civilite

Homme Femme

V A L I D

Depuis l'arrivée de Html5, nous pouvons également prévoir des suggestions automatiques venant en compléition de la saisie d'un internaute, exemple :

```
formulaire1.html

<!Doctype html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Mon Menu</title>
    <link rel="stylesheet" href="formulaire1.css">
  </head>
  <body>
    <form method="post" action="" enctype="multipart/form-data">
      <label for="pseudo">Pseudo *</label>
      <input type="text" id="pseudo" name="pseudo" required><br>

      <label for="mdp">Mot de passe</label>
      <input type="password" id="mdp" name="mdp"><br>

      <label for="prenom">Prenom</label>
      <input type="text" id="prenom" name="prenom" placeholder="Votre prenom"><br>

      <label for="email">Email</label>
      <input type="email" id="email" name="email"><br>

      <label for="avatar">Avatar</label>
      <input type="file" id="avatar" name="avatar" multiple><br>

      <label for="Pays">Pays</label>
      <select id="Pays" name="pays">
        <optgroup label="Europe">
          <option value="fr">France</option>
          <option value="it">Italie</option>
          <option value="es" selected>Espagne</option>
        </optgroup>
      </select><br>

      <label for="choix">Quel sont vos fruits préférés ?</label>
      <p><input type="checkbox" name="choix"> Orange
      <input type="checkbox" name="choix"> Fraise
      <input type="checkbox" name="choix"> Poire</p>

      <label for="civilite">Civilite</label>
      <p><input type="radio" name="civilite" value="m" checked>Homme
      <input type="radio" name="civilite" value="f">Femme</p>

      <label for="ville">Ville</label>
      <input type="text" id="ville" name="ville" list="france">
      <datalist id="france">
        <option>Paris</option>
        <option>Marseille</option>
        <option>Lyon</option>
      </datalist><br>

      <input type="submit" value="valider">
    </form>
  </body>
</html>
```

La case de saisie `input` permet de saisir du texte.

La balise `datalist` permet de déclarer une liste de suggestions.

Les balises `option` permettent d'annoncer les suggestions.

Faites le test ! Lorsque nous saisirons le P, Paris sera alors proposé automatiquement !

Pour cela, vous aurez remarqué qu'il faut relier la case de saisie `input` avec la balise `datalist` par l'intmédiaire des attributs `list` et `id`.

Résultat

Pseudo *

Mot de passe

Prenom**Email****Avatar** No file selected**Pays** **Quel sont vos fruits préférés ?** Orange Fraise Poire**Civilite** Homme Femme**Ville**

Grâce aux expressions régulières en HTML 5, nous pouvons également demander à ce que l'internaute saisisse uniquement des chiffres :

formulaire1.html

```
<!Doctype html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Mon Menu</title>
    <link rel="stylesheet" href="formulaire1.css">
  </head>
  <body>
    <form method="post" action="" enctype="multipart/form-data">
      <label for="pseudo">Pseudo *</label>
      <input type="text" id="pseudo" name="pseudo" required><br>

      <label for="mdp">Mot de passe</label>
      <input type="password" id="mdp" name="mdp"><br>

      <label for="prenom">Prenom</label>
      <input type="text" id="prenom" name="prenom" placeholder="Votre prenom"><br>

      <label for="email">Email</label>
      <input type="email" id="email" name="email"><br>

      <label for="avatar">Avatar</label>
      <input type="file" id="avatar" name="avatar" multiple><br>

      <label for="Pays">Pays</label>
      <select id="Pays" name="pays">
        <optgroup label="Europe">
          <option value="fr">France</option>
          <option value="it">Italie</option>
          <option value="es" selected>Espagne</option>
        </optgroup>
      </select><br>

      <label for="choix">Quel sont vos fruits préférés ?</label>
      <p><input type="checkbox" name="choix"> Orange
      <input type="checkbox" name="choix"> Fraise
      <input type="checkbox" name="choix"> Poire</p>

      <label for="civilite">Civilite</label>
      <p><input type="radio" name="civilite" value="m" checked>Homme
      <input type="radio" name="civilite" value="f">Femme</p>

      <label for="ville">Villes</label>
      <input type="text" id="ville" name="ville" list="france">
      <datalist id="france">
        <option>Paris</option>
        <option>Marseille</option>
        <option>Lyon</option>
      </datalist><br>

      <label for="cp">Code Postale</label>
      <input type="text" id="cp" name="cp" maxlength="5" pattern="[0-9]{5}" title="5 chiffres requis entre 0 et 9"><br>

      <input type="submit" value="valider">
    </form>
  </body>
</html>
```

L'attribut `maxlength` permet de préciser que nous bloquerons la saisie au bout de 5 caractères.

L'attribut `pattern` permet d'accepter uniquement les chiffres entre 0 et 9, au nombre de 5.

L'attribut `title` permet de préciser le message d'erreur si la saisie de l'internaute ne respecte pas les règles imposées par le site web.

Faites le test !

Résultat

Pseudo *

Mot de passe

Prenom

Email

Avatar

No file selected

Pays

Quel sont vos fruits préférés ?

Orange Fraise Poire

Civilite

Homme Femme

Ville

Code Postale

Ajoutons une zone de saisie :

```
formulaire1.html

<!Doctype html>
<html>
<head>
<meta charset="utf-8">
<title>Mon Menu</title>
<link rel="stylesheet" href="formulaire1.css">
</head>
<body>
<form method="post" action="" enctype="multipart/form-data">
<label for="pseudo">Pseudo *</label>
<input type="text" id="pseudo" name="pseudo" required><br>

<label for="mdp">Mot de passe</label>
<input type="password" id="mdp" name="mdp"><br>

<label for="prenom">Prenom</label>
<input type="text" id="prenom" name="prenom" placeholder="Votre prenom"><br>

<label for="email">Email</label>
<input type="email" id="email" name="email"><br>

<label for="avatar">Avatar</label>
<input type="file" id="avatar" name="avatar" multiple><br>

<label for="Pays">Pays</label>
<select id="Pays" name="pays">
<optgroup label="Europe">
<option value="fr">France</option>
<option value="it">Italie</option>
<option value="es" selected>Espagne</option>
</optgroup>
</select><br>

<label for="choix">Quel sont vos fruits préférés ?</label>
<p><input type="checkbox" name="choix"> Orange
<input type="checkbox" name="choix"> Fraise
```

```

<input type="checkbox" name="choix"> Poire</p>
<label for="civilite">Civilite</label>
<p><input type="radio" name="civilite" value="m" checked>Homme
<input type="radio" name="civilite" value="f">Femme</p>

<label for="ville">Ville</label>
<input type="text" id="ville" name="ville" list="france">
<datalist id="france">
  <option>Paris</option>
  <option>Marseille</option>
  <option>Lyon</option>
</datalist><br>

<label for="cp">Code Postale</label>
<input type="text" id="cp" name="cp" maxlength="5" pattern="[0-9]{5}" title="5 chiffres requis entre 0 et 9"><br>

<label for="adresse">Adresse</label>
<textarea id="adresse" name="adresse" rows="5" cols="25"></textarea><br>

<input type="submit" value="valider">
</form>
</body>
</html>

```

La balise `textarea` permet de définir une zone de saisie.

Input est une case de saisie, Textarea est une zone de saisie.

Comme d'habitude l'attribut `name` permet de préciser un nom qui pourra être ré-utiliser côté PHP pour récupérer la saisie de l'internaute.

Les attributs `rows` et `cols` permettent de préciser la hauteur et la largeur de la zone de saisie mais cela peut également être géré avec une feuille de style CSS.

Résultat

Pseudo *

Mot de passe

Prenom

Email

Avatar

 No file selected

Pays

Quel sont vos fruits préférés ?

Orange Fraise Poire

Civilite

Homme Femme

Ville

Code Postale

Adresse

V A L I D

Parmi les dernières innovations en matière de formulaires, nous pouvons ajouter d'autres champs (mais attention l'interprétation des navigateurs reste à vérifier).

formulaire1.html

```
<!Doctype html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Mon Menu</title>
    <link rel="stylesheet" href="formulaire1.css">
  </head>
  <body>
    <form method="post" action="" enctype="multipart/form-data">
      <label for="pseudo">Pseudo *</label>
      <input type="text" id="pseudo" name="pseudo" required><br>

      <label for="mdp">Mot de passe</label>
      <input type="password" id="mdp" name="mdp"><br>

      <label for="prenom">Prenom</label>
      <input type="text" id="prenom" name="prenom" placeholder="Votre prenom"><br>

      <label for="email">Email</label>
      <input type="email" id="email" name="email"><br>

      <label for="avatar">Avatar</label>
      <input type="file" id="avatar" name="avatar" multiple><br>

      <label for="Pays">Pays</label>
      <select id="Pays" name="pays">
        <optgroup label="Europe">
          <option value="fr">France</option>
          <option value="it">Italie</option>
          <option value="es" selected>Espagne</option>
        </optgroup>
      </select><br>

      <label for="choix">Quel sont vos fruits préférés ?</label>
      <p><input type="checkbox" name="choix"> Orange
      <input type="checkbox" name="choix"> Fraise
      <input type="checkbox" name="choix"> Poire</p>

      <label for="civilite">Civilite</label>
      <p><input type="radio" name="civilite" value="m" checked>Homme
      <input type="radio" name="civilite" value="f">Femme</p>

      <label for="ville">Ville</label>
      <input type="text" id="ville" name="ville" list="france">
      <datalist id="france">
        <option>Paris</option>
        <option>Marseille</option>
        <option>Lyon</option>
      </datalist><br>

      <label for="cp">Code Postale</label>
      <input type="text" id="cp" name="cp" maxlength="5" pattern="[0-9]{5}" title="5 chiffres requis entre 0 et 9"><br>

      <label for="adresse">Adresse</label>
      <textarea id="adresse" name="adresse" rows="5" cols="25"></textarea><br>

      <label for="datedenaissance">Date de Naissance</label>
      <input type="date" name="datedenaissance" id="datedenaissance"><br>

      <label for="jauge">Jauge</label>
      <input type="range" min="25" max="50" value="26" id="jauge"><br>

      <label for="telephone">Telephone</label>
      <input type="tel" pattern="^([0-9]{10})$"><br>

      <label for="recherche">recherche</label>
      <input type="search" pattern="^([0-9]{10})$"><br>

      <label for="url">url</label>
      <input type="url" pattern="^([0-9]{10})$"><br>

      <label for="couleur">Couleur</label>
      <input type="color"><br>

      <input type="submit" value="valider">
    </form>
  </body>
</html>
```

Le type `date` permet d'afficher un calendrier pour sélectionner une date.

Le type `range` permet d'afficher une jauge et de choisir un chiffre entre deux nombres donnés (pratique pour des filtres).

Le type `tel` permet à un smartphone de proposer le clavier numérique.

Le type `search` permet de construire un champ de recherche pour conserver en mémoire les dernières recherches effectuées.

Le type `color` permet de saisir une adresse url.

Le type `color` permet de faire sortir une palette de couleurs.

Résultat

Pseudo *

Mot de passe

Prénom

Email

Avatar
 Choose File No file selected

Pays
 

Quel sont vos fruits préférés ?

Orange Fraise Poire

Civilité

Homme Femme

Ville

Code Postale

Adresse

Date de Naissance

Jauge


Téléphone

recherche

url

Couleur

V A L I D

Voilà, tous les champs existants n'ont pas été présentés, mais à priori vous devriez quand même trouver votre bonheur dans ces différentes possibilités.

Un peu de CSS pour améliorer la présentation de nos formulaires :

```
formulaire1.css
?
// ... votre code ...
```

Balise FORM					
Balise	Attribut	Valeur	code (exemple)	Description	Affichage
<form>	method	POST GET	<form method="post" action="" enctype=""> </form>	permet de préciser de quelle manière vont circuler les données C'est important pour la récupération et le traitement PHP.	-
	action	traitement.php	<form method="post" action="" enctype=""> </form>	permet de préciser sur quelle page nous allons arriver après l'envoi du formulaire.	-
	enctype	multipart/form-data	<form method="post" action="" enctype="multipart/form-data"> </form>	permet de préciser le type mime utilisé.	-

Balise LABEL					
Balise	Attribut	Valeur	code (exemple)	Description	Affichage
<label>	for	exemple	<label for="exemple">Pseudo :</label>	permet de préciser un nom identique à l'id d'une case de saisie <input> afin de les relier entre eux.	Pseudo :

Balise INPUT					
<input>	name	exemple	<input type="text" name="nom1">	permet de créer une case de saisie ayant pour nom : nom1.	<input type="text"/>
<input>	id	exemple	<input type="text" name="champ" id="exemple">	permet de créer une case de saisie ayant pour id : exemple. Cela permet de faire la liaison avec l'attribut for d'un label.	<input type="text"/>
<input>	required	-	<input type="text" name="champ" required>	permet de rendre la saisie sur le champ obligatoire.	<input type="text"/>
<input>	maxlength	10	<input type="text" name="champ" maxlength="10">	permet de fixer la saisie à 10 caractères maximum.	<input type="text"/>
<input>	readonly	-	<input type="text" name="champ" readonly>	permet de rendre le champ lisible seulement, sans modification possible.	<input type="text"/>
<input>	disabled	-	<input type="text" name="champ" disabled>	permet de désactiver le champ.	<input type="text"/>
<input>	list	exemple	<input type="text" name="ville" list="France">	permet de faire la liaison avec une balise datalist ayant pour id : France.	<input type="text"/>
<input>	pattern	[0-9]{5}	<input type="text" name="cp" pattern="[0-9]{5}>	expression régulière : 5 chiffres requis entre 0 et 9.	<input type="text"/>
<input>	title	texte	<input type="text" name="cp" pattern="[0-9]{5}" title="5 chiffres requis entre 0 et 9">	le title permet de personnaliser le message d'erreur dans une infobulle.	<input type="text"/>
<input>	value	texte	<input type="submit" value="mon texte">	l'attribut value permet de saisir du texte dans le champ (ou dans un bouton submit).	<input type="text"/>
			<input type="text" placeholder="mon placeholder">	placeholder permet de donner des indications au utilisateur.	<input type="text"/>

<input>	placeholder	texte	<pre><input type="text" placeholder="mon texte"></pre>	l'attribut placeholder permet de pré-saisir du texte.	<input type="text"/>
<input>	type	text	<pre><input type="text"></pre>	permet de créer une case de saisie.	<input type="text"/>
<input>	type	password	<pre><input type="password" name="nom"></pre>	permet de créer une case de saisie avec saisie des caractères cryptés.	<input type="password"/>
<input>	type	email	<pre><input type="email" name="nom"></pre>	permet de créer une case de saisie acceptant uniquement la saisie d'une adresse email.	<input type="email"/>
<input>	type	file	<pre><input type="file" name="nom"></pre>	permet de créer une case de saisie pour l'upload de fichiers.	<input type="file"/> Choose File No file selected
<input>	type	checkbox	<pre><p><input type="checkbox" name="choix" checked> Orange</p></pre>	permet de créer une case à cocher.	<input checked="" type="checkbox"/>
<input>	type	radio	<pre><p><input type="radio" name="civilite" value="m">Homme</p></pre>	permet de créer une case à cocher unique.	<input type="radio"/>
<input>	type	date	<pre><input type="date" name="dateDeNaissance"></pre>	permet d'ouvrir un calendrier pour choisir une date.	<input type="date"/>
<input>	type	range	<pre><input type="range" min="1" max="100" name="nombre"></pre>	permet d'afficher une jauge.	
<input>	type	tel	<pre><input type="tel" name="nombre"></pre>	permet d'afficher un clavier numérique sur les smartphones.	<input type="tel"/>
<input>	type	color	<pre><input type="color" name="couleur"></pre>	permet d'afficher une palette de couleur.	<input type="color"/>
<input>	type	hidden	<pre><input type="hidden" id="cache" name="cache"></pre>	permet d'afficher un champ caché.	
<input>	type	submit	<pre><input type="submit"></pre>	permet de valider un formulaire.	<input type="submit" value="S U B M"/>
<input>	type	reset	<pre><input type="reset"></pre>	permet de réinitialiser le formulaire.	<input type="button" value="Reset"/>

Balise DATALIST						
Balise	Attribut	Valeur	code (exemple)	Description	Affichage	
<datalist>	id	exemple	<pre><input type="text" id="ville" name="ville" list="france"> <datalist id="france"> <option>Paris</option> <option>Marseille</option> <option>Lyon</option> </datalist>
</pre>	permet de proposer une liste de suggestions (l'id de datalist fonctionne avec l'attribut list de la balise input).	<input type="text"/>	

Balise SELECT						
Balise	Attribut	Valeur	code (exemple)	Description	Affichage	
<select>	name	exemple	<pre><select name="exemple"> </select></pre>	SELECT permet d'afficher une liste déroulante	<input type="button" value="▼"/>	
			<pre><select name="pays"> <option name="pays"></pre>			

<select>	option	exemple	<pre>value="fr">France</option> <option name="pays" value="it">Italie</option> <option name="pays" value="es" selected>Espagne</option> </select></pre>	La balise OPTION permet de proposer plusieurs choix à l'intérieur d'un SELECT	<input style="width: 100px; height: 25px; border: 1px solid #ccc; padding: 2px 10px; border-radius: 5px; font-size: 10px;" type="button" value="France"/>
<select>	optgroup	exemple	<pre><select name="pays"> <optgroup label="Europe"> <option name="pays" value="fr">France</option> <option name="pays" value="it">Italie</option> <option name="pays" value="es">Espagne</option> </optgroup> </select></pre>	La balise OPTGROUP permet de classer les options avec l'aide d'une indication écrite	<input style="width: 100px; height: 25px; border: 1px solid #ccc; padding: 2px 10px; border-radius: 5px; font-size: 10px;" type="button" value="France"/>
<select>	multiple	-	<pre><select name="pays" multiple> <optgroup label="Europe"> <option name="pays" value="fr">France</option> <option name="pays" value="it">Italie</option> <option name="pays" value="es">Espagne</option> </optgroup> </select></pre>	L'attribut MULTIPLE permet d'effectuer plusieurs choix	
<select>	selected	-	<pre><select name="pays"> <optgroup label="Europe"> <option name="pays" value="fr">France</option> <option name="pays" value="it">Italie</option> <option name="pays" value="es" selected>Espagne</option> </optgroup> </select></pre>	L'attribut SELECTED permet de sélectionner une option par défaut.	<input style="width: 100px; height: 25px; border: 1px solid #ccc; padding: 2px 10px; border-radius: 5px; font-size: 10px;" type="button" value="Espagne"/>

Balise TEXTAREA					
Balise	Attribut	Valeur	code (exemple)	Description	Affichage
<textarea>	name	exemple	<pre><textarea name="exemple"> </textarea></pre>	La balise textarea permet d'afficher une zone de saisie	<input style="width: 100%; height: 100px; border: 1px solid #ccc; padding: 2px; border-radius: 5px; font-size: 10px;" type="text"/>
<textarea>	rows	10	<pre><textarea name="exemple" rows="10"> </textarea></pre>	L'attribut rows de la balise textarea permet de régler la hauteur	<input style="width: 100%; height: 100px; border: 1px solid #ccc; padding: 2px; border-radius: 5px; font-size: 10px;" type="text"/>
<textarea>	cols	10	<pre><textarea name="exemple" cols="10"> </textarea></pre>	L'attribut cols de la balise textarea permet de régler la largeur	<input style="width: 100px; height: 100px; border: 1px solid #ccc; padding: 2px; border-radius: 5px; font-size: 10px;" type="text"/>

Données Multimédia

Imaginez que vous ayez 2 pages : page A et page B

Avec la balise iframe il est possible d'incorporer la page B dans le contenu de la page A.

Généralement, nous évitons de nous servir de la balise iframe de cette manière, nous l'utilisons plutôt pour intégrer une ressource type : GoogleMap, Animation, etc.

Exemple :

```
pageB.html
?>
<!Doctype html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Mon Iframe</title>
    <link rel="stylesheet" href="iframe.css">
  </head>
  <body>
    <p>Voici le contenu de ma page B</p>
  </body>
</html>
```

```
pageA.html
?>
<!Doctype html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Mon Iframe</title>
    <link rel="stylesheet" href="iframe.css">
  </head>
  <body>
    <p>Bonjour et bienvenue sur la page A</p>
    <iframe src="pageB.html" width="" height=""></iframe>
  </body>
</html>
```

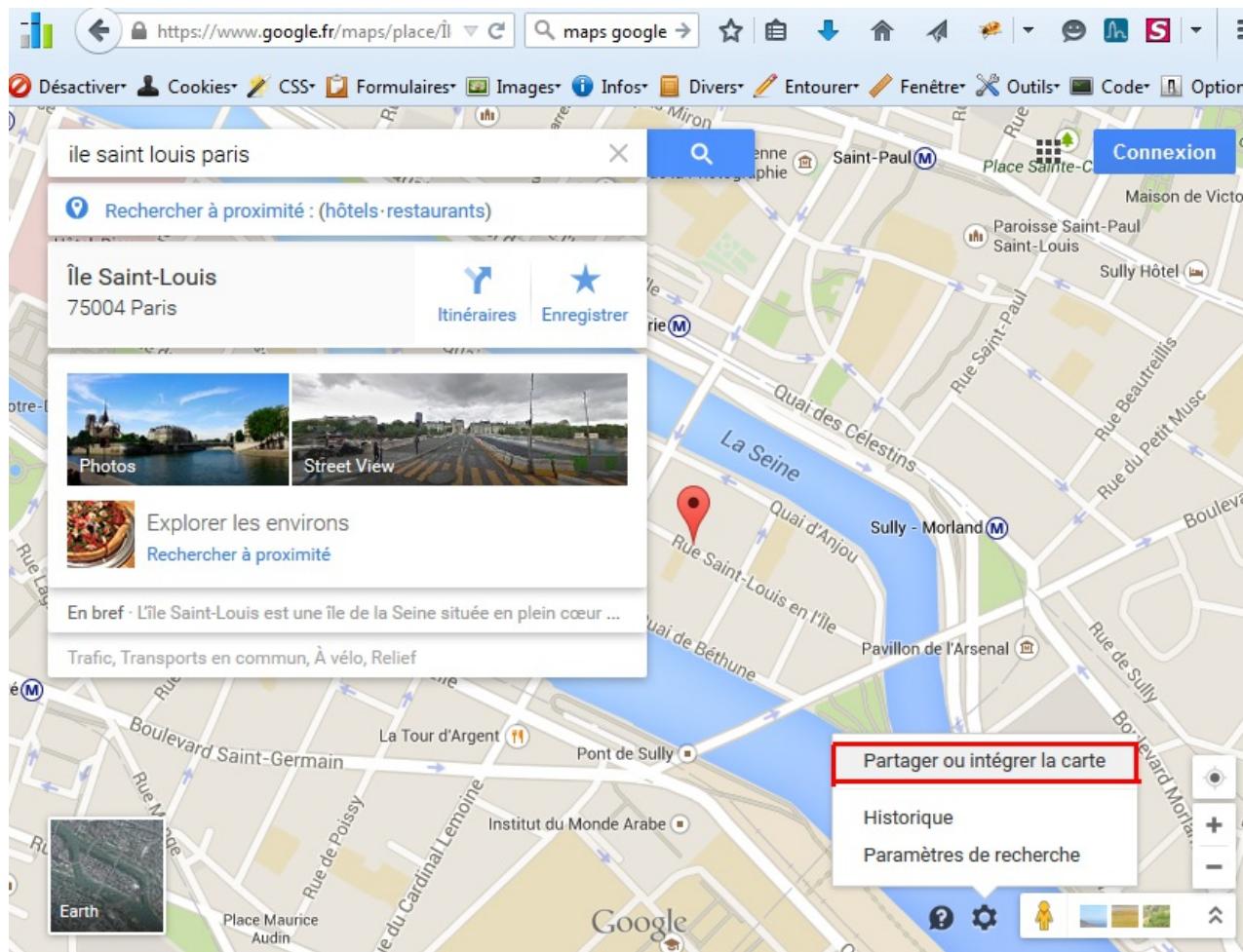
Résultat

Bonjour et bienvenue sur la page A

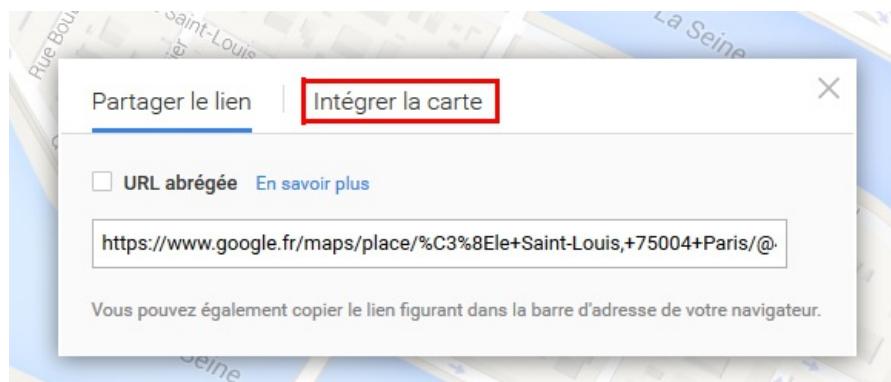
Voici le contenu de ma page B

Pour cela, il faut vous rendre à l'adresse suivante : [Google Maps](#)

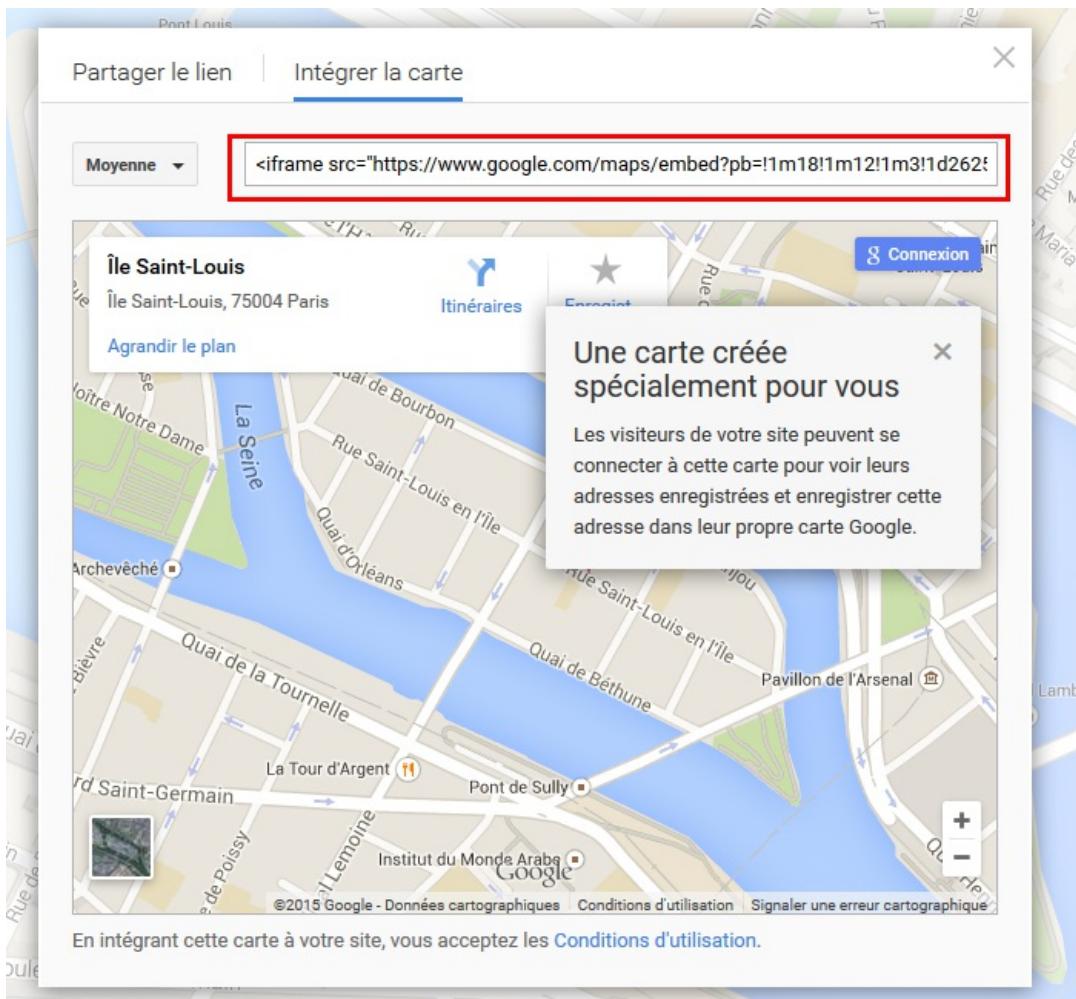
Saisir votre adresse dans la case de saisie de google :



Une fois que vous avez cliqué sur le lien "partager ou intégrer", une nouvelle fenêtre s'ouvre :



Nous allons donc choisir d'intégrer la carte et allons copier le code-source <iframe> pour le coller dans notre code html.



Voici ce que pourrait donner notre code html :

```
googleMap.html
```

```
<!Doctype html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Ma GoogleMap</title>
    <link rel="stylesheet" href="gmap.css">
  </head>
  <body>
    <iframe src="https://www.google.com/maps/embed?
pb=1m181m12!1m3!1d2625.346257453956!2d2.356677299999896!3d48.8516073!2m3!1f0!2f0!3f0!3m2!1i1024!2i768!4f13.1!3m3!1m2!1s0x47e671fc&bd4e8ee
Louis%2C+75004+Paris!5e0!3m2!1sfr!2sfr!4v1430829876812" width="600" height="450" frameborder="0" style="border:0"></iframe>
  </body>
</html>
```

Ce code doit être adapté en fonction du lieu que vous souhaitez cibler

Résultat

Insérer une vidéo YouTube dans une page web

Pour insérer une vidéo YouTube dans une page web, vous devez vous rendre sur le site officiel de [YouTube](#).

Rentrez dans l'une des vidéos de votre choix et cliquez ensuite sur la partie "partager" et "intégrer" :



Une nouvelle planète découverte, peut-être habitable selon les astronomes de la NASA

![Screenshot of a YouTube video page for 'euronews (en français)'. The video has 80 499 subscribers and 13 149 likes. A red box highlights the 'Partager' button. A red arrow points down to the 'Intégrer' button in the sharing menu, which contains the embed code: <iframe width=](https://www.youtube.com/embed/0AAAsU-02d4c)

euronews (en français)

S'abonner 80 499 13 149

+ Ajouter à Partager ... Plus

Partager Intégrer E-mail

<iframe width="560" height="315" src="https://www.youtube.com/embed/0AAAsU-02d4c" frameborder="0" allowfullscreen></iframe>

Copiez le code-source pour le récupérer et saisissez le dans un fichier de code :

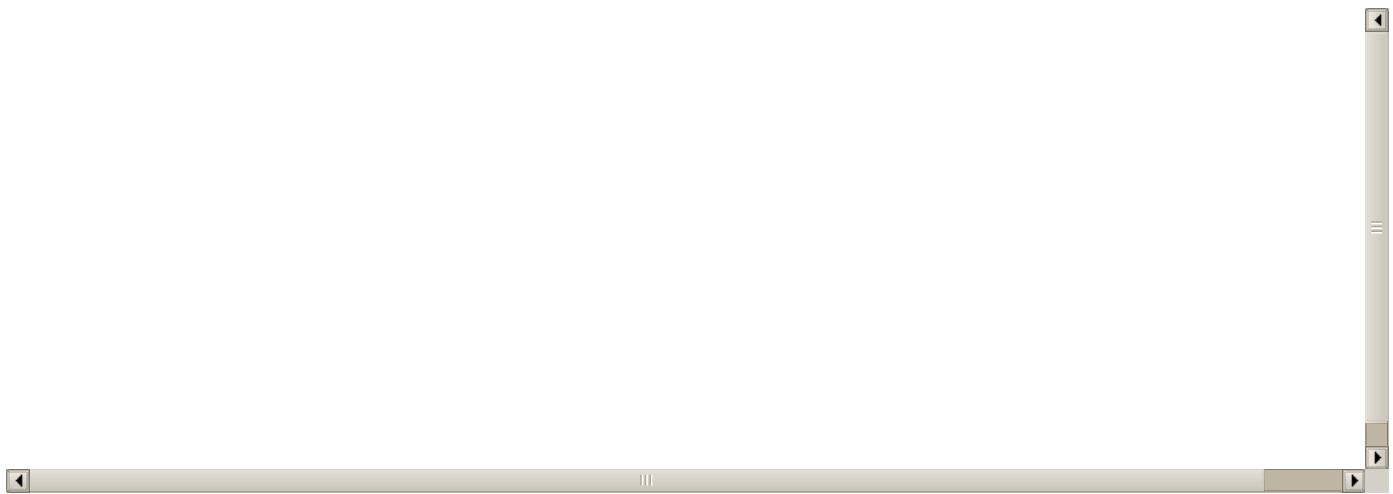
```
YouTube.html
```

```
<!Doctype html>
<html>
<head>
    <meta charset="utf-8">
    <title>Ma Vidéo YouTube</title>
    <link rel="stylesheet" href="yt.css">
</head>
<body>
    <iframe width="560" height="315" src="https://www.youtube.com/embed/0AAAsU-02d4c" frameborder="0" allowfullscreen></iframe>
</body>
</html>
```

Noter qu'il faut parfois vérifier la présence du protocole `http://` en début d'url (pas toujours fourni par YouTube) et l'écrire soi-même si cela est nécessaire.

Les valeurs pour les attributs `width(largeur)` et `height(hauteur)` peuvent être changées.

Résultat



Pour faire démarrer une vidéo YouTube à un moment précis ou un temps donné, vous pouvez utiliser le complément : ?start=10 (10 pour 10 secondes).

YouTube2.html

```
<!Doctype html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Ma Vidéo YouTube</title>
    <link rel="stylesheet" href="yt.css">
  </head>
  <body>
    <object width="560" height="315"><embed src="http://www.youtube.com/v/nSWfyx7pgfk&hl=fr&fs=1&start=10"
allowfullscreen="true"></embed></object>
  </body>
</html>
```

Noter ici l'utilisation de la balise `<object>` et `<embed>`.

Vous n'avez qu'à remplacer le code de la vidéo : `nSWfyx7pgfk` par le votre et vous pourrez ainsi garder la même structure de code.

Résultat

Pour lancer et démarrer une vidéo YouTube automatiquement : `?autoplay=1` ou `&autoplay=1`.

YouTube3.html

```
<!Doctype html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Ma Vidéo YouTube</title>
    <link rel="stylesheet" href="yt.css">
  </head>
  <body>
    <object width="560" height="315"><embed src="http://www.youtube.com/v/nSWfyx7pgfk&hl=fr&fs=1&start=10&autoplay=1"
allowfullscreen="true"></embed></object>
  </body>
</html>
```

Généralement, les sites web publient des vidéos grâce à YouTube plutôt que des fichiers locaux, pour plusieurs raisons:

1. Stocker sur YouTube, le fichier ne prend pas de place sur notre serveur (avec plusieurs vidéos de 50 ou 100 Mo, cela grimpe vite).
2. YouTube fournit une visibilité supplémentaire, les vidéos seront visitées sur votre site mais aussi sur celui de YouTube avec le trafic que l'on connaît.
3. On profite des fonctionnalités YouTube, on n'a pas besoin de développer le lecteur et pas non plus le bouton de partage, des partenaires peuvent prendre notre vidéo et la partager aussi (si on le souhaite).

87

Intégrer une vidéo locale avec la balise <video>

Pour intégrer notre propre vidéo, nous allons nous servir de la balise <video> ayant fait son apparition lors de la dernière mise à jour du langage (HTML5).

Si vous souhaitez réaliser ces tests, vous aurez besoin de récupérer 3 fichiers : [CIC - Fichier1](#) - [CIC - Fichier2](#) - [CIC - Fichier3](#).

Intégrer une vidéo au format .ogv

```
video.html
```

```
<!Doctype html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Ma Vidéo Perso</title>
    <link rel="stylesheet" href="video.css">
  </head>
  <body>
    <p>cic.ogv</p>
    <video preload controls poster="cic.gif"> <!-- video ogv -->
      <source src="cic.ogv" type="video/ogg">
    </video>
  </body>
</html>
```

Intégrer une vidéo au format .webm

```
video.html
```

```
<!Doctype html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Ma Vidéo Perso</title>
    <link rel="stylesheet" href="video.css">
  </head>
  <body>
    <p>cic.webm</p>
    <video preload controls poster="cic.gif"> <!-- video webm -->
      <source src="cic.webm">
    </video>
  </body>
</html>
```

Intégrer une vidéo au format .mp4

```
video.html
```

```
<!Doctype html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Ma Vidéo Perso</title>
    <link rel="stylesheet" href="video.css">
  </head>
  <body>
    <p>cic.mp4</p>
    <video preload controls poster="cic.gif"> <!-- video mp4 -->
```

```
<source src="cic.mp4">
</video>
</body>
</html>
```

Intégrer une vidéo compatible sous différents formats

```
video.html
?
<!Doctype html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Ma Vidéo Perso</title>
    <link rel="stylesheet" href="video.css">
  </head>
  <body>
    <p>cic</p>
    <video preload controls autoplay poster="cic.gif" > <!-- video ogv, webm, mp4 : 1 seule video avec 3 sources différentes--&gt;
      &lt;source src="cic.ogv" type="video/ogg"&gt;
      &lt;source src="cic.webm"&gt;
      &lt;source src="cic.mp4"&gt;
    &lt;/video&gt;
  &lt;/body&gt;
&lt;/html&gt;</pre>
```

Les attributs de la balise <video>

La balise <video> nous permet d'intégrer plusieurs attributs comme `preload` pour permettre un préchargement avant que l'internaute clique sur lecture.

Si vous n'ajoutez pas l'attribut `preload`, la vidéo sera chargée uniquement au moment où l'internaute cliquera sur "lecture".

L'attribut `controls` nous permet d'obtenir la barre de contrôle (lecture, pause, etc).

L'attribut `autoplay` permet de démarrer automatiquement la vidéo.

L'attribut `loop` (non présent dans cet exemple) permet d'effectuer une répétition sur la lecture de la vidéo.

L'attribut `poster` permet de recouvrir la vidéo par une image de notre choix plutôt qu'avoir l'aperçu de la première seconde de vidéo.

La balise <source> nous permet de préciser le fichier que l'on doit lire.



➤ Attention

Pensez à faire des tests de compatibilité sur différents navigateurs !!

Lors du dernier exemple, nous démarrons avec une source ogv, si le navigateur ne peut pas lire ogv, il essaiera de lire la source webm, si ce n'est pas possible il tentera de lire la source mp4.

Cela lui permettra d'avoir plusieurs options afin d'être sûr que la vidéo puisse se lire.

Résultat

C'est à la mode ! intégrer une vidéo en fond de page avec la balise `video` de html5. Pour se faire, il faut faire attention à ne pas prendre une vidéo trop lourde en terme de poids.

Récupérons ce fichier : [videoBackground.webm](#)

Ecrivons le code :

```
video-fond.html
```

```
<!Doctype html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Ma Vidéo en Fond</title>
    <link rel="stylesheet" href="video-fond.css">
  </head>
  <body>
    <video autoplay loop poster="videoBackground.jpg">
      <source src="videoBackground.webm" type="video/webm">
      <source src="videoBackground.mp4" type="video/mp4">
    </video>
    <div class="contenu">
      <p>Le texte ... Le texte ...
        Le texte ... Le texte ... Le texte ... Le texte ... Le texte ... Le texte ...
        Le texte ... Le texte ... Le texte ... Le texte ... Le texte ... Le texte ...
        Le texte ... Le texte ... Le texte ... Le texte ... Le texte ... Le texte ...
        Le texte ... Le texte ... Le texte ... Le texte ... Le texte ... Le texte ...
        Le texte ... Le texte ... Le texte ... Le texte ... Le texte ... Le texte ...
        Le texte ... Le texte ... Le texte ... Le texte ... Le texte ... Le texte ...
        Le texte ... Le texte ... Le texte ... Le texte ... Le texte ... Le texte ...
        Le texte ... Le texte ... Le texte ... Le texte ... Le texte ... Le texte ...</p>
    </div>
  </body>
</html>
```

La balise `<video>` nous permet de déclarer l'intégration d'une vidéo et la balise `<source>` permet de préciser le chemin de la vidéo à lire.

L'attribut `autoplay` permet la lecture automatique.

L'attribut `loop` permet d'effectuer une lecture en boucle (répétition).

video-fond.css :

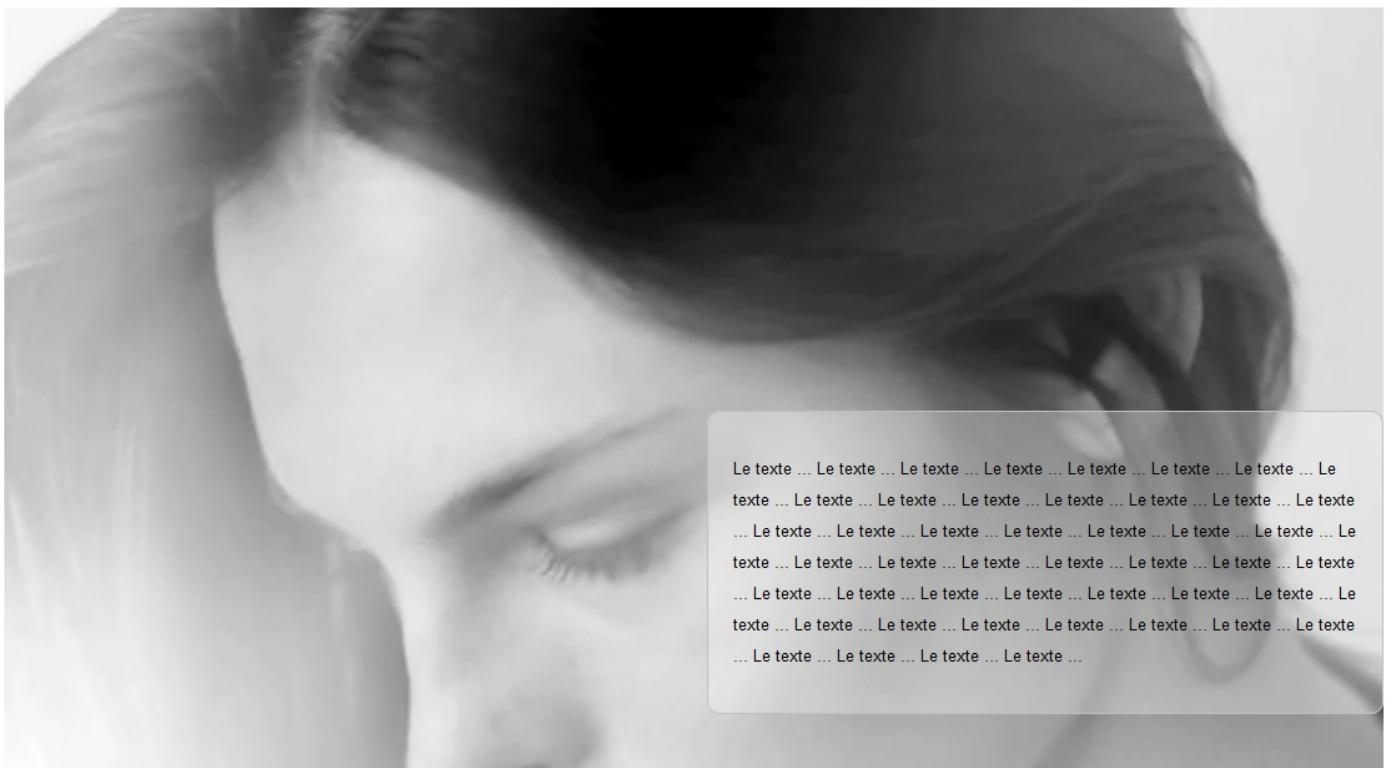
```
video{
  position: fixed;
  top: 0;
  left: 0;
  min-width: 100%;
  min-height: 100%;
  z-index: -1000;
  background: url(videoBackground.jpg) no-repeat;
  background-size: cover;
  /* width: auto; height: auto; */
}
div{
  font-family: arial;
  background: rgba(255,255,255,0.3);
  width: 500px;
  min-height: 300px;
  float: right;
  margin: 150px 0 0 0;
  padding: 20px;
  border-radius: 10px;
  border: 1px solid #c0c0c0;
}
```

Décryptons ensemble le code css :

- `Video: background` cela permet d'éviter un fond noir lorsque la première lecture se termine et que la 2e redémarre avec `loop`
- `Video: background-size: cover;` cela permet au `background` de couvrir tout l'écran
- `Video: position : fixed;` -- maintien la video fixée même si l'on descend dans la page web avec l'ascenseur (scroll).
- `Video: top : 0 ;` -- permet d'être à 0 pixel du haut et ainsi éviter un décalage.
- `Video: left : 0;` -- permet d'être à 0 pixel de la gauche et ainsi éviter un décalage.
- `Video: min-width: 100%;` -- largeur minimum
- `Video: min-height: 100%;` -- hauteur minimum
- `Video: z-index : -1000;` permet de définir une valeur qui détermine la position de l'élément au niveau de la superposition (premier plan, second plan).
- `div : background: rgba(255,255,255,0.3);` -- permet d'avoir un fond atténué.

- *div* : width: 500px; largeur
- *div* : min-height: 300px; hauteur minimum
- *div* : float: right; élément flottant
- *div* : margin: 150px 0 0 0; marges extérieures
- *div* : padding: 20px; marges intérieures.
- *div* : border-radius: 10px; bordure arrondie
- *div* : border: 1px solid #c0c0c0; bordure

Résultat



89

Intégrer une musique audio avec la balise <audio>

Pour intégrer une musique ou un son audio dans une page web en html, nous allons utiliser la balise <audio>.

Pour que l'exemple fonctionne, vous allez avoir besoin du fichier suivant : [Son mp3 à télécharger](#)+ [Son Ogg à télécharger](#)

Le code :

```
audio.html
```

```
<!Doctype html>
<html>
<head>
<meta charset="utf-8">
<title>Mon son audio</title>
<link rel="stylesheet" href="audio.css">
</head>
<body>
```

```
<audio src="son.mp3" controls preload></audio>
<audio src="son.ogg" controls></audio>
</body>
</html>
```

Nous ajoutons notre musique au format mp3 et pouvons également l'ajouter au format .ogg pour les navigateurs ne supportant pas l'extension mp3.

`controls` est un attribut qui permet d'utiliser les contrôles audios classiques (lecture, pause...).

`autoplay` est un attribut qui permet d'utiliser les contrôles audios classiques (lecture, pause...).

`preload` permet un préchargement avant que l'internaute clique sur lecture.

L'attribut `preload` peut prendre 3 valeurs :

`none` = pas de préchargement (économise donc de la bande passante si le contenu audio n'est pas lu)

`auto` = préchargement

`metadata` = préchargement mais uniquement des métadonnées du fichier audio.

Résultat

90

Intégrer une animation Flash

Pour intégrer une animation flash dans une page web en html, nous allons utiliser la balise `<object>`.

Il faut savoir que les produits Apple (iphone, ipad) ne supportent pas la technologie flash et donc que cette technologie est de moins en moins utilisée (pareil pour les langages AS ActionScript).

Pour que l'exemple fonctionne, vous allez avoir besoin du fichier suivant : [Animation a télécharger](#)

Ecrivons le code :

```
<!Doctype html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Mon son audio</title>
    <link rel="stylesheet" href="animationFlash.css">
  </head>
  <body>
    <object>
      <embed src="animation.swf"></embed>
    </object>
  </body>
</html>
```

animationFlash.html

La balise `<object>` permet de définir une zone d'animation.

La balise `<embed>` permet de préciser la source de l'animation.

Autre exemple :

```
<!Doctype html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Mon son audio</title>
    <link rel="stylesheet" href="animationFlash.css">
  </head>
```

animationFlash2.html

```

<body>
  <object type="application/x-shockwave-flash" data="animation.swf" width="800" height="250">
    <param name="movie" value="animation.swf">
    <param name="wmode" value="transparent">
    <param name="quality" value="high">
    <p>L'animation flash n'est pas prise en charge</p>
  </object>
</body>
</html>

```

Résultat

Balise Html et Propriété Css

91

Les balises

Balises Principales

Balise	Description	Exemple
<Doctype html>	Doctype est une balise permettant de préciser le type du document, la version du langage Html utilisé et est indispensable pour valider notre code avec le W3C	<Doctype html>
<html>	Balise principale contenant le document entier (indispensable)	<html></html>
<head>	Entêtes du document (indispensable)	<head></head>
<body>	Corps du document, contenu de la page web (indispensable)	<body></body>

Balises d'en-têtes

(à l'intérieur de la balise head)

Balise	Description	Exemple
<meta>	Informations sur la page web (description, encodage, etc.)	<meta charset="utf-8"> <meta description="description du site web">
<link>	Liaison à une feuille de style, police CSS3, etc. <i>Il s'agit d'une balise autofermante.</i>	<link rel="stylesheet" href="style.css">
<style>	Intègre du code CSS directement dans le document Html	<style>...</style>
<script>	Intègre du code JavaScript	<script src="http://"/> </script> <script> votre code JS</script>
<title>	Titre de la page web	<title> Titre de la page web </title>

Voici un exemple récapitulant les principales balises que nous venons de voir :

EXEMPLE HTML

```
<!Doctype html>
<html>
  <head>
    <meta charset="utf-8">
    <link rel="stylesheet" href="style.css">
  </head>
  <body>
    ...
  </body>
</html>
```

Balises de structuration de contenu sémantique (zoning)

(zoning)

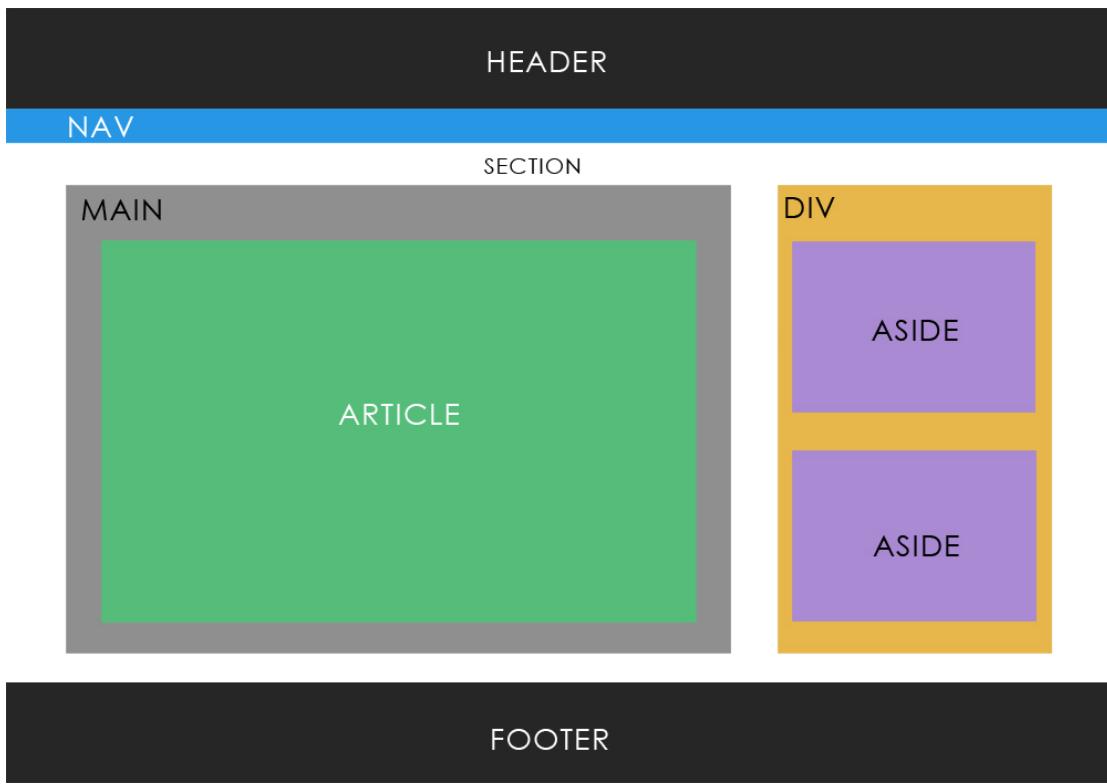
Balise	Description	Exemple
<header>	Haut de site	<header> </header>
<nav>	Zone de navigation avec des liens (menu).	<nav> </nav>
<section>	Partie d'une page web	<section> </section>
<main>	Contenu principal	<main> </main>
<article>	Contenu thématique	<article> </article>
<aside>	Contenu apparenté	<aside> </aside>
<footer>	Bas de page	<footer> </footer>

Balises de structuration de contenu neutre

(zoning)

Balise	Description	Exemple
<div>	Zone de site (Div = Division) de type block	<div> </div>
	Balise d'encapsulation de type inline	

Voici ceux à quoi pourrrait ressembler l'utilisation de ces balises dans le cadre de la structuration d'un site web :



Balise de lien hypertext

Balise	Description	Exemple
<a>	Permet de faire un lien	 Lien

Balises de contenu texte

Balise	Description	Exemple
<h1>	Titre principal (de niveau 1)	<h1> Titre 1 </h1>
<h2>	Sous Titre (de niveau 2)	<h2> Titre 2 </h2>
<h2>	Titre (de niveau 3)	<h2> Titre 3 </h2>
<h4>	Titre (de niveau 4)	<h4> Titre 4 </h4>
<h5>	Titre (de niveau 5)	<h5> Titre 5 </h5>
<h6>	Titre (de niveau 6)	<h6> Titre 6 </h6>
<p>	Paragraphe	<p> texte dans un paragraphe </p>
 	Texte en gras	 texte en gras
<strike> <s>	Texte barré	<strike> texte barré </strike> texte barré <s> texte barré </s>
<cite>		<cite> ... </cite>

<blockquote> <q>	Citation	<cite> Citation... </cite>
 	Passage à la ligne (balise autofermante)	 ?
<hr>	Tire un trait de séparation (balise autofermante)	<hr> ?
 <i>	Texte en italique	 ... <i> ... </i> ?
<u> <u>	Texte souligné	<u> ... </u> ?
<mark>	Texte marqué (surlignage)	<mark> texte </mark> ?
<small>	Texte petit	<small> texte </small> ?

Balises de contenu média

Balise	Description	Exemple				
	Affiche une image (balise autofermante)	 <source>	Vidéo en lecture	<video> <source src="mavideo.mp4"> </video> ?		
<audio>	Permet de lire une musique	<audio src="musique.mp3" controls></audio> ?				
<object> <embed> <param>	Intègre une animation flash, ou une autre ressource type applet java (balise applet non prise en charge par Html5)	<object> <embed src="animation.swf"></embed> </object> ?				
<iframe>	Permet d'introduire un contenu dans un autre (exemple : GoogleMap, Vidéo YouTube, etc.)	<iframe src="..."></iframe> ?				
<figure> <figcaption>	Permet de contenir un élément média et d'y ajouter une légende	<figure> <figcaption> legend <figcaption> </figure> ?				
<map> <area>	Map définit l'image à traiter, area permet de créer des zones cliquables sur l'image	<map name="monImage"> <area shape="rect" href="http://www.site.com" title="..." alt="..." target="_blank" coords="12,12,80,80"> <area shape="rect" href="http://www.site.com" title="..." alt="..." target="_blank" coords="20,20,90,90"> </map> 	Permet de dessiner un graphique (avec l'aide d'un langage complémentaire type JavaScript)	<canvas> ... </canvas> 	Ouvre une liste non ordonnée	 ... ?

	Ouvre une liste ordonnée	 ...
	Element de la liste	
<dl>	Définition d'une liste	<dl></dl>
<dt>	Définition d'un terme	<dt></dt>
<dd>	Définition description	<dd></dd>

Balises de tableau

Balise	Description	Exemple
<table>	Ouvre un tableau de données	<table> ... </table>
<caption>	Titre du tableau	<caption> ... </caption>
<tr>	Ligne du tableau	<table> <tr> ... </tr> </table>
<thead>	Partie haute du tableau	<table> <tr> <thead> ... </thead> </tr> </table>
<tbody>	Corps du tableau	<table> <tr> <tbody> ... </tbody> </tr> </table>
<tfoot>	Bas du tableau	<table> <tr> <tfoot> ... </tfoot> </tr> </table>
<th>	cellule d'entête	<table> <tr> <th> ... </th> </tr> </table>
<td>	cellule	<table> <tr> <td> ... </td> </tr> </table>

Balises de formulaire

Balise	Description	Exemple
--------	-------------	---------

<form>	Formulaire	<form> ... </form>
<label>	Etiquette de formulaire (souvent reliée à une case de saisie)	<label for="..."> ... </label>
<input>	Champ de formulaire (souvent pour la saisie)	<input type="text">
<textarea>	Zone de saisie	<textarea> ... </textarea>
<select> <option> <optgroup>	Liste déroulante (select) à option (option), parfois regroupée (optgroup)	<select> <option> ... </option> <option> ... </option> </select>
<datalist>	Liste de suggestions	<datalist> <option> ... </option> <option> ... </option> </datalist>
<fieldset> <legend>	Groupe de champs (fieldset) avec une légende (legend)	<fieldset> <legend>...</legend> ... </fieldset>

[Cliquez ici pour voir les balises de formulaire en détails](#)

Balises diverses

Le lien nous emmènera vers l'adresse site.com/boutique.

Balise	Description	Exemple
<!-- -->	Commentaire (non interprété)	<!-- commentaire -->
<base>	Etiquette de formulaire (souvent reliée à une case de saisie)	Aller sur notre boutique
<bdo> <bdi>	Texte avec direction de lecture différente	<bdo> <bdi> ... </bdi> </bdo>
<button>	Bouton	<button> ... </button>
<code>	Définit une portion de code	<code> ... </code>
<colgroup> <col>	Groupe de colonnes	<colgroup> <col> ... </col> <col> ... </col> </colgroup>
<summary> <details>	Intitulé et détails sur un sujet	<details> </details>
<dfn>	Définition	<dfn> </dfn>

<dialog>	Boite de dialogue dans une page web	<dialog> </dialog>
<abbr> (anciennement : <acronym>)	Définit une abbréviation	<abbr> ... </abbr>
<address>	Précise l'adresse de l'auteur du document	<address> ... </address>
<ins>	Texte inséré dans un document	<ins> ... </ins>
<kbd>	Définit la saisie au clavier	<kbd> ... </kbd>
<keygen>	Définit une clé de sécurité pour les échanges avec un formulaire	<keygen keytype="" challenge="" name=""></keygen>
<menu> <menuitem>	Définit une barre d'outils	<menu type="" label=""> </menu>
<noscript>	Detecte et propose un contenu alternatif pour les internautes n'ayant pas la possibilité d'utiliser JavaScript	<noscript></noscript>
<output>	Affiche le résultat d'un calcul	<output></output>
<pre>	Texte préformaté	<pre></pre>
<progress>	Barre de progression	<progress value="500" max="1000"></progress>
<sub>	Texte en indice	
<time>	Définit l'heure et la date	<time>09:30</time>
<track>	Définit du texte pour les éléments médias	<track>...</track>
<var>	Permet de mettre en forme une variable lors de l'affichage (programmation)	<var>...</var>
<xmp>	Permet de ne pas interpréter du code html au sein d'une page html	<xmp> <div></div> </xmp>
<hgroup>	Regroupement d'éléments de haut niveau	<hgroup> ... </hgroup>

D'autres balises non officielles existent :

Balise	Description	Exemple
<marquee>	Permet de faire défiler du texte	<marquee> ... </marquee>
<blink>	Permet de faire clignoter du texte	<blink> ... </blink>

Certaines balises non officielles ne vous permettront pas de passer la validation W3C, et pas non plus une compatibilité tout navigateur, il faut donc éviter de les utiliser. Il existe des alternatives pouvant avoir des comportements similaires avec CSS ou JS.

Les balises absentes de ces listes sont soit trop anciennes et par conséquent plus valides w3c, soit peu importantes (ou soit un oubli :p, et oui ça peut

Liste des propriétés CSS pour la mise en forme de texte

Propriété	Question	Description	Exemple
color	Comment changer la couleur du texte ?	Couleur du texte	<code>color: red;</code>
font-size font-size-adjust font	Comment changer la taille du texte ?	Taille du texte	<code>font-size: 1em;</code>
font-style font	Comment passer un texte en italique ?	Orientation du texte	<code>font-style: italic;</code>
font-weight font	Comment mettre un texte en gras ?	Fonte du texte	<code>font-weight: bold;</code>
font-family font	Comment modifier la police d'écriture ?	Police d'écriture	<code>font-family: verdana;</code>
@font-face font	Comment importer une police d'écriture css3 ?	Import d'une police personnalisée	<code>@font-face{}</code>
font-variant font	Comment modifier l'apparence du texte ?	Type de lettre (ex: petites capitales)	<code>font-variant: small-caps;</code>
text-transform	Comment mettre des majuscules au texte ?	Transformation du texte (ex: majuscules)	<code>text-transform: capitalize;</code>
text-align text-align-last	Comment centrer un texte ? Comment justifier un texte ?	Alignment du texte	<code>text-align: center; text-align: justify;</code>
vertical-align	Comment aligner du texte vertical ?	Alignment du texte	<code>vertical-align: middle;</code>
line-height	Comment modifier l'interligne du texte ?	Interligne du texte	<code>line-height: 20px;</code>
text-indent	Comment modifier l'indentation du texte ?	Indentation du texte	<code>text-indent: 20px;</code>
letter-spacing word-spacing	Comment modifier l'espace entre les lettres d'un texte ? Comment modifier l'espace entre les mots d'un texte ?	Espacement des lettres entres elles	<code>letter-spacing: 2px; word-spacing: 3px;</code>
white-space	Comment modifier la césure d'un texte ?	Césure du texte	<code>letter-spacing: nowrap;</code>
word-wrap	Comment forcer la césure d'un texte ?	Règle de césure pour le texte	<code>word-wrap: break-word;</code>
text-shadow	Comment ajouter de l'ombre sur un texte ?	Ombre sur le texte	<code>text-shadow: 3px 3px 3px #c0c0c0;</code>
text-decoration text-decoration-line text-decoration-style	Comment retirer l'effet souligné sur un lien ? Comment ajouter un effet souligné ?	Présentation du texte sous forme de lien	<code>text-decoration: none; text-decoration: underline;</code>

Liste des propriétés CSS pour les fonds

Propriété	Question	Description	Exemple
background background-color background-image	comment changer la couleur de fond ? comment mettre une image de fond ?	couleur ou image de fond	background: red; background: url(image.png);
background-attachment	comment fixer une image au fond de la page ?	Fixation de l'image de fond	background-attachment: fixed;
background-repeat	comment stopper la répétition du fond de page ?	Répétition du fond de page	background-repeat: no-repeat;
background-position	comment modifier la position du fond de la page ?	position de l'image de fond (arrière plan)	background-position: left top;
background-cover	comment adapter la taille d'une image au fond de la page ?	couverture de fond	background-size: cover;
background-linear-gradient	comment faire un dégradé ?	Dégradé	background: linear-gradient(#f6eadd, #f9dac3);
background-origin	comment mettre un fond précisément sous le texte ?	Positionnement du fond par rapport au contenu	background-origin: content-box;
opacity background	comment gérer une opacité ?	Opacité	opacity: 0.8; /* donne un effet opaque également au texte contenu dans une zone, pas idéal */ background-color: rgba(255, 255, 255, 0.3); /* a privilégié */

Liste des propriétés CSS appliquable pour les zones

Propriété	Question	Description	Exemple
width min-width max-width	Comment modifier la largeur d'une zone ?	Largeur	width: 50%; min-width: 30px;
height min-height max-height	Comment modifier la hauteur d'une zone ?	Largeur	height: 10vh; min-height: 10px;
margin margin-top margin-right margin-bottom margin-left	Comment modifier la marge extérieure d'une zone ?	Marges extérieures	margin: 10px 9px 8px 7px; /* 10px en haut - 9px à droite - 8px en bas - 7px à gauche */ margin: 10px 7px; /* 10px en haut et en bas - 7px à gauche et à droite */ margin: 10px 9px 8px; /* 10px en haut - 9px à droite - 8px en bas. */ margin: 10px; /* 10px tout autour (haut, droite, bas, gauche). */ margin-top: 10px; /* top en haut */ margin-right: 10px; /* right à droite */ margin-bottom: 10px; /* bottom en bas */ margin-left: 10px; /* left à gauche */
padding padding-top padding-right padding-bottom padding-left	Comment modifier la marge intérieure d'une zone ?	Marges intérieures	padding: 10px 9px 8px 7px; /* 10px en haut - 9px à droite - 8px en bas - 7px à gauche */ padding: 10px 7px; /* 10px en haut et en bas - 7px à gauche et à droite */ padding: 10px 9px 8px; /* 10px en haut - 9px à droite - 8px en bas. */ padding: 10px; /* 10px tout autour (haut, droite, bas, gauche). */ padding-top: 10px; /* top en haut */ padding-right: 10px; /* right à droite */ padding-bottom: 10px; /* bottom en bas */ padding-left: 10px; /* left à gauche */
border-width border	Comment modifier l'épaisseur d'une bordure ?	Epaisseur d'une bordure	border-width: 3px;

border-color border	Comment modifier la couleur d'une bordure ?	Couleur d'une bordure	<code>border-color: red;</code>
border-style border	Comment modifier le trait d'une bordure	Type de bordure	<code>border-style: dotted;</code>
border-radius border	Comment faire une bordure arrondie ?	Arrondie d'une bordure	<code>border-radius: 10px;</code>
border-top border-right border-bottom border-left border	Comment mettre une bordure seulement en bas ?	Position d'une bordure	<code>border-bottom: 2px;</code>
border-image border	Comment mettre une image en guise de bordure ?	Image de bordure	<code>border-image: url(border.png);</code>
box-shadow	Comment ajouter de l'ombre ?	Ombre d'une zone	<code>box-shadow: 3px 3px 3px #c0c0c0;</code>

Liste des propriétés CSS permettant de positionner les zones d'une page web

Propriété	Question	Description	Exemple
display	Comment modifier le type d'une zone ?	Type d'élément	<code>display: flex; display: block; display: inline; display: table;</code>
position top right bottom left	Comment modifier l'emplacement d'une zone ?	Position d'une zone	<code>position: absolute; position: relative; position: fixed; top: 50%; left: 0; /* right: 0; bottom: 0; */</code>
float clear	Comment mettre deux éléments côté à côté ?	Element flottant	<code>float: left; float: right; clear: left;</code>

Liste des propriétés CSS permettant de mettre en forme les listes

Propriété	Question	Description	Exemple
list-style-type list-style	Comment modifier le style d'une liste ?	Style de puces dans une liste	<code>list-style-type: circle; /* disc, square, decimal */</code>
list-style-position list-style	Comment modifier la position des puces dans une liste ?	Position des puces dans une liste	<code>list-style-position: inside;</code>
list-style-image list-style	Comment mettre une image à la place d'une puce dans une liste ?	Image dans une puce de liste	<code>list-style-image: url(monimage.png);</code>

Liste des propriétés CSS permettant de mettre en forme un formulaire

Propriété	Question	Description	Exemple
resize	Comment empêcher qu'un textarea soit redimensionné ?	Dimensions Textarea	resize: horizontal; ?

Liste des propriétés CSS permettant de mettre en forme les tableaux

Propriété	Question	Description	Exemple
border-collapse	Comment modifier les bordures d'un tableau ?	Applati le trait d'un tableau en fusionnant les cellules	border-collapse: collapse; ?
empty-cells	Comment afficher les cellules vides d'un tableau ?	Affichage des cellules vides	empty-cells: show; ?
caption-side	Comment afficher le titre d'un tableau en bas ?	Position du titre du tableau	caption-side: bottom; ?
border-spacing	Comment modifier l'écart entre les cellules d'un tableau ?	Ecart entre les cellules	border-spacing: 10px 30px; ?

Liste des propriétés CSS d'animation

Propriété	Question	Description	Exemple
animation, animation-delay, animation-direction, animation-duration	Comment gérer une animation ?	Animation	- ?
transform	-	Transform	transform: rotate(-360deg); ?
transition, transition-duration, transition-delay	Comment ... ?	Effet proressif	clip: rect(0px,60px,200px,0px); ?
marquee-direction	Comment faire défiler du texte ?	Défilement de texte	- ?

Liste des propriétés CSS d'affichage divers

Propriété	Question	Description	Exemple
z-index	Comment afficher un élément au dessus de l'autre ?	Permet de donner une priorité d'affichage lorsque 2 éléments se superposent	z-index: 1000; ?
overflow overflow-x overflow-y	Comment gérer la scrollbar (ascenseur, barre de défilement) ?	Scrollbar	overflow-x: hidden; ?
clip	Comment afficher une partie d'une image ?	Affichage coupé	clip: rect(0px,60px,200px,0px); ?

Propriétés diverses :

Propriété	Question	Description	Exemple
tab-size	-	Taille utilisé pour diffuser le texte	text-shadow: 3px 3px 3px #c0c0c0; ?

direction	Comment changer la direction du texte ?	Direction du texte	direction: rtl;
cursor	Comment changer le curseur ?	Curseur	cursor: pointer;

Vous pouvez aussi consulter les [sélecteurs](#) en CSS.

Cas pratique

93

Règle de priorité : Le !important en CSS

Il arrive parfois qu'en CSS certaines lignes se contredisent, exemple :

code Html

```
<p>Bonjour tout le monde !</p>
```

code Css

```
p{ color: red; }
p{ color: green; }
```

résultat

Bonjour tout le monde

Dans cet exemple, nous voyons qu'on demande à appliquer la couleur rouge sur les textes contenues dans la balise `<p>` paragraphe et juste en dessous, nous demandons la couleur verte.

Comment va réagir le navigateur ?

Il prendra en compte la dernière règle (la plus en bas du code), car il lit le code-source comme vous et moi : séquentiellement (ligne après ligne).

Vous me direz certainement, « mais pourquoi tu mets rouge puis vert comme couleur, tu devrais plutôt faire un choix ! »

Oui c'est vrai, mais parfois nous chargeons plusieurs feuilles de style déjà prédefinies (surtout avec l'utilisation des CMS) et pour personnaliser l'affichage il faut surcharger les classes CSS existantes en appliquant des propriétés complémentaires ou différentes.

Surcharger une classe CSS permet de ne pas la modifier à l'endroit où elle a été déclarée mais de l'appeler dans notre feuille de style pour lui donner des propriétés complémentaires ou différentes.

A la question : « Pourquoi nous ne modifions pas le CSS directement où il se trouve ? Ce serait plus facile ! »

En général nous ne modifions pas l'endroit où la classe CSS a été prédefinie car s'il s'agit d'un framework, d'un module (plugin) ou d'un cms il sera certainement mis à jour à un moment donné et les lignes css seront réinitialisées sans conservation de nos modifications (car cela ne fera pas partie de la mise à jour du framework, module ou cms).

C'est la raison pour laquelle vous devez garder à l'esprit que pour imposer une règle contraire en CSS, il faudra sans doute utiliser le `!important`, exemple :

code Html

```
<p>Bonjour tout le monde !</p>
```

code Css

```
p{ color: red !important; }
p{ color: green; }
```

Dans le cas ci-dessus, la couleur rouge sera appliquée :

résultat

Bonjour tout le monde

The screenshot shows a browser's developer tools with the "HTML" tab selected. The DOM tree on the left shows a single `<p>Bonjour tout le monde !</p>` node under the `<body>` node. The right panel displays the CSS styles for this element. There are two rules:

- `html body p { color: red; }` (text.html ligne 8)
- `p { color: green; }` (text.html ligne 9)

The `color: red;` property is highlighted in red, indicating it is the active style being applied.

Comme vous pouvez le constater, une des deux propriétés est barrée `color:green;`.

Vous pouvez aussi préciser la hiérarchie afin de prendre la priorité sur les lignes de CSS contraires qui pourraient se retrouver plus bas dans le code, exemple :

code Html

```
<p>Bonjour tout le monde !</p>
```

code Css

```
html body p{ color: red; }
p{ color: green; }
```

Par le fait que nous avons précisé les parents de la balise `<p>`, notre ligne css avec la couleur rouge prendra la priorité :

résultat

Bonjour tout le monde

The screenshot shows a browser's developer tools with the "HTML" tab selected. The DOM tree on the left shows a single `<p>Bonjour tout le monde !</p>` node under the `<body>` node. The right panel displays the CSS styles for this element. There are two rules:

- `html body p { color: red; }` (text.html ligne 8)
- `p { color: green; }` (text.html ligne 9)

The `color: green;` property is highlighted in red, indicating it is the active style being applied.

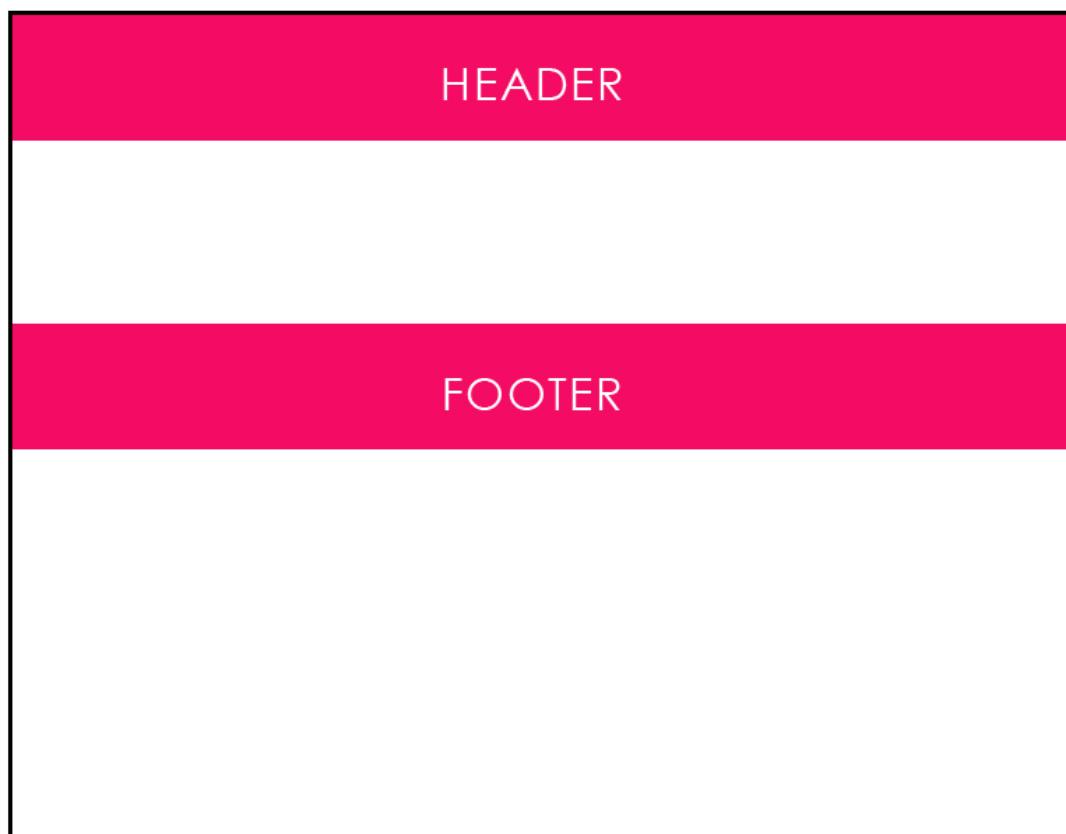
Là aussi, l'une des deux propriétés est barrée `color:green;`.

Pour conclure, vous pouvez écrire votre code CSS normalement et lorsque vous travaillerez avec l'aide d'un outil et que vous n'aurez pas forcément la priorité sur les lignes de code CSS, vous vous rappelerez que vous pourrez utiliser le `!important` ou préciser la hiérarchie pour prendre le dessus sur les autres règles CSS ! ;-).

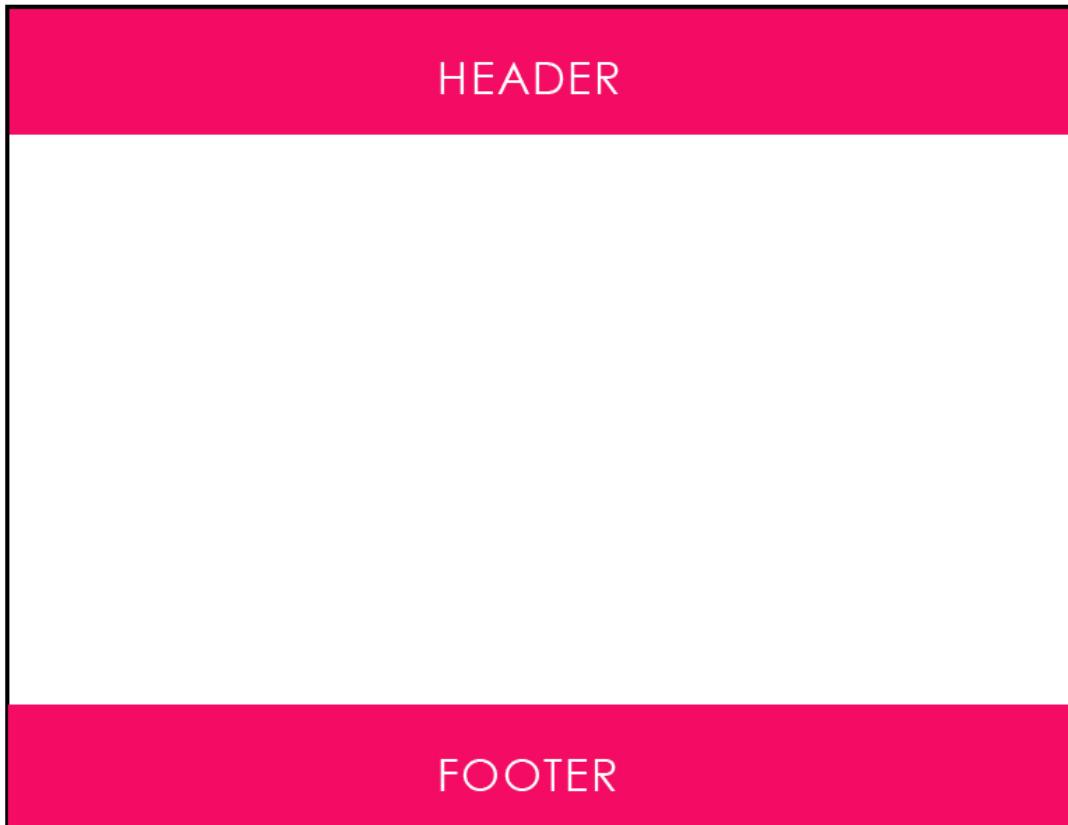
94

Comment maintenir un footer en bas de page ?

Problématique, si votre conteneur à une hauteur fixe (sans scrollbar vertical) et que votre footer remonte dans la page web



Vous pouvez adapter le code afin de viser ce résultat souhaité :



footer.html

footer.css

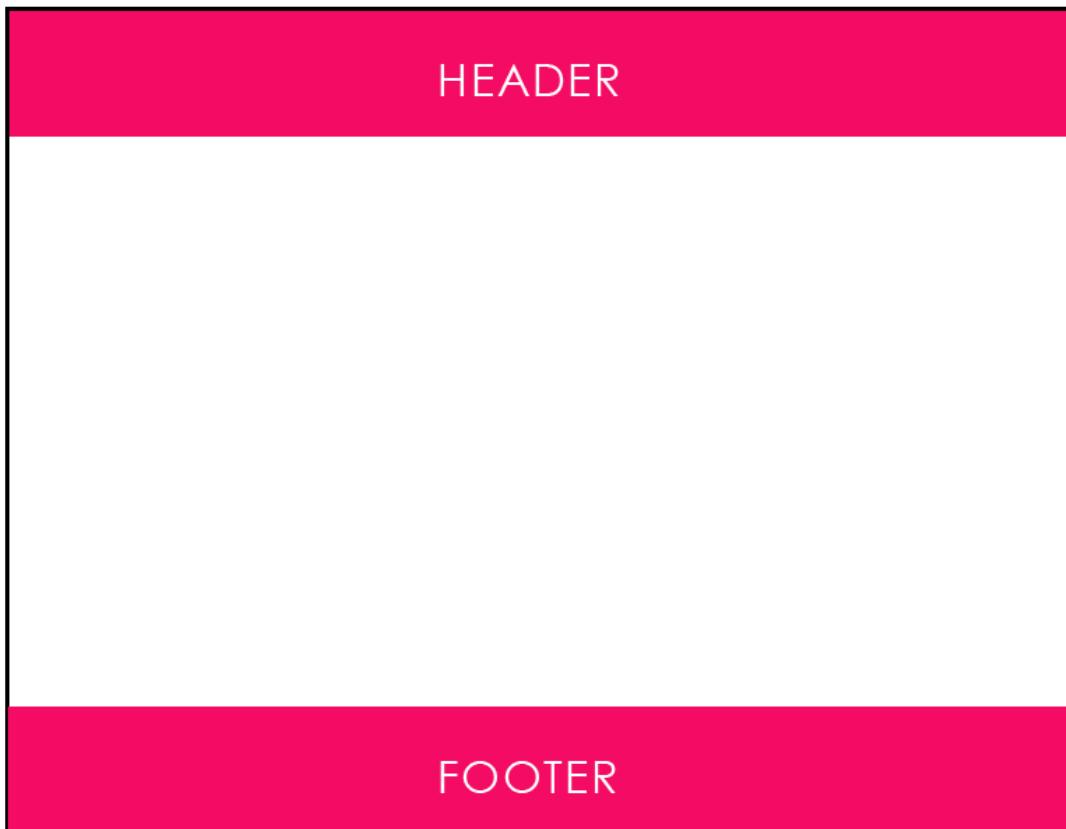
```
* {  
    margin: 0;  
}  
html, body {  
    height: 100%;  
}  
.conteneur {  
    min-height: 100%;  
    height: 100%;  
    margin: 0 auto -100px;  
}  
header{  
    height: 100px;  
    background: pink;  
}  
footer {  
    height: 100px;  
    position: relative;
```

```
background: pink;  
}
```

Si vous êtes sûr d'avoir une hauteur de conteneur fixe (sans scrollbar vertical), vous pourriez également mettre en oeuvre le code suivant :

```
footer.css  
?  
* {  
    margin: 0;  
}  
html, body {  
    height: 100%;  
}  
.conteneur {  
    min-height: 100%;  
    height: 100%;  
}  
header{  
    height: 100px;  
    background: pink;  
}  
footer {  
    height: 100px;  
    position: absolute;  
    width: 100%;  
    bottom: 0;  
    background: pink;  
}
```

Résultat



Si votre conteneur n'a pas une hauteur fixe, votre footer se positionnera automatiquement en bas de page sans besoin d'adaptation.

Et si votre conteneur possède parfois une hauteur faible et d'autres fois une hauteur plus importante, vous pourrez toujours appliquer la propriété `min-height` à votre conteneur.

Il arrive parfois que nous ayons besoin de centrer une zone verticalement dans une autre en hauteur.

Pour cela, il existe la propriété : `display flex` :

```
zoneVerticale.html
```

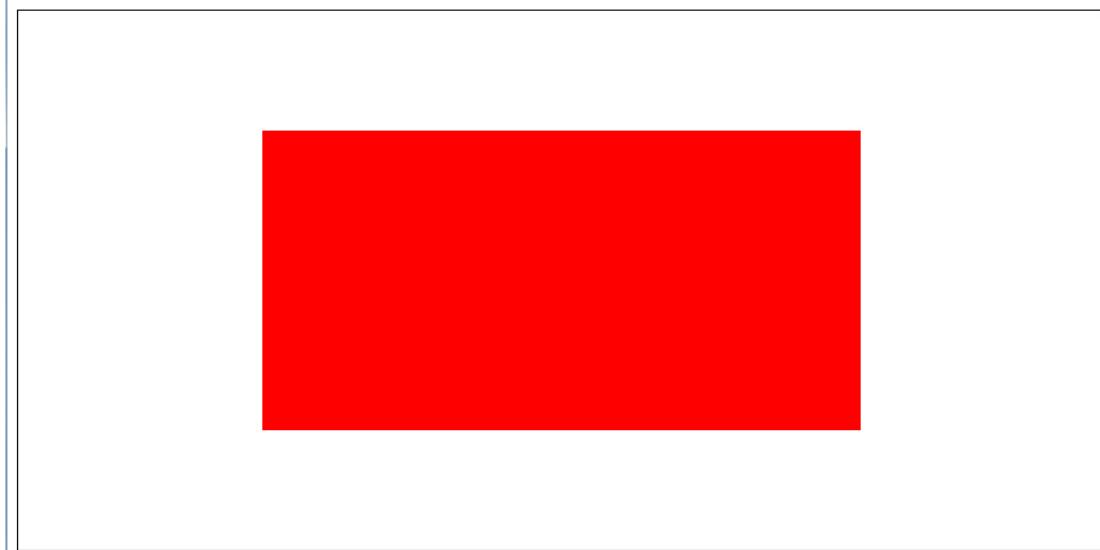
```
<!Doctype html>
<html>
  <head>
    <meta charset="utf-8">
    <link rel="stylesheet" href="zoneVerticale.css">
    <title>Ma Page</title>
  </head>
  <body>
    <div id="conteneur">
      <div id="flex">
        <div class="mazone"></div>
        <!--<div class="mazone"></div>-->
      </div>
    </div>
  </body>
</html>
```

```
zoneVerticale.css
```

```
#conteneur{
  height: 450px;
  border:1px solid #000;
}
#flex {
  height: 100%;
  display: flex;
  align-items: center;
  justify-content: center;
}
.mazone{
  background:red;
  width:250px;
  height:250px;
}
```

Quelques explications :

Résultat



Nous pouvons apercevoir que la zone `.mazone` avec un fond rouge apparaît bien à mi-hauteur de la zone `#flex` :

`#conteneur` nous permet d'avoir une div qui englobe le reste de notre page web.

`height: 450px;` permet de définir la hauteur.

`border:1px solid #000;` permet de définir la bordure.

```
#flex représente la div qui contient notre zone .mazone.  
height: 100%; permet de récupérer la hauteur de la div parent (100% soit 450px).  
display: flex; Affichage de type "flex" pour un affichage flexible.  
align-items: center; Propriété qui sera attribuée aux éléments enfants devenus éléments de type "flex-item" pour obtenir un centrage vertical.  
justify-content: center; Permet de centrer les zones enfants à l'horizontale au milieu de la div parent.
```

```
#mazone représente la div qui contient notre zone .mazone.  
background: red;; permet d'appliquer une couleur de fond.  
width: 250px; permet d'appliquer une largeur à la zone.  
height: 250px; permet d'appliquer une hauteur à la zone.
```

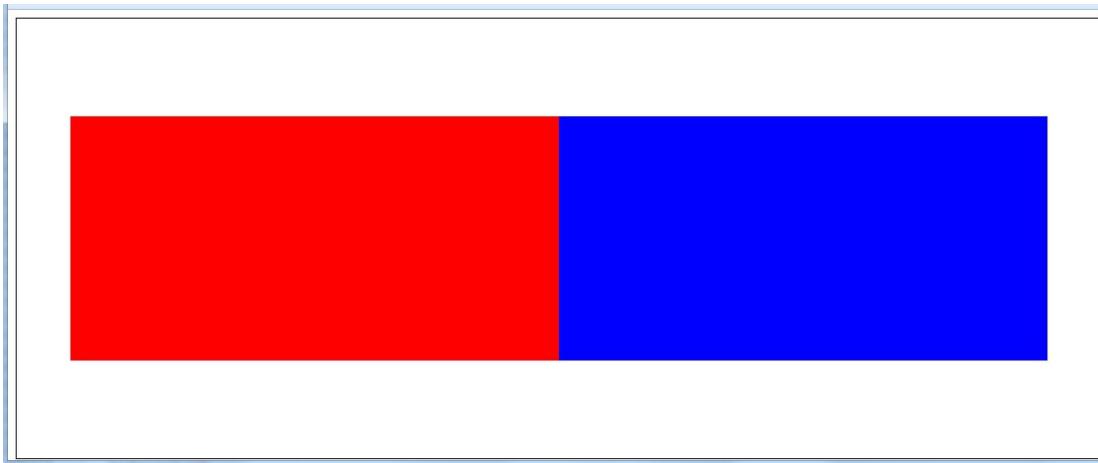
La propriété FLEX en détails

Prenons ce code en exemple :

```
flex1.html  
  
<!Doctype html>  
<html>  
  <head>  
    <meta charset="utf-8">  
    <link rel="stylesheet" href="flex1.css">  
    <title>Ma Page</title>  
  </head>  
  <body>  
    <div id="conteneur">  
      <div id="flex">  
        <div class="mazone"></div>  
        <!--<div class="mazone"></div>-->  
      </div>  
    </div>  
  </body>  
</html>
```

```
flex1.css  
  
#conteneur{  
  height: 450px;  
  border: 1px solid #000;  
}  
#flex {  
  height: 100%;  
  display: flex;  
  align-items: center;  
  justify-content: center;  
}  
.mazone{  
  background:red;  
  width:250px;  
  height:250px;  
}  
.mazone {  
  background:blue;  
  width:250px;  
  height:250px;  
}
```

Résultat



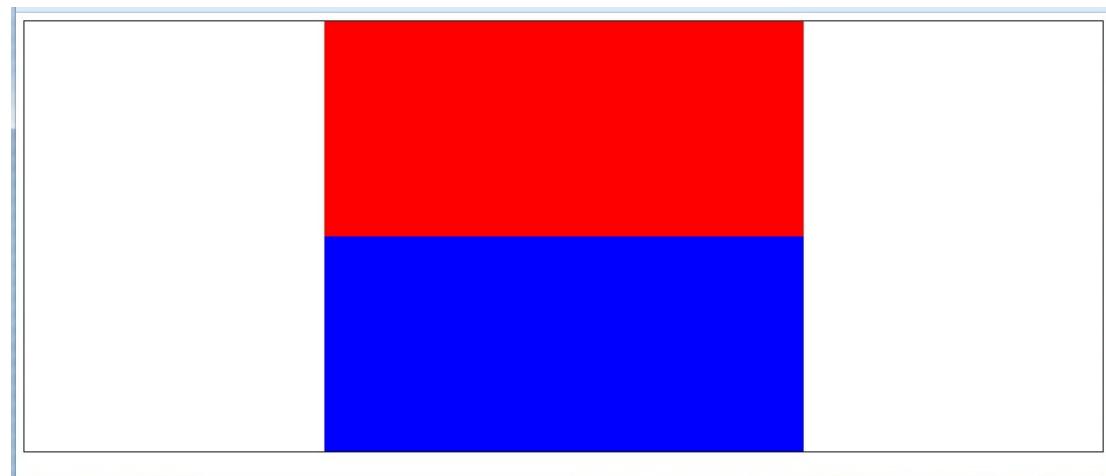
Les zones enfants d'une zone parent ayant la propriété `display: flex;` seront placées côte à côte automatiquement.

Si vous souhaitez mettre les 2 zones enfants l'une en dessous de l'autre (et non pas à côté de l'autre), vous devrez utiliser la propriété `flex-direction:column;`, exemple :

flex1.css

```
#conteneur{
  height: 450px;
  border:1px solid #000;
}
#flex {
  height: 100%;
  display: flex;
  align-items: center;
  justify-content: center;
}
.mazone{
  background:red;
  width:250px;
  height:250px;
  flex-direction:column;
}
.mazone2{
  background:blue;
  width:250px;
  height:250px;
}
```

Résultat



L'utilisation de la propriété `display: block;` ne fonctionnera pas dans ce cas présent.

Dans les débuts des langages Html et Css, les sites web étaient conçus à l'aide de tableaux avec les balises `table`, `tr`, `td`, cela n'est plus le cas aujourd'hui pour de l'affichage en FRONT (sauf pour représenter des données tabulaires, des statistiques, des affichages de backoffice ou des newsletter).

Cependant les tables avaient l'avantage de pouvoir centrer verticalement le texte (`vertical-align: middle;`) contenu dans leur cellule, ce qui est moins évident avec des zones `div`.

Alors nous n'allons pas retourner dans le passé pour construire nos sites web avec des tableaux mais pour centrer un texte verticalement à l'intérieur d'une zone en hauteur, nous aurons besoin de la propriété `display: table;` :

```
alignementVertical.html
?
<!Doctype html>
<html>
  <head>
    <meta charset="utf-8">
    <link rel="stylesheet" href="alignementVertical.css">
    <title>Ma Page</title>
  </head>
  <body>
    <div class="zone-principale">
      <p>Texte centré verticalement.</p>
    </div>
  </body>
</html>
```

Notre texte est contenu à l'intérieur d'une zone et voici le code css associé :

```
alignementVertical.css
?
.zone-principale {
  display: table;
  height: 400px;
  width: 30%;
  border: 3px solid ;
}
p{
  display: table-cell;
  vertical-align: middle;
  text-align: center;
}
```

Quelques explications :

`.zone-principale` notre zone principale comporte les caractéristiques suivantes :

`display: table;` permet de simuler le comportement d'une table (afin que plus tard l'utilisation des propriétés `display: table-cell;` et `vertical-align` soit possible)

`height: 400px;` permet de fixer une hauteur de 400 pixels

`width: 30%;` permet de fixer une largeur variable de 30 % par rapport à la largeur de l'écran

`border: 3px solid ;` permet de dessiner une bordure tout autour de la zone pour mieux la cerner à l'affichage

La balise `p` comporte les caractéristiques suivantes :

`display: table-cell;` permet de simuler le comportement d'une cellule afin que le `vertical-align` soit possible.

`vertical-align: middle;` permet d'aligner le texte verticalement (dans la hauteur).

`text-align: center;` permet de centrer le texte horizontalement (dans la largeur).

Résultat

Texte centré verticalement et horizontalement.

Il arrive souvent que l'on intègre une image et que l'on doive aligner du texte à côté, cela ne peut pas toujours être calculé de manière approximative si la taille du texte (ou de l'image) est amenée à changer.

Pour plus de compréhension, voici la problématique rencontrée :



Le Lorem Ipsum est simplement du faux texte employé dans la composition et la mise en page avant impression. Le Lorem Ipsum est le faux texte standard de l'imprimerie depuis les années 1500, quand un peintre anonyme assembla ensemble des morceaux de texte pour réaliser un livre spécimen de polices de texte.

Pour remédier à cette situation nous allons utiliser la propriété `display: table;` afin de simuler le comportement d'une table :

```
alignementTextelImage.html
```

```
<!Doctype html>
<html>
  <head>
    <meta charset="utf-8">
    <link rel="stylesheet" href="alignementTextelImage.css">
    <title>Ma Page</title>
  </head>
  <body>
    <p>
      <span></span>
      <span>
        Le Lorem Ipsum est simplement du faux texte employé dans la composition et la mise en page avant impression. Le Lorem Ipsum est le faux texte standard de l'imprimerie depuis les années 1500, quand un peintre anonyme assembla ensemble des morceaux de texte pour réaliser un livre spécimen de polices de texte.<br>
      </span>
    </p>
  </body>
</html>
```

Notre texte est contenu à l'intérieur d'une zone et voici le code css associé :

```
alignementTextelImage.css
```

```
p{ display:table; /* simule le comportement d'une table avec des colonnes */}
span{
display: table-cell;
vertical-align:middle;
}
img{
margin: 0 20px 0 0;
}
```

Quelques explications :

`p` contient les autres éléments et aura les caractéristiques suivantes :

`display: table;` permet de simuler le comportement d'une table (afin que plus tard l'utilisation des propriétés `display: table-cell;` et `vertical-align` soit possible)

`span` contient à la fois l'image et le texte avec les caractéristiques suivantes :

`display: table-cell;` permet de simuler le comportement d'une cellule afin que le `vertical-align` soit possible.
`vertical-align: middle;` permet d'aligner le texte verticalement (dans la hauteur).

`img` nous sélectionnons l'image uniquement pour l'écartez un peu du texte en lui apportant un peu de marge :

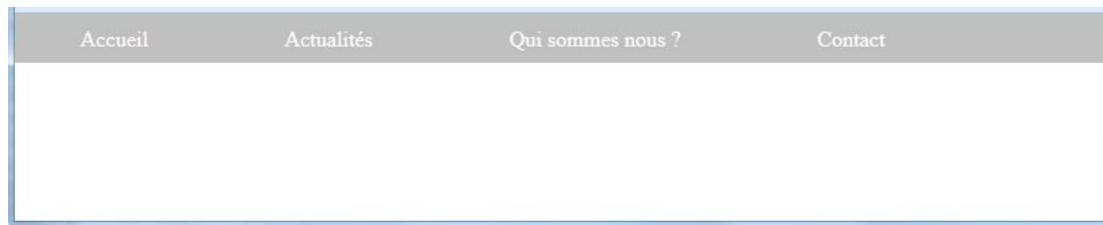
`margin: 0 20px 0 0;` permet d'appliquer de la marge à droite.

Résultat



Il arrive parfois que le nombre de liens dans un menu soit variable et changeant et pour cela il n'est pas toujours évident de calculer la bonne marge à appliquer à gauche comme à droite.

Problématique :



Les liens sont alignés à gauche de notre zone.

Pour cette raison, nous allons utiliser la propriété `display: inline` et `display: inline-block`;

```
menuCentre.html
?
<!Doctype html>
<html>
  <head>
    <meta charset="utf-8">
    <link rel="stylesheet" href="menuCentre.css">
    <title>Ma Page</title>
  </head>
  <body>
    <nav>
      <ul>
        <li><a href="#">Accueil</a></li>
        <li><a href="#">Actualités</a></li>
        <li><a href="#">Qui sommes nous ?</a></li>
        <li><a href="#">Contact</a></li>
      </ul>
    </nav>
  </body>
</html>
```

Le menu est déclaré via la balise `<nav>`, à l'intérieur se trouve une liste ` ` et enfin nos liens représentés par la balise `<a>`

```
menuCentre.css
?
body{
  margin: 0;
}
nav ul{
  padding: 0.5em 0;
  margin: 0;
  background : #c0c0c0;
  text-align: center;
}
nav li {
  display: inline;
}
nav a {
  margin: 0 3em;
  color: #ff0;
  text-decoration: none;
}
```

Quelques explications :

`body{ margin: 0; }` nous retirons la marge héritée de notre zone `body`

`nav ul` voici les caractéristiques de notre liste `ul` contenu dans la balise `nav` :

`padding: 0.5em 0;` permet d'ajouter de la marge intérieure à la fois en haut et en bas mais aucune sur les côtés.

`margin: 0;` nous retirons la marge héritée de notre zone `ul`

`text-align: center;` permet de centrer le texte (en l'occurrence, nos liens)

`nav li{ display: inline; }` permet de modifier le comportement de nos éléments de liste `li` afin qu'ils se comportent comme étant inline (pour se succéder les uns à côté des autres en prenant uniquement la place dont ils ont besoin) et non de type block (sur toute la largeur)

`nav a` permet de styliser les éléments `<a>` contenu dans notre balise de menu `<nav>`, voici les caractéristiques :

`margin: 0 3em;` permet d'espacer les liens entre eux.

`color: #fff;` permet d'appliquer une couleur blanche sur nos liens

`text-decoration: none;` permet de retirer l'effet souligné hérité par nos liens

Résultat



Les liens de menu sont bien centré à l'horizontal.

99

Comment faire des bordures arrondies ?

Pour faire des bordures arrondies sur une zone, une image ou tout autre élément qui s'y prête, vous pouvez utiliser la propriété CSS `border-radius` :

Voici un exemple :

```
bordureArrondie.html
```

```
<!Doctype html>
<html>
  <head>
    <meta charset="utf-8">
    <link rel="stylesheet" href="bordureArrondie.css">
    <title>Ma Page</title>
  </head>
  <body>
    <div class="mazone">...</div>
  </body>
</html>
```

```
bordureArrondie.css
```

```
div.mazone{
  border: 3px solid;
  border-radius: 10px;
  min-height: 100px;
  width: 50%;
}
```

Quelques explications :

`div` nous sélectionnons notre zone `div` ayant la classe `.mazone`

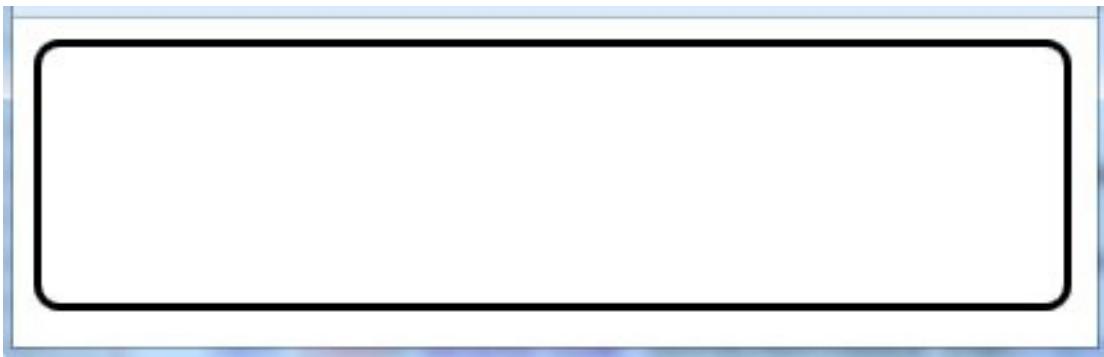
`border: 3px solid;` permet de tracer une bordure tout autour

`border-radius: 10px;` permet d'ajouter un effet arrondi de 10 pixels sur les bords en haut à gauche, en haut à droite, en bas à gauche et en bas à droite.

`min-height: 100px;` permet de fixer une hauteur minimum à 100 pixels

`width: 50%;` permet d'appliquer une largeur variable de 50% par rapport à la taille de l'écran.

Résultat



La bordure est bien arrondie.

100

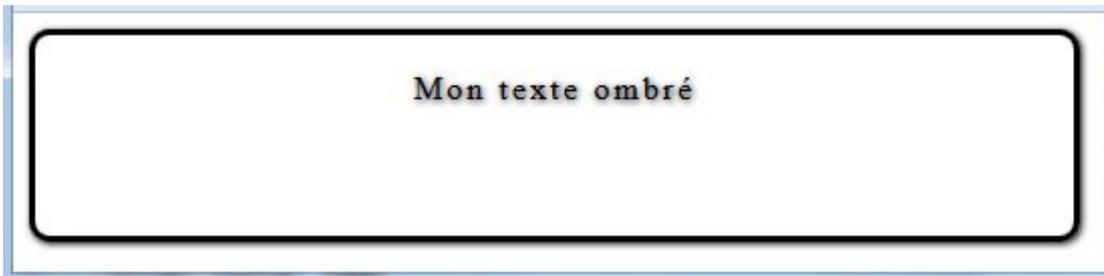
Comment mettre de l'ombre sur une image ou sur une zone ?

Pour mettre de l'ombre sur une image ou sur une zone, nous allons utiliser la propriété css `box-shadow`, voici un exemple

```
ombre.html
?
<!Doctype html>
<html>
  <head>
    <meta charset="utf-8">
    <link rel="stylesheet" href="ombre.css">
    <title>Ma Page</title>
  </head>
  <body>
    <div>
      <p>Mon texte ombré</p>
    </div>
  </body>
</html>
```

```
ombre.css
?
div {
  border: 3px solid;
  min-height: 100px;
  width: 50%;
  text-align: center;
  box-shadow: 1px 1px 3px;
}
p{
  letter-spacing: 2px;
  text-shadow: 1px 1px 3px;
}
```

Résultat



Dans ce résultat, l'ombre se trouve à la fois sur la zone (`box-shadow`) et sur le texte (`text-shadow`).

Quelques explications :

`div` nous sélectionnons notre zone `div`

`border: 3px solid;` permet de tracer une bordure tout autour

```
min-height: 100px; permet de fixer une hauteur minimum à 100 pixels
width: 50%; permet d'appliquer une largeur variable de 50% par rapport à la taille de l'écran.
text-align: center; permet d'aligner le texte au centre
box-shadow: 1px 1px 3px; permet d'appliquer de l'ombre tout autour de la zone
```

p permet de sélectionner notre paragraphe contenant notre texte
letter-spacing: 2px; permet l'espacement des lettres entre elles
text-shadow: 1px 1px 3px; permet d'appliquer de l'ombre sur le texte
box-shadow se décompose en plusieurs valeurs possibles :

Valeur	Description	Exemple
position horizontale (x)	permet d'afficher l'ombre sur la gauche ou la droite de l'élément.	Par exemple, 5px donneront une ombre sur la droite, -5px donneront une ombre sur la gauche
position verticale (y)	permet d'afficher l'ombre sur le haut ou le bas de l'élément.	Par exemple, 5px donneront une ombre sur la bas, -5px donneront une ombre sur le haut
Distance du flou (blur)	cela correspond à l'étendue du flou (dégrade la couleur de l'ombre vers le transparent).	Par exemple, 3px donneront une ombre légèrement étendue, et 8px donneront une ombre plus étendue
Taille de l'ombre (spread)	cela nous donne la possibilité de redimensionner l'ombre.	Taille adaptable selon le nombre de pixels
Couleur (color)	permet de gérer la couleur de l'ombre.	#000; donnera une ombre noir, tandis que #336699; donnera une ombre bleue
Intérieur (inset)	permet de gérer la position de l'ombre.	le mot clé inset vous donnera une ombre intérieure, tandis que ne rien préciser reviendra à faire une ombre portée vers l'extérieur (par défaut)

Voici quelques exemples avec **box-shadow** :

```
box-shadow: x y blur spread color;
```

Autre cas, si vous souhaitez obtenir de l'ombre uniquement en haut :

```
box-shadow: 0 -5px 5px -5px #111;
```

de l'ombre uniquement à droite :

```
box-shadow: 5px 0 5px -5px #111;
```

de l'ombre uniquement en bas :

```
box-shadow: 0 5px 5px -5px #111;
```

de l'ombre uniquement à gauche :

```
box-shadow: -5px 0 5px -5px #111;
```

de l'ombre partout :

```
box-shadow: 0 0 5px #333;
```

Il arrive souvent qu'un site ait besoin de visuel pour rendre le meilleur affichage possible (Un site sans image est rare).

Imaginons le cas d'une bannière de 2000 pixels :



Inscrivons le code Html suivant :

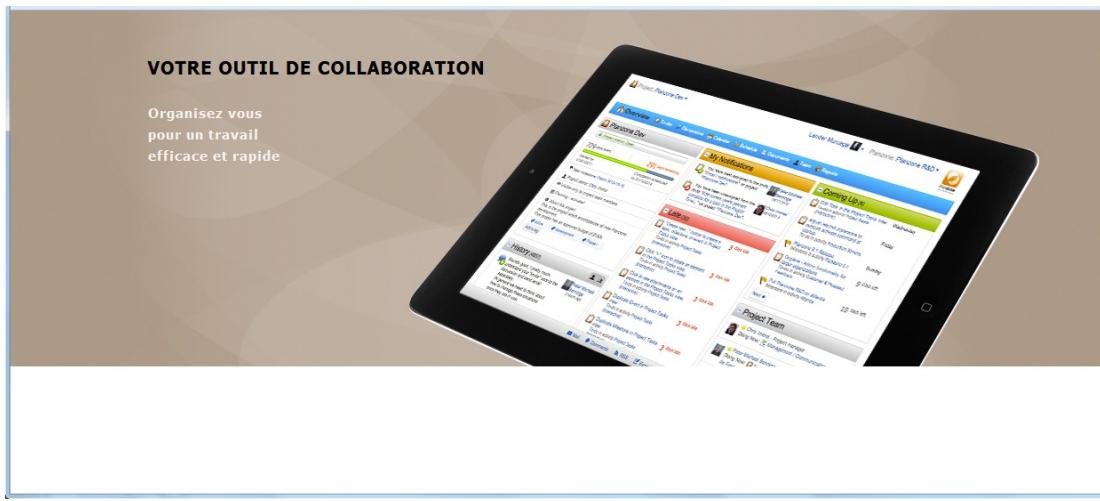
```
banniere.html

<!Doctype html>
<html>
  <head>
    <meta charset="utf-8">
    <link rel="stylesheet" href="banniere.css">
    <title>Ma Page</title>
  </head>
  <body>
    <div class="contenuBanniere">
      
    </div>
    <div class="conteneur">
      <div class="bandeauTitre">
        <h1>VOTRE OUTIL DE COLLABORATION</h1>
      </div>
      <div class="bandeauTexte">
        <h2>Organisez vous<br>
        pour un travail<br>
        efficace et rapide</h2>
      </div>
    </div>
  </body>
</html>
```

```
banniere.css

body{
  margin: 0;
  font-family: Tahoma;
  font-size: 10px;
  overflow-x: hidden; /* evite la scrollbar horizontal */
}
.contenuBanniere{
  position: relative; /* position relative pour pouvoir redéfinir son positionnement */
  left: 50%; /* je m'écarte de 50% de la gauche */
  margin-left: -1000px; /* 1000px de moins pour garder le contenu du fichier image centré (c'est la moitié car l'image fait 2000px) */
}
.conteneur{
  width: 960px;
  margin: 0 auto;
}
.bandeauTitre {
  color: #000000;
  font-weight: bold;
  left: 20px;
  letter-spacing: 1px;
  position: relative;
  top: -365px;
}
.bandeauTexte {
  color: #F6F4F1;
  left: 20px;
  letter-spacing: 1px;
  line-height: 24px;
  position: relative;
  top: -353px;
}
```

Résultat



Quelques explications :

body permet d'appeler toute la page, voici les propriétés précisées :

margin: 0; permet de retirer les marges héritées

font-family: Tahoma; permet de modifier la police d'écriture.

font-size: 10px; permet de modifier la taille du texte

overflow-x: hidden; évite une scrollbar horizontale (du fait de notre grande bannière, cette ligne est importante).

contenuBanniere permet d'appeler la zone contenant la bannière :

position: relative; permet de pouvoir redéfinir son positionnement

left: 50%; permet de nous écarter de la gauche de 50% (la moitié).

margin-left: -1000px; permet de créer un décalage de 1000 pixels (la moitié des 2000 pixels) afin de rester centré sur la page web

overflow-x: hidden; évite une scrollbar horizontale (du fait de notre grande bannière, cette ligne est importante).

conteneur permet de créer une "enveloppe" qui contiendra le reste des informations :

width: 960px; correspond à la largeur

margin: 0 auto; permet de centrer notre conteneur et tout ce qui s'y trouve à l'intérieur (0 marge en haut et en bas, automatique à gauche et à droite).

La bannière de 2000 pixels est bien centrée ! Le reste du code concerne l'affichage des textes.

102

Des effets sur nos images

Pour rendre vivant un site web il est recommandé de prévoir quelques effets visuels interactifs lorsque l'internaute survole certaines zones.

Voici un exemple Html :

```
effet1.html

<!Doctype html>
<html>
  <head>
    <meta charset="utf-8">
    <link rel="stylesheet" href="effet1.css" />
    <title>Ma Page</title>
  </head>
  <body>
    <figure>
```

```

<figcaption>
    <h2>
        <a href="">Ma Legende</a>
    </h2>
</figcaption>
</figure>
</body>
</html>
```

effet1.css

```
* { margin: 0; padding: 0; }

figure {
width: 300px;
position: relative;
overflow:hidden;
border: 3px solid #c0c0c0;
}
figure img{ display: block; }

figure figcaption a { text-decoration: none; color: #fff; }

figcaption {
position: absolute;
left: 0;
bottom: -30%;
background: rgba(0,0,0,0.75);
padding: 5px 0;
text-align: center;
width:300px;
transition: 0.6s;
}
figure:hover figcaption {
bottom: 0; /* lorsque l'on survol l'image, on fait remonter la légende */
}
```

Résultat



Au passage de notre souris au dessus de l'image, lors du survol, nous voyons une légende apparaître.

Quelques explications :

<figure> et <figcaption> sont mis en oeuvre conjointement pour attribuer une légende à une image, une vidéo ou tout autre élément média.

Figure permet de contenir des éléments médias (images ou autres), Figcaption permet de préciser une légende associée.

```
* { margin: 0; padding: 0; } permet de casser les héritages de marges intérieures et extérieures sur tous les éléments html.
```

figure permet de déclarer un bloc qui contiendra des éléments médias, voici ses propriétés :

```
width: 300px; permet d'appliquer une largeur de 300 pixels à notre zone figure  
position: relative; permet de redéfinir la position de la zone.  
overflow: hidden; évite une scrollbar.  
border: 3px solid #c0c0c0; permet d'appliquer une bordure tout autour.
```

figure img{ display: block; } permet de sélectionner l'image dans la figure pour lui appliquer la propriété **block** afin qu'elle prenne toute la place

```
figure figcaption a { text-decoration: none; color: #fff; } permet de sélectionner les liens présents à l'intérieur de figcaption pour leur appliquer une couleur blanche et éviter l'effet souligné (text-decoration: none)
```

figcaption permet de préciser une légende, voici le code associé :

```
position: absolute; permet de positionner précisément la légende (dans son parent qui est en position relative) sans prendre en compte l'affichage des éléments déjà présents  
left: 0; nous n'aurons aucun pixel de décalage avec le bord gauche  
bottom: -30%; nous descendons 30% plus bas (cela aura pour effet de masquer la légende)  
background: rgba(0,0,0,0.75); permet d'avoir un fond de couleur noir légèrement opaque (ce qui permettra de toujours voir l'image dessous)  
padding: 5px 0; permet d'appliquer de la marge intérieure en haut et en bas.  
text-align: center; permet d'aligner le texte de la légende au centre.  
width: 300px; permet d'appliquer une largeur de 300 pixels.  
transition: 0.6s; Cette transition permettra (au moment du survol hover) de ne pas afficher la légende d'un coup brutal mais progressivement.
```

```
figure:hover figcaption Lorsque le bloc figure sera survolé, nous irons chercher la balise figcaption pour lui appliquer la propriété suivante :  
bottom: 0; cela fera remonter la légende et annulera la propriété (bottom: -30%) que nous avions précédemment déclaré.
```

Création d'un site web

103

Préparatifs

En entreprise, pour créer un site web, nous aurons besoin d'une créa graphique à suivre.

La plupart du temps cela est fourni pas le designer au format PSD (pour photoshop) mais cela peut également être un fichier illustrator ou encore un mockup wireframe (de balsamik).

Pour l'exemple, nous allons décider de se créer un site CV, cela aura double emploi puisque d'une part vous apprendrez à réaliser un site web de A à Z et d'autre part vous pourriez pourquoi pas vous en reseervoir plus tard et ainsi vous démarquer lors d'entretiens d'embauche (pensez à l'améliorer en personnalisant les textes, changer les couleurs, etc. avant de le passer en ligne ;-)).

Nous pouvons également appeler cela un portfolio si vous ajoutez quelques-unes de vos réalisations / créations

Voici la créa graphique :

Nom Prenom
Communication / Marketing / Digital

Télécharger mon CV
Partagez

Accueil Mon CV Nos Projets Contact

Présentation

Vous êtes à la recherche d'une personne compétente et efficace pour gérer vos projets web ?
Actuellement, je suis ..., je fais ... actuellement, je suis ..., je fais ... actuellement, je suis ..., je fais ... actuellement, je suis ..., je fais ...

Graphisme **Intégration** **Développement** **Référencement**

Sûr en offrant à vos visiteurs une navigation et une ergonomie adaptées à leurs habitudes et leurs équipements.

Intégration responsive design compatible tout navigateur et respectueux du w3c.

Développement parfaitement fonctionnel et sécurisé.

Un maximum de trafic pour votre site web.

Qui suis-je ?
Description en quelques lignes ...
Description en quelques lignes ...
Description en quelques lignes ...

Expérience

- Expérience 1 ...
- Expérience 2 ...
- Expérience 3 ...

Contact

Votre email :
Votre message :

envoyer le message

Informations • Mentions Légales • Contact
© 2013 Conception et réalisation par Nom Prenom. Tous droits réservés.

READER

CONTENTEUR

HAUT-GAUCHE

HAUT-DROITE

NAV

SECTION

MILIEU-HAUT-GAUCHE

MILIEU-HAUT-DROITE

MILIEU-GAUCHE

MILIEU-DROITE

POOTE

CONTENEUR

Zoning : Pour le moment nos zones possèdent des couleurs et même si cela peut paraître "étrange", tous les sites se développent de cette manière dans le milieu

professionnel, nous mettons d'abord des couleurs de fond pour identifier les zones disponibles et ensuite nous travaillons à l'intérieur.

Et oui cela fait beaucoup de zones à prévoir, mais nous n'avons pas le choix si nous souhaitons être précis il est judicieux de prévoir un maximum de zones.



➤ Informations

Il ne faut pas hésiter à créer des zones pour encapsuler les éléments

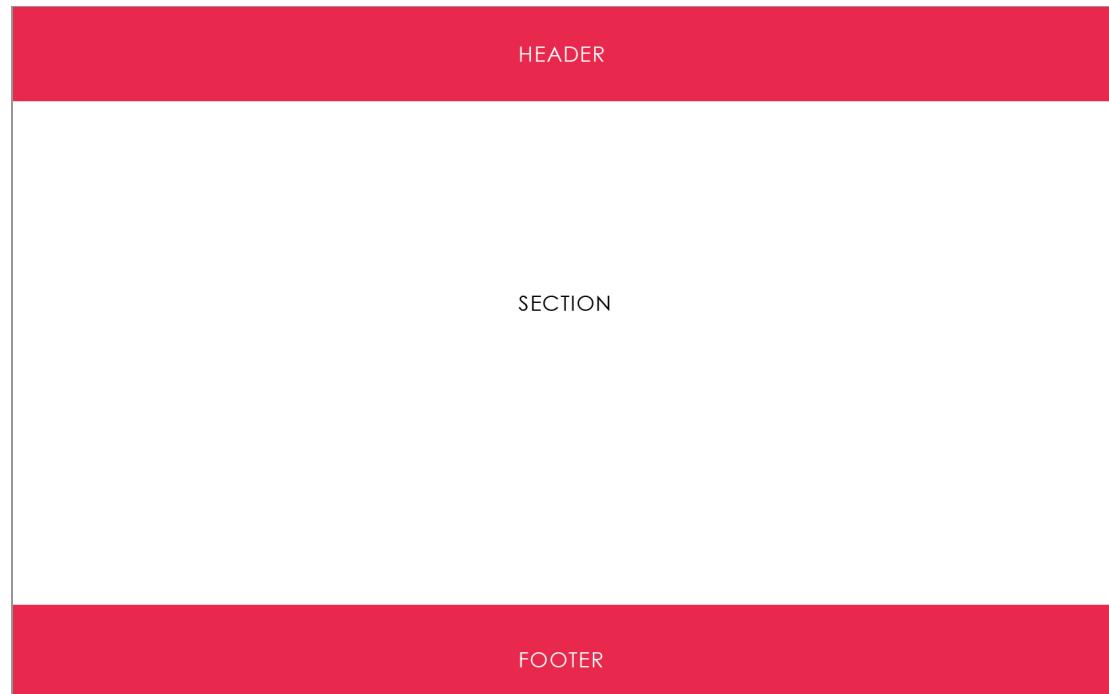
Bien entendu, ce zoning représente les zones majeures de la page web mais il y en aura d'autres à prévoir à l'intérieur (des sous zones ou zones secondaires).

Une fois que les texte, images, et autres informations sont intégrées dans ces zones. Nous retirons bien entendu les couleurs petit à petit (pour montrer que nous avons fini de travailler dans la zone en question).

Nos zones principales seront : **header**, **section**, **footer** et nous aurons systématiquement à l'intérieur l'appel de la <div class="conteneur"> pour centrer le contenu.

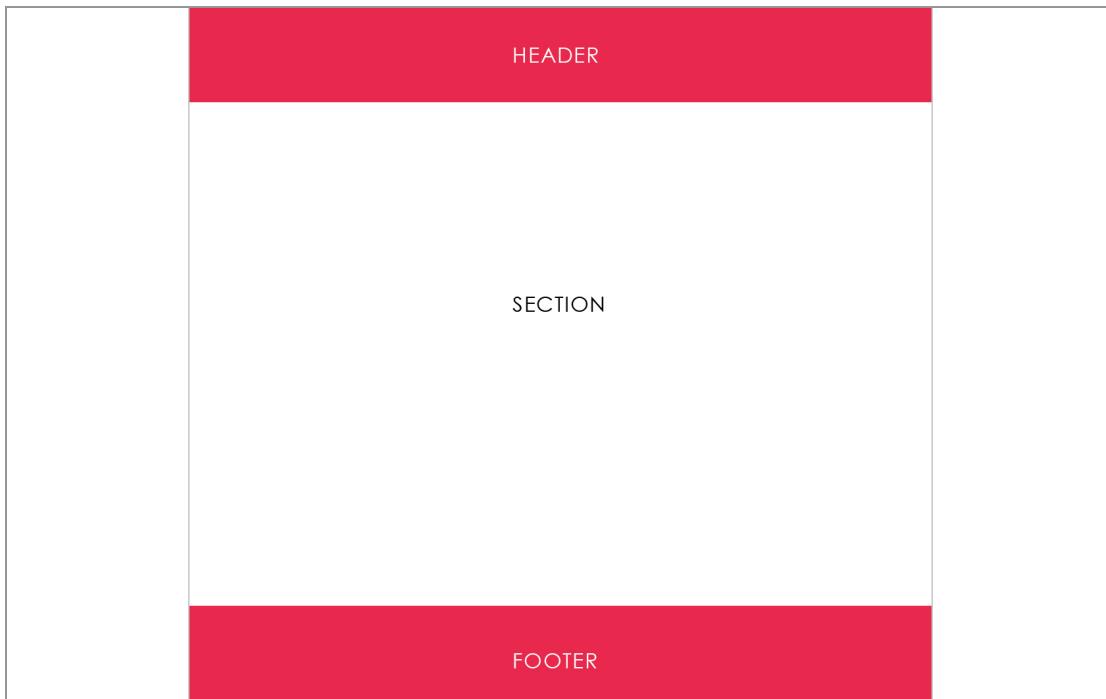
Nous partons sur un design de site pleine largeur (full width) et non pas avec un conteneur central qui englobe le tout. Nous n'aurons donc pas qu'un seul conteneur mais plusieurs.

Exemple d'une structure de site pleine largeur (notre cas dans le cadre de ce cours) :



Comme vous pouvez le constater, la couleur prend toute la largeur de la page web.

Exemple d'une structure de site avec un conteneur unique et central :



Créons notre arborescence, je vous propose l'architecture suivante :

```

📁 mon_site
  ---- 📁 css
  ----- 📄 style.css
  ---- 📁 js
  ---- 📁 img
  ----- 📄 index.html
  
```

104

Etape 1 - Créons nos premières lignes de code et notre structure !

index.html - etape 1 : structure du site web

```

<!Doctype html>
<html>
  <head>
    <meta charset="utf-8" >
    <title>Mon Site Web</title>
    <link rel="stylesheet" href="css/style.css">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <link href="http://fonts.googleapis.com/css?family=Roboto:400,900italic,900,700italic,700,500italic,500,400italic,300italic,300,100italic,100rel="stylesheet" type="text/css">
  </head>
  <body>
    <header>
      <div class="conteneur">
        ...
      </div>
    </header>
    <section>
      <div class="conteneur">
        ...
      </div>
    </section>
    <footer>
      <div class="conteneur">
        ...
      </div>
    </footer>
  </body>
</html>
  
```

Nous connaissons déjà la plupart de ces lignes de code (étudiées précédemment) : title pour un titre de page, link pour se relier à une feuille de style css, les balises meta pour l'encodage des caractères (utf-8) et initialiser l'affichage de l'écran.

Nous avons juste ajouté une ligne supplémentaire pour charger une google font.

Notre site web se divise en 3 tiers : header (haut), section (milieu), footer (bas).

```
style.css - etape 1 : structure du site web

***** Etape 1 : Structure *****
h1,h2,h3,p,a,ul,li{
margin: 0;
padding: 0;
text-decoration: none;
}
html, body {
margin: 0;
font-family: 'Roboto', sans-serif;
font-size: 0.9em;
}
header{
background: #F2F2F2;
padding: 0 0 0.3em 0;
margin: 0 0 2em 0;
border-bottom: 1px solid #c0c0c0;
}
section{ min-height: 700px; }
footer{ /* zone du bas */
background: #F2F2F2; /* fond */
border-top: 1px solid #c0c0c0; /* bordure en haut */
margin: 2em 0 0 0; /* marge du dessus */
text-align: center; /* centrer le texte */
padding: 20px 0; /* marge intérieur */
}
.conteneur{
max-width: 1170px;
margin: 0 auto;
border: 1px dotted;
}
```

Etape 1 CSS - Explications

Nous appelons plusieurs balises pour casser leur héritage (nous aurions également pu faire un reset CSS complet), par exemple nous ne souhaitons pas que les paragraphes <p> puissent avoir une marge imposée et décaler le reste de notre site. nous préférons dire qu'ils n'auront aucune marge pour gérer cela plus tard manuellement et au cas par cas.

La police "Roboto" est ajoutée sur la balise body et html afin de s'appliquer sur toute la page web.

Concernant la balise header, nous lui donnons une couleur de fond (gris), de l'espace intérieur (padding), de la marge en bas (margin, pour décoller header et section) et pour finir une fine bordure en dessous (border). Nous faisons pareil avec la balise footer.

La balise section ne contient qu'une propriété min-height (hauteur minimum) car nous savons que du contenu va rentrer et faire agrandir cette zone. Cela est facultatif mais permet de repousser le footer plus bas en attendant d'avoir du contenu.

La classe conteneur sera ré-utilisée à plusieurs endroits (c'est la raison pour laquelle nous l'avons déclarée sous forme de classe et non pas d'id) : La largeur maximum sera de 1170 pixels.

Pour obtenir l'effet centré, nous avons utilisé margin: 0 auto;

La bordure quant à elle n'est présente que pour la distinguer durant la phase de conception et les tests, nous la retirons à la fin de notre création.

Voici notre résultat (provisoire) :

Résultat (après l'étape 1)

105

Etape 2 - Remplissons notre structure avec des zones

index.html - etape 2 : remplissons notre structure

```
<!Doctype html>
<html>
  <head>
    <meta charset="utf-8" >
    <title>Mon Site Web</title>
    <link rel="stylesheet" href="css/style.css">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <link href="http://fonts.googleapis.com/css?family=Roboto:400,900italic,900,700italic,700,500italic,500,400italic,300italic,300,100italic,100" rel="stylesheet" type="text/css">
  </head>
  <body>
    <header>
      <div class="conteneur">
        <div class="haut-gauche">
          ...
        </div>
        <div class="haut-droite">
          ...
        </div>
        <div class="clear"></div>
        <nav>
          ...
        </nav>
      </div>
    </header>
    <section>
      <div class="conteneur">
        <div class="milieu-haut-gauche">
          ...
        </div>
        <div class="milieu-haut-droite">
          ...
        </div>
        <div class="clear"></div>
        <div class="milieu-gauche">
          ...
        </div>
        <div class="milieu-droite">
          ...
        </div>
        <div class="clear"></div>
      </div>
    </section>
    <footer>
      <div class="conteneur">
        ...
      </div>
    </footer>
  </body>
</html>
```

Etape 2 Html - Explication

Nous remplissons nos zones principales (header, section, footer) avec des zones secondaires (haut-gauche, haut-droite, milieu-haut-gauche, milieu-haut-droite, milieu-gauche, milieu-droite).

La classe `clear` est uniquement présente pour stopper l'effet flottant de certaines zones qui auront la propriété (`float: left;`).

style.css - etape 2 : remplissons notre structure

```
***** Etape 1 : Structure *****
h1,h2,h3,p,a,ul,li{
margin: 0;
padding: 0;
text-decoration: none;
}
html, body {
margin: 0;
font-family: 'Roboto', sans-serif;
font-size: 0.9em;
}
header{
background: #F2F2F2;
padding: 0 0 0.3em 0;
margin: 0 0 2em 0;
border-bottom: 1px solid #c0c0c0;
}
section{ min-height: 700px; }
footer/* zone du bas */
background: #F2F2F2; /* fond */
border-top: 1px solid #c0c0c0; /* bordure en haut */
margin: 2em 0 0 0; /* marge du dessus */
text-align: center; /* centrer le texte */
padding: 20px 0; /* marge intérieur */
}
.conteneur{
max-width: 1170px;
margin: 0 auto;
border: 1px dotted;
}

***** Etape 2 : Zones *****
.clear{
clear: both;
}
.haut-gauche{
float: left;
width: 70%;
background: #074eee;
}
.haut-droite{
float: right;
width: 29%;
height: 75px;
background: #bcb600;
}
.nav{
margin: 0;
text-align: center;
background: #fda500;
}
.milieu-haut-gauche{
float: left;
width: 30%;
background: #074eee;
}
.milieu-haut-droite{
float: left;
width: 70%;
font-size: 1.2em;
vertical-align: middle;
display: table;
line-height: 1.3em;
text-align: center;
height: 20vh;
background: #bcb600;
}
.milieu-gauche{
float: left;
width: 60%;
margin: 1.5em 0 1.5em 0;
background: #074eee;
}
.milieu-droite{
float: right;
width: 30%;
margin: 1.5em 0 1.5em 0;
background: #bcb600;
}
```

Etape 2 CSS - Explication

Nous prévoyons la classe CSS `clear` pour stopper les effets flottants `float: left;`

La zone haut-gauche se retrouve à côté de la zone haut-droite, idem pour milieu-haut-gauche avec milieu-haut-droite et les zones milieu-gauche avec milieu-droite.

Les largeurs (width) sont taillées en pourcentage % afin que les zones se redimensionnent automatiquement selon la taille de l'écran (responsive design fluide).

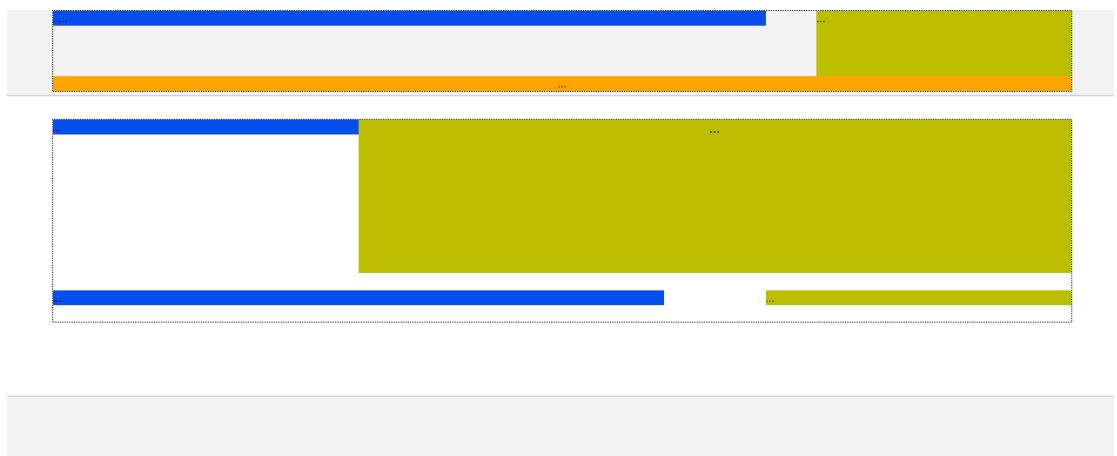
Les couleurs de fond (background) ne sont présentent uniquement dans la phase de construction du site web.

La zone `nav` prend toute la largeur et nous prévoyons avec la propriété `text-align: center;` que les rubriques soient centrées.

Pour la zone milieu-haut-droite, les propriétés `vertical-align: middle;` `display: table;` permettront au texte d'être centré verticalement.

Certaines zones possèdent un peu de marge pour s'écartier les unes des autres.

Voici notre résultat (provisoire) :



Résultat (après l'étape 2)

106

Etape 3 - Un peu de contenu en haut du site

index.html - étape 3 : Un peu de contenu en haut du site

```
<!Doctype html>
<html>
  <head>
    <meta charset="utf-8" >
    <title>Mon Site Web</title>
    <link rel="stylesheet" href="css/style.css">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <link href="http://fonts.googleapis.com/css?family=Roboto:400,900italic,900,700italic,700,500italic,500,400italic,300italic,300,100italic,100" rel="stylesheet" type="text/css">
  </head>
  <body>
    <header>
      <div class="conteneur">
        <div class="haut-gauche">
          <a href="index.html">
            <h1>Nom Prenom</h1>
          </a>
          <h2><span>C</span>ommunication / <span>M</span>arketing / <span>D</span>igital</h2>
        </div>
        <div class="haut-droite">
          <a href="#">Télécharger mon CV</a>
          <a href="#">Partagez</a>
        </div>
        <div class="clear"></div>
      </header>
      <nav>
        <ul>
          <li><a href="index.html">Accueil</a></li>
          <li><a href="moncv.html">Mon CV</a></li>
          <li><a href="mesprojets.html">Nos Projets</a></li>
          <li><a href="contact.php">Contact</a></li>
        </ul>
      </nav>
    </div>
  </header>
  <section>
    <div class="conteneur">
      <div class="milieu-haut-gauche">
        ...
      </div>
      <div class="milieu-haut-droite">
        ...
      </div>
      <div class="clear"></div>
      <div class="milieu-gauche">
        ...
      </div>
    </div>
  </section>

```

```

</div>
<div class="milieu-droite">
...
</div>
<div class="clear"></div>
</div>
</section>
<footer>
<div class="conteneur">
...
</div>
</footer>
</body>
</html>

```

Etape 3 Html - Explication

Nous remplissons le haut de site (balise header) avec des titres, textes et liens.

style.css - etape 3 : Un peu de contenu en haut du site

```

***** Etape 1 : Structure *****
h1,h2,h3,p,a,ul,li{
margin: 0;
padding: 0;
text-decoration: none;
}
html, body {
margin: 0;
font-family: 'Roboto', sans-serif;
font-size: 0.9em;
}
header{
background: #F2F2F2;
padding: 0 0 0.3em 0;
margin: 0 0 2em 0;
border-bottom: 1px solid #c0c0c0;
}
section{ min-height: 700px; }
footer{ /* zone du bas */
background: #F2F2F2; /* fond */
border-top: 1px solid #c0c0c0; /* bordure en haut */
margin: 2em 0 0 0; /* marge du dessus */
text-align: center; /* centrer le texte */
padding: 20px 0; /* marge intérieur */
}
.conteneur{
max-width: 1170px;
margin: 0 auto;
border: 1px dotted;
}
***** Etape 2 : Zones *****
.clear{
clear: both;
}
.haut-gauche{
float: left;
width: 70%;
/*background: #074eee;*/
}
.haut-droite{
float: right;
width: 25%;
height: 75px;
/*background: #bcbe00; */
}
.nav{
margin: 0;
text-align: center;
/* background: #fd4500; */
min-height: 30px;
}
.milieu-haut-gauche{
float: left;
width: 30%;
background: #074eee;
}
.milieu-haut-droite{
float: left;
width: 70%;
font-size: 1.2em;
vertical-align: middle;
display: table;
line-height: 1.3em;
text-align: center;
height: 20vh;
background: #bcbe00;
}
.milieu-gauche{
float: left;
width: 60%;
margin: 1.5em 0 1.5em 0;
background: #074eee;
}
.milieu-droite{
float: right;
width: 30%;

```

```

margin: 1.5em 0 1.5em 0;
background: #bcbe00;
}
***** Etape 3 : Contenu en haut du site *****/
.haut-gauche h{
color: #000000;
font-size: 1.7em;
margin: 10px 0 0 55px;
text-shadow: 1px 1px 1px;
}
.haut-gauche h{
margin: 5px 0 0 0;
font-size: 1.1em;
letter-spacing: 3px;
}
.haut-gauche h> span{
color: #8daab7;
}
.haut-droite a{
color: #8daab7;
display: block;
text-align: right;
/* border: 1px solid; */
}
.haut-droite a:first-child{
margin: 10px 0 0 0;
}
.haut-droite a:hover{
text-decoration: underline;
}
nav li {
display: inline;
list-style: none;
}
nav a{
display:inline-block;
line-height: 0px;
color: #000000;
padding: 0 5%;
}
nav a:hover{
color: #fff;
background: #8daab7;
border-radius: 10px;
transition: padding .2s;
}

```

Etape 3 CSS - Explication

Nous stylisons la balise h1 et h2 contenue dans la zone haut-gauche. Idem pour les liens contenus dans la zone haut-droite.

Nous mettons également en forme le menu avec un effet de transition lors du hover.

Voici notre résultat (provisoire) :



Résultat (après l'étape 3)

Notez que nous pouvons retirer les couleurs de fond des zones haut-gauche, haut-droite et nav car nous avons terminé notre haut de site

```

<!Doctype html>
<html>
<head>
    <meta charset="utf-8" >
    <title>Mon Site Web</title>
    <link rel="stylesheet" href="css/style.css">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <link href="http://fonts.googleapis.com/css?family=Roboto:400,900italic,900,700italic,700,500italic,500,400italic,300italic,300,100italic,100"
rel="stylesheet" type="text/css">
</head>
<body>
    <header>
        <div class="conteneur">
            <div class="haut-gauche">
                <a href="index.html">
                    <h1>Nom Prenom </h1>
                </a>
                <h2><span>C</span>ommunication / <span>M</span>arketing / <span>D</span>igital</h2>
            </div>
            <div class="haut-droite">
                <a href="">Télécharger mon CV</a>
                <a href="">Partagez</a>
            </div>
            <div class="clear"></div>
        <nav>
            <ul>
                <li><a href="index.html">Accueil</a></li>
                <li><a href="moncv.html">Mon CV</a></li>
                <li><a href="mesprojets.html">Nos Projets</a></li>
                <li><a href="contact.php">Contact</a></li>
            </ul>
        </nav>
    </div>
</header>
<section>
    <div class="conteneur">
        <div class="milieu-haut-gauche">
            
        </div>
        <div class="milieu-haut-droite">
            <p><span class="gras bleu">Vous êtes à la recherche d'un développeur pour gérer vos projets web ?</span><br><br>
            Je peux renforcer votre équipe autant sur la partie graphique que développement<br>
            Seriez-vous prêt à me donner ma chance ?<br></p>
        </div>
        <div class="clear"></div>
        <div class="milieu-gauche">
            <h2>Présentation</h2>
            <div class="milieu-gauche-profil">
                
            </div>
            <div class="milieu-droite-profil">
                <p>
                    Vous êtes à la recherche d'une personne compétente et efficace pour gérer vos projets web ?<br><br>
                    Actuellement, je suis ..., je fais ... actuellement, je suis ..., je fais ... actuellement, je suis ..., je
                    fais ... <br>
                </p>
            </div>
        </div>
        <div class="clear"></div>
        <div class="milieu-bas-competences">
            <div class="milieu-bas-competences-design">
                <h2>Graphisme</h2><br>
                <a href=""></a><br><br>
                <p>Séduire en offrant à vos visiteurs une navigation et une ergonomie adaptées à leurs habitudes et leurs équipements. </p>
            </div>
            <div class="milieu-bas-competences-integration">
                <h2>Intégration</h2><br>
                <a href=""></a><br><br>
                <p>Intégration responsive design compatible tout navigateur et respectueux du w3c. </p>
            </div>
            <div class="milieu-bas-competences-developpement">
                <h2>Développement</h2><br>
                <a href=""></a><br><br>
                <p>Développement parfaitement fonctionnel et sécurisé. </p>
            </div>
            <div class="milieu-bas-competences-referencement">
                <h2>Référencement</h2><br>
                <a href=""></a><br><br>
                <p>Un maximum de trafic pour votre site web. </p>
            </div>
            <div class="clear"></div>
        </div>
        <div class="milieu-droite">
            <div class="cadre">
                <h2>Qui suis-je ?</h2>
                <p>
                    Description en quelques lignes ... <br>
                    Description en quelques lignes ... <br>
                    Description en quelques lignes ... <br>
                </p>
            </div>
            <div class="cadre">
                <h2>Expérience</h2>
                <ul>
                    <li>Expérience 1 ... </li>
                    <li>Expérience 2 ... </li>
                    <li>Expérience 3 ... </li>
                </ul>
            </div>
        </div>
    </div>

```

```

<div class="cadre">
    <h2>Contact</h2>
    <form method="post">
        <label for="expediteur">Votre email :</label><br>
        <input type="email" name="expediteur" id="expediteur" title="votre email"><br>
        <label for="expediteur">Votre message :</label><br>
        <textarea id="message" name="message" title="votre message"></textarea><br>
        <input type="submit" value="Envoyer le message"><br>
    </form>
</div>
</div>
<div class="clear"></div>
</div>
</section>
<footer>
    <div class="conteneur">
        ...
    </div>
</footer>
</body>
</html>

```

Etape 4 Html - Explication

Nous remplissons le milieu de site (balise section) avec des titres, textes, images, formulaire et liens.

Voici un lien où vous pourrez vous procurer les images : [Télécharger les images](#)

style.css - etape 4 : Du contenu au milieu du site

```

***** Etape 1 : Structure *****
h1,h2,h2,p,a,ul,li{
margin: 0;
padding: 0;
text-decoration: none;
}
html, body {
margin: 0;
font-family: 'Roboto', sans-serif;
font-size: 0.9em;
}
header{
background: #F2F2F2;
padding: 0 0 0.3em 0;
margin: 0 0 2em 0;
border-bottom: 1px solid #c0c0c0;
}
section{ min-height: 700px; }
footer/* zone du bas */
background: #F2F2F2; /* fond */
border-top: 1px solid #c0c0c0; /* bordure en haut */
margin: 2em 0 0 0; /* marge du dessus */
text-align: center; /* centrer le texte */
padding: 20px 0; /* marge intérieur */
}
.conteneur{
max-width: 1170px;
margin: 0 auto;
border: 1px dotted;
}
***** Etape 2 : Zones *****
.clear{
clear: both;
}
.haut-gauche{
float: left;
width: 70%;
/*background: #074eee;*/
}
.haut-droite{
float: right;
width: 25%;
height: 75px;
/*background: #bcbe00; */
}
nav{
margin: 0;
text-align: center;
/* background: #fda500; */
min-height: 30px;
}
.milieu-haut-gauche{
float: left;
width: 30%;
background: #074eee;
}
.milieu-haut-droite{
float: left;
width: 70%;
font-size: 1.2em;
vertical-align: middle;
display: table;
line-height: 1.3em;
text-align: center;
height: 20vh;
background: #bcbe00;
}

```

```

}

.milieu-gauche{
float: left;
width: 60%;
margin: 1.5em 0 1.5em 0;
background: #074eee;
}
.milieu-droite{
float: right;
width: 30%;
margin: 1.5em 0 1.5em 0;
background: #bcbe00;
}
***** Etape 3 : Contenu en haut du site *****/
.haut-gauche h1{
color: #000000;
font-size: 1.7em;
margin: 10px 0 0 55px;
text-shadow: 1px 1px 1px;
}
.haut-gauche h2{
margin: 5px 0 0 0;
font-size: 1.1em;
letter-spacing: 3px;
}
.haut-gauche h2 span{
color: #8daab7;
}
.haut-droite a{
color: #8daab7;
display: block;
text-align: right;
/* border: 1px solid; */
}
.haut-droite a:first-child{
margin: 10px 0 0 0;
}
.haut-droite a:hover{
text-decoration: underline;
}
nav li {
display: inline;
list-style: none;
}
nav a{
display:inline-block;
line-height: 30px;
color: #000000;
padding: 0 5%;
}
nav a:hover{
color: #fffff;
background: #8daab7;
border-radius: 10px;
transition: padding .2s;
}
***** Etape 4 : Du contenu au milieu du site *****/
.bleu{ color: #8daab7; }
.gras{ font-weight: bold; }
.italic{ font-style: italic; }
.milieu-haut-gauche{
float: left;
width: 30%;
/* background: yellow; */
}
.milieu-haut-droite{
float: left;
width: 70%;
font-size: 1.2em;
vertical-align: middle;
display: table;
line-height: 1.3em;
text-align: center;
height: 20vh;
/* background: orange; */
}
.milieu-haut-droite p{
display: table-cell;
vertical-align: middle;
}
.milieu-gauche{
float: left;
width: 60%;
margin: 1.5em 0 1.5em 0;
/*background: #f12f65; */
}
.milieu-gauche h2, .milieu-bas h2 {
color: #8daab7;
padding: 0.5em;
border-bottom: 1px solid;
border-top: 1px solid;
}
.milieu-gauche-profil{ float: left; width: 15%; }
.milieu-droite-profil{ float: left; margin-left: 5%; width: 80%; padding: 1em 0 0 0; }
.milieu-bas-competences{ /*background: green;*/ margin: 3em 0 0 0; }
.milieu-bas-competences-design{ width: 21%; float: left; margin-right: 5%; /*background: yellow;*/ text-align: center; min-height: 300px; }
.milieu-bas-competences-integration{ width: 21%; float: left; margin-right: 5%; /*background: yellow;*/ text-align: center; min-height: 300px; }
.milieu-bas-competences-developpement{ width: 21%; float: left; margin-right: 5%; /*background: yellow;*/ text-align: center; min-height: 300px; }
.milieu-bas-competences-referencement{ width: 21%; float: left; /*background: yellow;*/ text-align: center; min-height: 300px; }
.milieu-bas-competences img{transition: 0.5s; }
.milieu-bas-competences img:hover{ transform: rotate( 360deg); }
.img-responsive{ width: 100%; }
.img-responsive50{ width: 60%; }
.img-responsive70{ width: 70%; }

```

```
.milieu-gauche ul li{
list-style-type: none;
background: url(img/fleche.jpg) left center no-repeat;
padding: 0 0 0 30px;
margin: 3% 0;
}
.milieu-droite{
float: right;
width: 30%;
margin: 1.5em 0 1.5em 0;
/* background: #3298ff; */
}
.milieu-droite .cadre{
border: 1px solid #00c1d0;
border-radius: 10px;
margin: 15px 0;
}
.milieu-droite .cadre:first-child{ margin: 0; }
.milieu-droite .cadre h2{
color: #333333;
font-size: 15pt;
text-align: center;
padding: 5px 10px;
background: #f2f3f0;
border-bottom: 1px solid #e0e0e0;
border-radius: 10px 10px 0 0;
}
.milieu-droite .cadre p{
padding: 10px;
}
.milieu-droite .cadre form{
padding: 10px;
}
.milieu-droite .cadre ul li{
margin: 10px 0 0 30px;
padding: 5px;
}
input[type="submit"]{
color: #ff0;
background-color: #Bdaab7;
margin: 1em 0 0 0;
border-color: #222;
display: inline-block;
padding: .5em 1em;
font-weight: 400;
line-height: 1.2;
text-align: center;
white-space: nowrap;
vertical-align: middle;
cursor: pointer;
border: 1px solid transparent;
border-radius: .2em;
text-decoration: none;
}
input[type="submit"]:hover{
background: #e0e0e0;
}
input { width: 100%; }
textarea{ resize: none; width: 100%; }
```

Etape 4 CSS - Explication

Nous stylisons tous les éléments dans la partie du milieu.

Voici notre résultat (provisoire) :

Nom Prenom
Communication / Marketing / Digital

Télécharger mon CV
Partagez

Accueil Mon CV Nos Projets Contact



Vous êtes à la recherche d'un développeur pour gérer vos projets web ?

Je peux renforcer votre équipe autant sur la partie graphique que développement
Seriez-vous prêt à me donner ma chance ?

Présentation



Vous êtes à la recherche d'une personne compétente et efficace pour gérer vos projets web ?
Actuellement, je suis ... je fais ... actuellement, je suis ... je fais ... actuellement, je suis ... je fais ...
... actuellement, je suis ..., je fais ...

Graphisme



Séduire en offrant à vos visiteurs une navigation et une ergonomie adaptées à leurs habitudes et leurs équipements.

Intégration



Intégration responsive design compatible tout navigateur et respectueux du w3c.

Développement



Développement parfaitement fonctionnel et sécurisé.

Référencement



Un maximum de traffic pour votre site web.

Qui suis-je ?

Description en quelques lignes ...
Description en quelques lignes ...
Description en quelques lignes ...

Expérience

- Expérience 1 ...
- Expérience 2 ...
- Expérience 3 ...

Contact

Votre email :

Votre message :

envoyer le message

Résultat (après l'étape 4)

108

Etape 5 - Du contenu dans le bas du site

index.html - etape 5 : Du contenu dans le bas du site

```
<!Doctype html>
<html>
  <head>
    <meta charset="utf-8" >
    <title>Mon Site Web</title>
    <link rel="stylesheet" href="css/style.css">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <link href="http://fonts.googleapis.com/css?family=Roboto:400,900italic,900,700italic,700,500italic,500,400italic,300italic,300,100italic,100rel='stylesheet' type="text/css">
  </head>
  <body>
    <header>
      <div class="conteneur">
        <div class="haut-gauche">
          <a href="index.html">
            <h1>Nom Prenom</h1>
          </a>
          <h2><span>C</span>ommunication / <span>M</span>arketing / <span>D</span>igital</h2>
        </div>
        <div class="haut-droite">
          <a href="#">Télécharger mon CV</a>
          <a href="#">Partagez</a>
        </div>
        <div class="clear"></div>
        <nav>
          <ul>
            <li><a href="index.html">Accueil</a></li>
            <li><a href="moncvhtml">Mon CV</a></li>
            <li><a href="mesprojets.html">Nos Projets</a></li>
            <li><a href="contact.php">Contact</a></li>
          </ul>
        </nav>
      </div>
    </header>
    <section>
      <div class="conteneur">
        <div class="milieu-haut-gauche">
          
        </div>
```

```

<div class="milieu-haut-droite">
    <p><span class="gras bleu">Vous êtes à la recherche d'un développeur pour gérer vos projets web ?</span><br><br>
    Je peux renforcer votre équipe autant sur la partie graphique que développement<br>
    Seriez-vous prêt à me donner ma chance ?<br></p>
</div>
<div class="clear"></div>
<div class="milieu-gauche">
    <h2>Présentation</h2>
    <div class="milieu-gauche-profil">
        
    </div>
    <div class="milieu-droite-profil">
        <p>
            Vous êtes à la recherche d'une personne compétente et efficace pour gérer vos projets web ?<br><br>
            Actuellement, je suis ..., je fais ... actuellement, je suis ..., je fais ... actuellement, je suis ..., je
            fais ... <br>
        </p>
    </div>
</div>
<div class="clear"></div>
<div class="milieu-bas-competences">
    <div class="milieu-bas-competences-design">
        <h2>Graphisme</h2><br>
        <a href=""></a><br><br>
        <p>Séduire en offrant à vos visiteurs une navigation et une ergonomie adaptées à leurs habitudes et leurs équipements. </p>
    </div>
    <div class="milieu-bas-competences-integration">
        <h2>Intégration</h2><br>
        <a href=""></a><br><br>
        <p>Intégration responsive design compatible tout navigateur et respectueux du w3c. </p>
    </div>
    <div class="milieu-bas-competences-developpement">
        <h2>Développement</h2><br>
        <a href=""></a><br><br>
        <p>Développement parfaitement fonctionnel et sécurisé. </p>
    </div>
    <div class="milieu-bas-competences-referencement">
        <h2>Référencement</h2><br>
        <a href=""></a><br><br>
        <p>Un maximum de trafic pour votre site web. </p>
    </div>
    <div class="clear"></div>
</div>
<div class="milieu-droite">
    <div class="cadre">
        <h2>Qui suis-je ?</h2>
        <p>
            Description en quelques lignes ... <br>
            Description en quelques lignes ... <br>
            Description en quelques lignes ... <br>
        </p>
    </div>
    <div class="cadre">
        <h2>Expérience</h2>
        <ul>
            <li>Experience 1 ... </li>
            <li>Experience 2 ... </li>
            <li>Experience 3 ... </li>
        </ul>
    </div>
    <div class="cadre">
        <h2>Contact</h2>
        <form method="post">
            <label for="expediteur">Votre email :</label><br>
            <input type="email" name="expediteur" id="expediteur" title="votre email"><br>
            <label for="expediteur">Votre message :</label><br>
            <textarea id="message" name="message" title="votre message"></textarea><br>
            <input type="submit" value="envoyer le message"><br>
        </form>
    </div>
    <div class="clear"></div>
</div>
</div>
</section>
</footer>
<div class="conteneur">
    <a href="">Informations </a> -
    <a href="">Mentions Légales </a> -
    <a href="contact.php">Contact</a>
    <br>
    <p>© 2015 Conception et réalisation par Nom Prenom. Tous droits réservés.</p>
</div>
</footer>
</body>
</html>

```

Etape 5 Html - Explication

Le footer contient quelques liens et une phrase de texte.

style.css - etape 5 : Du contenu au milieu du site

```

***** Etape 1 : Structure *****
h1,h2,h2,p,a,ul,li{
margin: 0;
padding: 0;
text-decoration: none;

```

```
}

html, body {
margin: 0;
font-family: 'Roboto', sans-serif;
font-size: 0.9em;
}
header{
background: #F2F2F2;
padding: 0 0 0.3em 0;
margin: 0 0 2em 0;
border-bottom: 1px solid #c0c0c0;
}
section{ min-height: 700px; }
footer/* zone du bas */
background: #F2F2F2; /* fond */
border-top: 1px solid #c0c0c0; /* bordure en haut */
margin: 2em 0 0 0; /* marge du dessus */
text-align: center; /* centrer le texte */
padding: 20px 0; /* marge intérieur */
}
.conteneur{
max-width: 1170px;
margin: 0 auto;
border: 1px dotted;
}
***** Etape 2 : Zones *****
.clear{
clear: both;
}
.haut-gauche{
float: left;
width: 70%;
/*background: #074eee;*/
}
.haut-droite{
float: right;
width: 25%;
height: 75px;
/*background: #bcbe00; */
}
nav{
margin: 0;
text-align: center;
/* background: #fda500; */
min-height: 30px;
}
.milieu-haut-gauche{
float: left;
width: 30%;
background: #074eee;
}
.milieu-haut-droite{
float: left;
width: 70%;
font-size: 1.2em;
vertical-align: middle;
display: table;
line-height: 1.3em;
text-align: center;
height: 20vh;
background: #bcbe00;
}
.milieu-gauche{
float: left;
width: 60%;
margin: 1.5em 0 1.5em 0;
background: #074eee;
}
.milieu-droite{
float: right;
width: 30%;
margin: 1.5em 0 1.5em 0;
background: #bcbe00;
}
***** Etape 3 : Contenu en haut du site *****
.haut-gauche .h1{
color: #000000;
font-size: 1.7em;
margin: 10px 0 0 55px;
text-shadow: 1px 1px 1px;
}
.haut-gauche .h2{
margin: 5px 0 0 0;
font-size: 1.1em;
letter-spacing: 3px;
}
.haut-gauche .h2 span{
color: #8daab7;
}
.haut-droite a{
color: #8daab7;
display: block;
text-align: right;
/* border: 1px solid; */
}
.haut-droite a:first-child{
margin: 10px 0 0 0;
}
.haut-droite a:hover{
text-decoration: underline;
}
nav li {
display: inline;
list-style: none;
}
```

```

}
nav a{
display:inline-block;
line-height: 30px;
color: #000000;
padding: 0 5%;
}
nav a:hover{
color: #ffff;
background: #8daab7;
border-radius: 10px;
transition: padding .2s;
}
***** Etape 4 : Du contenu au milieu du site *****/
.bleu{ color: #8daab7; }
.gras{ font-weight: bold; }
.italic{ font-style: italic; }
.milieu-haut-gauche{
float: left;
width: 30%;
/* background: yellow; */
}
.milieu-haut-droite{
float: left;
width: 70%;
font-size: 1.2em;
vertical-align: middle;
display: table;
line-height: 1.3em;
text-align: center;
height: 20vh;
/* background: orange; */
}
.milieu-haut-droite p{
display: table-cell;
vertical-align: middle;
}
.milieu-gauche{
float: left;
width: 60%;
margin: 1.5em 0 1.5em 0;
/*background: #f12f65; */
}
.milieu-gauche h2, .milieu-bas h2 {
color: #8daab7;
padding: 0.5em;
border-bottom: 1px solid;
border-top: 1px solid;
}
.milieu-gauche-profil{ float: left; width: 15%; }
.milieu-droite-profil{ float: left; margin-left: 5%; width: 80%; padding: 1em 0 0 0; }
.milieu-bas-competences{ /*background: green;*/ margin: 3em 0 0 0; }
.milieu-bas-competences-design{ width: 21%; float: left; margin-right: 5%; /*background: yellow;*/ text-align: center; min-height: 300px; }
.milieu-bas-competences-integration{ width: 21%; float: left; margin-right: 5%; /*background: yellow;*/ text-align: center; min-height: 300px; }
.milieu-bas-competences-developpement{ width: 21%; float: left; margin-right: 5%; /*background: yellow;*/ text-align: center; min-height: 300px; }
.milieu-bas-competences-referencement{ width: 21%; float: left; /*background: yellow;*/ text-align: center; min-height: 300px; }
.milieu-bas-competences img{transition: 0.5s; }
.milieu-bas-competences img:hover{ transform: rotate(-360deg); }
.img-responsive{ width: 100%; }
.img-responsive50{ width: 80%; }
.img-responsive70{ width: 70%; }
.milieu-gauche ul li{
list-style-type: none;
background: url(img/fleche.jpg) left center no-repeat;
padding: 0 0 30px;
margin: 3% 0;
}
.milieu-droite{
float: right;
width: 30%;
margin: 1.5em 0 1.5em 0;
/* background: #3298ff; */
}
.milieu-droite .cadre{
border: 1px solid #c0c0c0;
border-radius: 10px;
margin: 15px 0;
}
.milieu-droite .cadre:first-child{ margin: 0; }
.milieu-droite .cadre h2{
color: #333333;
font-size: 1.5px;
text-align: center;
padding: 5px 10px;
background: #f2f3f3;
border-bottom: 1px solid #c0c0c0;
border-radius: 10px 10px 0 0;
}
.milieu-droite .cadre p{
padding: 10px;
}
.milieu-droite .cadre form{
padding: 10px;
}
.milieu-droite .cadre ul li{
margin: 10px 0 0 30px;
padding: 5px;
}
input[type="submit"]{
color: #fff;
background-color: #8daab7;
margin: 1em 0 0 0;
border-color: #222;
display: inline-block;
}

```

```

padding: .5em 1em;
font-weight: 400;
line-height: 1.2;
text-align: center;
white-space: nowrap;
vertical-align: middle;
cursor: pointer;
border: 1px solid transparent;
border-radius: .2em;
text-decoration: none;
}
input[type="submit"]:hover{
background: #000;
}
input { width: 100%; }
textarea{ resize: none; width: 100%; }
***** Etape 5 : Du contenu en bas du site *****
footer{
background: #F2F2F2;
border-top: 1px solid #c0c0c0;
margin: 2em 0 0 0;
text-align: center;
padding: 20px 0;
}
footer a{
color: #8dab7;
}
footer a:hover{
text-decoration: underline;
}
footer p{
margin: 7px 0 0 0;
}

```

Etape 5 CSS - Explication

Nous stylisons tous les éléments dans la partie du bas.

Voici notre résultat :

The screenshot displays a resume website template. At the top, there's a header with the name "Nom Prenom" and a subtitle "Communication / Marketing / Digital". Below the header is a navigation bar with links for "Accueil", "Mon CV", "Nos Projets", and "Contact". To the right of the navigation, there are two buttons: "Télécharger mon CV" and "Partagez". The main content area features a graphic of three devices (laptop, tablet, smartphone) connected by arrows. To the right of the graphic is a text block: "Vous êtes à la recherche d'un développeur pour gérer vos projets web ? Je peux renforcer votre équipe autant sur la partie graphique que développement Seriez-vous prêt à me donner ma chance ?". Below this is a "Présentation" section with a profile picture placeholder and a text block: "Vous êtes à la recherche d'une personne compétente et efficace pour gérer vos projets web ? Actuellement, je suis ... je fais ... actuellement, je suis ... je fais ... actuellement, je suis ... je fais ... actuellement, je suis ... je fais ...". To the right of this is a "Qui suis-je ?" section with three lines of placeholder text. Further down are sections for "Graphisme", "Intégration", "Développement", and "Référencement", each with a corresponding icon and a brief description. At the bottom of the page is a footer with links for "Informations - Mentions Légales - Contact" and a copyright notice: "© 2013 Conception et réalisation par Nom Prenom. Tous droits réservés."

Résultat (après l'étape 5)

```

<!Doctype html>
<html>
<head>
</head>
<body>
<header>
<div class="conteneur">
<div class="haut-gauche">
<a href="index.html">
<h1>Nom Prenom</h1>
</a>
<h2><span>C</span>ommunication / <span>M</span>arketing / <span>D</span>igital</h2>
</div>
<div class="haut-droite">
<a href="">Télécharger mon CV</a>
<a href="">Partagez</a>
</div>
<div class="clear"></div>
<nav>
<ul>
<li><a href="index.html">Accueil</a></li>
<li><a href="moncv.html">Mon CV</a></li>
<li><a href="mesprojets.html">Nos Projets</a></li>
<li><a href="contact.php">Contact</a></li>
</ul>
</nav>
</div>
</header>
<section>
<div class="conteneur">
<div class="milieu-haut-gauche">

</div>
<div class="milieu-haut-droite">
<p><span class="gras bleu">Vous êtes à la recherche d'un développeur pour gérer vos projets web ?</span><br><br>Je peux renforcer votre équipe autant sur la partie graphique que développement<br>Seriez-vous prêt à me donner ma chance ?<br></p>
</div>
<div class="clear"></div>
<div class="milieu-gauche">
<h2>Présentation</h2>
<div class="milieu-gauche-profil">

</div>
<div class="milieu-droite-profil">
<p>Vous êtes à la recherche d'une personne compétente et efficace pour gérer vos projets web ?<br><br>Actuellement, je suis ..., je fais ... actuellement, je suis ..., je fais ... actuellement, je suis ..., je fais ...<br>
</p>
</div>
<div class="clear"></div>
<div class="milieu-bas-competences">
<div class="milieu-bas-competences-design">
<h2>Graphisme</h2><br>
<a href=""></a><br><br><p>Séduire en offrant à vos visiteurs une navigation et une ergonomie adaptées à leurs habitudes et leurs équipements. </p>
</div>
<div class="milieu-bas-competences-integration">
<h2>Intégration</h2><br>
<a href=""></a><br><br><p>Intégration responsive design compatible tout navigateur et respectueux du w3c. </p>
</div>
<div class="milieu-bas-competences-developpement">
<h2>Développement</h2><br>
<a href=""></a><br><br><p>Développement parfaitement fonctionnel et sécurisé. </p>
</div>
<div class="milieu-bas-competences-referencement">
<h2>Référencement</h2><br>
<a href=""></a><br><br><p>Un maximum de trafic pour votre site web. </p>
</div>
<div class="clear"></div>
</div>
<div class="milieu-droite">
<div class="cadre">
<h2>Qui suis-je ?</h2>
<p>
Description en quelques lignes ...
Description en quelques lignes ...
Description en quelques lignes ...
</p>
</div>
<div class="cadre">
<h2>Expérience</h2>
<ul>
<li>Expérience 1 ...
<li>Expérience 2 ...
<li>Expérience 3 ...
</ul>
</div>

```

```

</div>
<div class="cadre">
<h2>Contact</h2>
<form method="post">
  <label for="expediteur">Votre email :</label><br>
  <input type="email" name="expediteur" id="expediteur" title="votre email"><br>
  <label for="expediteur">Votre message :</label><br>
  <textarea id="message" name="message" title="votre message"></textarea><br>
  <input type="submit" value="envoyer le message"><br>
</form>
</div>
<div class="clear"></div>
</div>
</section>
</footer>
<div class="conteneur">
  <a href="">Informations </a> -
  <a href="">Mentions Légales </a> -
  <a href="contact.php">Contact</a>
  <br>
  <p>© 2015 Conception et réalisation par Nom Prenom. Tous droits réservés.</p>
</div>
</footer>
</body>
</html>

```

Etape 6 Html - Explication

Le code HTML ne change pas.

style.css - etape 6 : Du Responsive Design

```

***** Etape 1 : Structure *****
h1,h2,p,a,ul,li{
margin: 0;
padding: 0;
text-decoration: none;
}
html, body {
margin: 0;
font-family: 'Roboto', sans-serif;
font-size: 0.9em;
}
header{
background: #F2F2F2;
padding: 0 0 0.3em 0;
margin: 0 0 2em 0;
border-bottom: 1px solid #c0c0c0;
}
section{ min-height: 700px; }
footer{ /* zone du bas */
background: #F2F2F2; /* fond */
border-top: 1px solid #c0c0c0; /* bordure en haut */
margin: 2em 0 0 0; /* marge du dessus */
text-align: center; /* centrer le texte */
padding: 20px 0; /* marge intérieur */
}
.conteneur{
max-width: 1170px;
margin: 0 auto;
border: 1px dotted;
}
***** Etape 2 : Zones *****
.clear{
clear: both;
}
.haut-gauche{
float: left;
width: 70%;
/*background: #074eee;*/
}
.haut-droite{
float: right;
width: 25%;
height: 75px;
/*background: #bcbe00; */
}
.nav{
margin: 0;
text-align: center;
/* background: #fda500; */
min-height: 30px;
}
.milieu-haut-gauche{
float: left;
width: 30%;
background: #074eee;
}
.milieu-haut-droite{
float: left;
width: 70%;
font-size: 1.2em;
vertical-align: middle;
display: table;
line-height: 1.3em;
text-align: center;
}

```

```

height: 20vh;
background: #bcbe00;
}
.milieu-gauche{
float: left;
width: 60%;
margin: 1.5em 0 1.5em 0;
background: #074eee;
}
.milieu-droite{
float: right;
width: 30%;
margin: 1.5em 0 1.5em 0;
background: #bcbe00;
}
***** Etape 3 : Contenu en haut du site *****
.haut-gauche h1{
color: #000000;
font-size: 1.7em;
margin: 10px 0 0 55px;
text-shadow: 1px 1px 1px;
}
.haut-gauche h2{
margin: 5px 0 0 0;
font-size: 1.1em;
letter-spacing: 3px;
}
.haut-gauche h2 span{
color: #8daab7;
}
.haut-droite a{
color: #8daab7;
display: block;
text-align: right;
/* border: 1px solid; */
}
.haut-droite a:first-child{
margin: 10px 0 0 0;
}
.haut-droite a:hover{
text-decoration: underline;
}
nav li {
display: inline;
list-style: none;
}
nav a{
display:inline-block;
line-height: 30px;
color: #000000;
padding: 0 5%;
}
nav a:hover{
color: #fffff;
background: #8daab7;
border-radius: 10px;
transition: padding .2s;
}
***** Etape 4 : Du contenu au milieu du site *****
.bleu{ color: #8daab7; }
.gras{ font-weight: bold; }
.italic{ font-style: italic; }
.milieu-haut-gauche{
float: left;
width: 30%;
/* background: yellow; */
}
.milieu-haut-droite{
float: left;
width: 70%;
font-size: 1.2em;
vertical-align: middle;
display: table;
line-height: 1.3em;
text-align: center;
height: 20vh;
/* background: orange; */
}
.milieu-haut-droite p{
display: table-cell;
vertical-align: middle;
}
.milieu-gauche{
float: left;
width: 60%;
margin: 1.5em 0 1.5em 0;
/*background: #f12f65; */
}
.milieu-gauche h2, .milieu-bas h2 {
color: #8daab7;
padding: 0.5em;
border-bottom: 1px solid;
border-top: 1px solid;
}
.milieu-gauche-profil{ float: left; width: 15%; }
.milieu-droite-profil{ float: left; margin-left: 5%; width: 80%; padding: 1em 0 0 0; }
.milieu-bas-competences{ /*background: green;*/ margin: 3em 0 0 0; }
.milieu-bas-competences-design{ width: 21%; float: left; margin-right: 5%; /*background: yellow;*/ text-align: center; min-height: 300px; }
.milieu-bas-competences-integration{ width: 21%; float: left; margin-right: 5%; /*background: yellow;*/ text-align: center; min-height: 300px; }
.milieu-bas-competences-developpement{ width: 21%; float: left; margin-right: 5%; /*background: yellow;*/ text-align: center; min-height: 300px; }
.milieu-bas-competences-referencement{ width: 21%; float: left; /*background: yellow;*/ text-align: center; min-height: 300px; }
.milieu-bas-competences img{transition: 0.5s; }
.milieu-bas-competences img:hover{ transform: rotate(-360deg); }
.img-responsive{ width: 100%; }

```

```

.img-responsive50{ width: 80%; }
.img-responsive70{ width: 70%; }
.milieu-gauche ul li{
list-style-type: none;
background: url(img/fleche.jpg) left center no-repeat;
padding: 0 0 30px;
margin: 3% 0;
}
.milieu-droite{
float: right;
width: 30%;
margin: 1.5em 0 1.5em 0;
/* background: #3298ff; */
}
.milieu-droite .cadre{
border: 1px solid #c0c0c0;
border-radius: 10px;
margin: 15px 0;
}
.milieu-droite .cadre:first-child{ margin: 0; }
.milieu-droite .cadre h2{
color: #333333;
font-size: 15px;
text-align: center;
padding: 5px 10px;
background: #f2f3f3;
border-bottom: 1px solid #c0c0c0;
border-radius: 10px 10px 0 0;
}
.milieu-droite .cadre p{
padding: 10px;
}
.milieu-droite .cadre form{
padding: 10px;
}
.milieu-droite .cadre ul li{
margin: 10px 0 0 30px;
padding: 5px;
}
input[type="submit"]{
color: #fff;
background-color: #8daab7;
margin: 1em 0 0 0;
border-color: #222;
display: inline-block;
padding: .5em 1em;
font-weight: 400;
line-height: 1.2;
text-align: center;
white-space: nowrap;
vertical-align: middle;
cursor: pointer;
border: 1px solid transparent;
border-radius: .2em;
text-decoration: none;
}
input[type="submit"]:hover{
background: #000;
}
input { width: 100%; }
textarea{ resize: none; width: 100%; }
***** Etape 5 : Du contenu en bas du site *****/
footer{
background: #F2F2F2;
border-top: 1px solid #c0c0c0;
margin: 2em 0 0 0;
text-align: center;
padding: 20px 0;
}
footer a{
color: #8daab7;
}
footer a:hover{
text-decoration: underline;
}
footer p{
margin: 7px 0 0 0;
}
***** Etape 6 : Du responsive design *****/
@media (max-width: 768px) {
.haut-gauche{ width: 100%; margin-bottom: 1em;}
.haut-droite{ display: none; } /* on ne propose pas le dl de cv sur mobile */
nav a{ display: block; }
.milieu-gauche{ width: 100%; /* background: red; */ }
.milieu-droite{ width: 100%; /* background: green; */ }
.milieu-droite .cadre{ width: 100%; }
.milieu-haut-gauche, .milieu-haut-droite{ width: 100%; }
.milieu-bas-competences-design{ width: 40%; }
.milieu-bas-competences-integration{ width: 40%; }
.milieu-bas-competences-developpement{ width: 40%; }
.milieu-bas-competences-referencement{ width: 40%; }
}
@media (max-width: 992px) { }
@media (max-width: 1200px) { }

```

Etape 6 CSS - Explication

Pour les mobiles : Si l'internaute possède une résolution d'écran inférieur à 768 pixels, nous cacherons la zone en haut à droite et laisserons la zone en haut à gauche prendre 100% de la largeur.

De la même manière les liens du menu prendront toute la place nécessaire en pleine largeur.

Les zones du milieu seront également redimensionnées pour une meilleure qualité de navigation

Pour les tablettes et les ordinateurs portables : nous ne définissons aucune règle d'affichage en particulier.

Pour les ordinateurs de bureau : le style css par défaut s'applique.

Voici notre résultat final sur mobile :



Vous êtes à la recherche d'un développeur pour gérer vos projets web ?

Je peux renforcer votre équipe autant sur la partie graphique que développement
Seriez-vous prêt à me donner ma chance ?

Présentation



Vous êtes à la recherche d'une personne compétente et efficace pour gérer vos projets web ?

Actuellement, je suis ..., je fais ... actuellement, je suis ..., je fais ... actuellement, je suis ..., je fais ... actuellement, je suis ..., je fais ...

Graphisme



Séduire en offrant à vos visiteurs une navigation et une ergonomie adaptées à leurs habitudes et leurs

Intégration



Intégration responsive design compatible tout navigateur et respectueux du w3c.

équipements.

Développement



Développement
parfaitement
fonctionnel et sécurisé.

Référencement



Un maximum de traffic
pour votre site web.

Qui suis-je ?

Description en quelques lignes ...
Description en quelques lignes ...
Description en quelques lignes ...

Expérience

- Expérience 1 ...
- Expérience 2 ...
- Expérience 3 ...

Contact

Votre email :

Votre message :

envoyer le message

[Informations](#) - [Mentions Légales](#) - [Contact](#)

© 2015 Conception et réalisation par Nom Prenom. Tous droits réservés.

Résultat (après l'étape 6, en responsive design)

Pour arriver à ce résultat, il est nécessaire de redimensionner la fenêtre pour simuler un petit écran.



Jump To: Notes and Potential Issues Congratulations - Icons

This document was successfully checked as HTML5!

Result: Passed, 2 warning(s)

Source :

```
<!Doctype html>
<html>
  <head>
    <meta charset="utf-8" >
    <title>Mon Site Web</title>
    <link rel="stylesheet" href="css/style.css">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <link href="http://fonts.googleapis.com/css?family=Roboto:400,900italic,900,700italic,700,500italic,500,400italic,300italic,300,100italic,100" rel="stylesheet" type="text/css">
  </head>
  <body>
    <header>
```

Encoding :

utf-8

(detect automatically)

Doctype :

HTML5

(detect automatically)

Root Element:

html

C'est bon signe ! en revanche pensez quand même à le tester sur différents navigateurs : Google Chrome, Firefox, Internet Explorer, Safari, Opéra.

111

Etape 8 - Créer d'autres pages

Comme vous l'avez vu les liens ne pointent nulle part en particulier. N'hésitez pas à créer les autres pages correspondantes et à améliorer ce site !

Par exemple, pour la page moncv.html, voici la strucutre à garder :

```
moncv.html - nouvelle page

<!Doctype html>
<html>
  <head>
    <meta charset="utf-8" >
    <title>Mon Site Web</title>
    <link rel="stylesheet" href="css/style.css">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <link href="http://fonts.googleapis.com/css?family=Roboto:400,900italic,900,700italic,700,500italic,500,400italic,300italic,300,100italic,100" rel="stylesheet" type="text/css">
  </head>
  <body>
    <header>
      <div class="conteneur">
        <div class="haut-gauche">
          <a href="index.html">
            <h1>Nom Prénom</h1>
          </a>
          <h2><span>C</span>ommunication / <span>M</span>arketing / <span>D</span>igital</h2>
        </div>
        <div class="haut-droite">
          <a href="#">Télécharger mon CV</a>
          <a href="#">Partagez</a>
        </div>
        <div class="clear"></div>
      </div>
      <nav>
        <ul>
          <li><a href="index.html">Accueil</a></li>
          <li><a href="moncv.html">Mon CV</a></li>
          <li><a href="mesprojets.html">Nos Projets</a></li>
          <li><a href="contact.php">Contact</a></li>
        </ul>
      </nav>
    </div>
    </header>
    <section>
      <div class="conteneur">
        </div>
      </div>
    </section>
    <footer>
      <div class="conteneur">
        <a href="#">Informations</a> -
      </div>
    </div>
  </body>
</html>
```

```
<a href="">Mentions Légales</a> -  
<a href="contact.php">Contact</a>  
<br>  
<p>© 2015 Conception et réalisation par Nom Prenom. Tous droits réservés.</p>  
</div>  
</footer>  
</body>  
</html>
```

Et bien entendu, le code CSS ne change pas, ce sera à vous d'ajouter les lignes en fonction du code Html que vous développerez.

112

Etape 9 - Hébergement & Mise en ligne

Après avoir développé plusieurs pages supplémentaires et vous être assuré que l'affichage correspondait à vos attentes ainsi que compatible (sur chacun des navigateurs couramment utilisés par vos internautes), vous pourriez éventuellement mettre ce site en ligne.

113

Etape 10 - Partagez votre lien de site web et améliorer votre référencement

Un site c'est bien, mais vous pouvez faire le meilleur site du monde, s'il n'est pas visible vous n'aurez pas de visite.

114

Créer son site avec l'aide d'un thème graphique

Si malgré toutes les explications et vos efforts, la création d'un site web vous paraît fastidieuse, vous pouvez toujours vous rabattre sur l'utilisation d'un thème graphique.

Il s'agit de thème graphique déjà écrit et prêt à l'utilisation (parfois gratuit, parfois payant).

Toutefois les clients ne sont pas toujours d'accord pour avoir le « design de tout le monde » .

Quoi qu'il en soit, il est important de connaître les langages Html et Css car vous aurez sans doute besoin d'apporter des modifications.

Plusieurs sites regroupant des thèmes pré-conçus existent dont [Thème Forest](#)

➤ Découvrez le Responsive Design

© Copyright Aucune reproduction, même partielle (textes, documents, images, etc.), ne peut être faite sans l'accord de son auteur.

Liam TARDIEU

EVOGUE.fr - EPROJET.fr

