

Sommaire

☒ Présentation	3	☒ Propriétés et Valeurs	20
Introduction		3	Mise en forme du texte et des titres – police, taille		20
Les enjeux et les avantages.....		3	Alignment		22
☒ Historique.....	6	Dimensions		22
Origine		6	Couleur et Fond		23
Developpements.....		6	Marges exterieures.....		24
Css 1.0		6	Marges Intérieures		25
Css 2.0		7	Bordures		26
Css 2.1		7	Positionnement		27
Css 3		7	Affichage		28
☒ Feuilles de style.....	8	Les Tableaux		29
Lier le langage css avec le langage html.....		8	Les Listes		30
Differences : <link /> ou @import ?		10	Effets		31
Ecrire du code css		11	Autres Propriétés		32
Les commentaires.....		11	L'impression		34
☒ Syntaxe	12	Pseudo-éléments		35
Fonctionnement		12	L'attribut Media		36
☒ Sélecteurs	Erreurs ! Signet non défini.	☒ Héritage	37
Différents types de sélecteurs		13	Le modèle Parent-Enfant(s)		37
Sélecteur de balises		13	Héritage d'origine		38
Sélecteur de classe.....		14	Deshéritage		38
Sélecteur d'identificateur		15	☒ Mise en page et positionnement	40
Sélecteur Imbrqué et Hierarchie		16	Type BLOC et Type INLINE		40
Sélecteur sous classe		16	La position absolue		41
Sélecteur et regroupement.....		17	La position relative		42
Sélecteur d'attributs		18	La position fixe		43
Sélecteur Pseudo-Element.....		18	Les flottants		44
			Comment centrer son site ?		47
			Les largeurs en pourcentage		47
			Hierarchie : Ancêtre, parents, enfants, frères		48
☒ Mise en forme		☒ Résolution d'écran	49
Introduction			Conception d'un site Internet		50
De la créa graphique à l'intégration			De la créa graphique à l'intégration		50
☒ Adapter son site pour les smart phones	51	☒ Quelques conseils	52
Introduction		51	Des outils indispensables : webdevelopper et FireBug		52
☒ Débogage 52			
Quelques conseils					
Des outils indispensables : webdevelopper et FireBug					

INTRODUCTION

Le langage CSS (Cascading Style Sheets ou feuilles de style en cascade) sert à décrire la présentation des documents HTML et XML.

Dans le domaine du web, le langage CSS (Cascading Style Sheets) est utilisé pour définir l'aspect futur de votre site, comme par exemple la couleur du fond de la page, le type de police ou encore le placement des différentes zones et du texte.

Les standards définissant CSS sont publiés par le World Wide Web Consortium (W3C).

Les feuilles de style ont été mises au point afin de compenser les manques du langage HTML en ce qui concerne la mise en page et la présentation.

En effet, le HTML offre un certain nombre de balises permettant de mettre en page et de définir le style d'un texte, toutefois chaque élément possède son propre style, indépendamment des éléments qui l'entourent.

Grâce aux feuilles de style, lorsque la charte graphique d'un site composé de plusieurs centaines de pages web doit être changée, il suffit de modifier la définition des feuilles de style en un seul endroit pour changer l'apparence du site tout entier.

LES ENJEUX ET LES AVANTAGES

L'un des objectifs majeurs de CSS est de permettre la mise en forme hors des documents.

Il est par exemple possible et vivement conseillé de ne décrire que la structure d'un document en HTML, et de décrire toute la présentation dans une feuille de style CSS séparée.

Voici plusieurs avantages liés à l'utilisation des feuilles de styles CSS :

- o Obtenir une présentation homogène sur tout un site en faisant appel sur toutes les pages à une même définition de style ; aisément modifiable.
- o Permettre le changement de l'aspect d'un site complet entier par la seule modification de quelques lignes.
- o Possibilité de doter une page de présentations alternatives au choix de l'utilisateur en appliquant différents styles à un même contenu.

Présentation



- o Une plus grande lisibilité du HTML, car les styles sont définis à part. Le code HTML est réduit en taille et en complexité. La structure et la présentation sont donc gérées séparément.
 - o Allègement des coûts de développement et de maintenance.
 - 8 Des chargements de page plus rapides (allègement du code source des pages Web, et donc économie de bande passante).
- Un positionnement plus rigoureux des éléments.
- o Possibilité de doter une page de feuilles de style spécifiques selon les médias (navigateurs graphiques, lecteurs d'écran...).
 - o Création de mise en page flexible s'adaptant aux différentes résolutions d'écran ;
 - o Meilleure accessibilité des pages Web par la disparition des tableaux détournés à des fins de présentation.

Je possède un site de 15 pages comportant 15 fichiers ayant l'extension HTML
Exemple Je souhaite modifier le fond de mes pages en couleur bleue
Avec un site de 15 pages, je vais devoir répéter 15 fois la ligne suivante :

```
<body style = "background: #336699">
```

En CSS, j'inscrirais dans mon fichier « style.css » le code suivant :

```
body { background: #336699; }
```

Si un jour je souhaite modifier la couleur de mes pages, avec l'utilisation du langage CSS, je n'aurais qu'un seul fichier à modifier.

De même, en intervenant sur le fichier définissant les styles, nous allons accélérer la maintenance : pour changer une couleur de titre, nous n'avons plus besoin de retoucher 50 articles un par un.

Nous changeons seulement, en une seule fois, la définition du style "titre".

D'où une maintenance plus simple, et la possibilité de "designer" un site.

A la différence des méthodes employées dans le début des années 1990, les CSS permettent une stricte séparation du contenu HTML et des informations de mise en page.

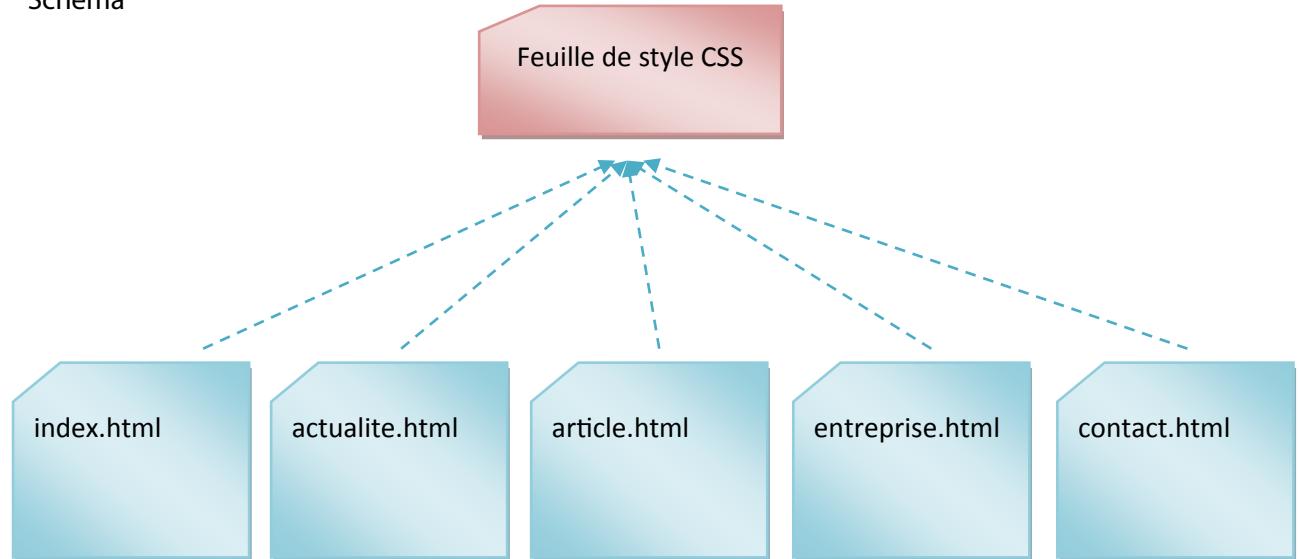
Le gain réalisé est considérable.

Le site ZenGarden réalisé en 2003, propose de tester la puissance des feuilles de style en permettant de choisir différents styles à appliquer pour un même contenu.

Il s'agit d'une démonstration de la possibilité de modifier librement le rendu affiché d'une même page web, uniquement grâce à CSS et sans aucune modification de son code HTML.

Avec CSS, nous pouvons modifier de façon dynamique l'apparence d'un site, bien plus puissamment qu'avec un code HTML mélangeant contenu et mise en forme.

Schéma



Ce schéma représente différentes pages HTML s'appuyant sur une feuille de style CSS.

Historique

ORIGINE

Les origines des CSS sont liées à trois questions majeures :

- o HTML est-il un format de structuration ou un format mixte de mise en forme et de structuration ?
- o Les choix de présentation de l'auteur doivent-ils avoir la primauté sur ceux de l'utilisateur ?
- o La réponse aux besoins est-elle un format de description de la mise en forme, ou un langage de transformation ?

Groupe de travail et réflexion

Le groupe de travail au sein du W3C, présidé par Chris Lilley comporte notamment des représentants d'Apple, Google, IBM, Microsoft, Adobe, de la Fondation Mozilla et d'Opera.

DEVELOPPEMENTS

Les spécifications CSS ne sont pas développées par versions successives, qui permettraient à un navigateur de se référer à une version en particulier

CSS est développé par « niveaux », ce qui contraint chaque nouveau niveau à intégrer le précédent, et chaque implémentation à être compatible avec la précédente : CSS1 est donc développé pour être un sous-ensemble de CSS2, qui est lui-même développé pour être un sous-ensemble de CSS3.

CSS 1.0

La spécification CSS1 finale est publiée le 17 décembre 1996, et définit une cinquantaine de propriétés.

CSS1 se définit comme un « mécanisme de feuilles de styles simples, permettant aux auteurs et aux lecteurs d'attacher des styles au document HTML ».

CSS devient couramment utilisé dans la conception de sites web et bien pris en charge par les navigateurs web dans les années 2000.

Le premier navigateur à avoir intégré les feuilles de style est Microsoft Internet Explorer 3.0 puis l'utilisation des feuilles de styles s'est généralisée.

Suite

CSS 2.0

Publié comme une recommandation en mai 1998, le second niveau de CSS étend considérablement les possibilités théoriques des feuilles de styles en cascade, avec en particulier environ 70 propriétés supplémentaires.

Depuis le 12 mai 1998, la norme CSS 2.0 est donc le standard en vigueur.

La norme CSS2 apporte un grand nombre de nouvelles fonctionnalités par rapport à la précédente norme (1.0), parmi lesquelles :

- o La prise en compte de la diversité de médias (écran, imprimante, braille, synthétiseur vocal, etc.).
- o L'ajout de propriétés auditives pour les aveugles et malvoyants.
- o La manipulation de curseurs.
- o La gestion des débordements et de la visibilité des éléments.

CSS 2.1

Les retours d'implémentation de CSS2 conduisent le groupe de travail CSS du W3C à rédiger à partir de 2001 une version révisée CSS 2.1 (« CSS Level 2, Revision 1 »), sur la base de ce qui était effectivement adopté par les différents navigateurs. CSS 2.1 corrige CSS2 sur de nombreux points de détail.

CSS 3

Le développement du troisième niveau des feuilles de styles en cascade commence dès 1999, parallèlement à celui de CSS 2.1.

CSS3 devient « modulaire », afin de faciliter ses mises à jour, mais aussi son implémentation par des agents utilisateurs aux capacités et aux besoins de plus en plus variés : navigateurs graphiques, navigateurs pour mobiles, navigateurs vocaux.

Feuille de style

LIER LE LANGAGE CSS AVEC LE LANGAGE HTML

Afin de lier du code CSS avec du code HTML ; Plusieurs méthodes s'offrent à vous.

Les voici dans l'ordre de la moins bonne à la meilleure :

Méthode 1 – CSS dans un attribut HTML

Directement dans le code d'une balise HTML en passant par l'attribut « style » :

```
<p style=".....Code Css ici.....">le texte</p>
```

Conclusion

Non seulement le CSS sera peu lisible à l'intérieur des balises, mais en plus cela ne nous permet pas de profiter de tous les avantages du CSS, comme par exemple la possibilité de changer la couleur de tous les titres d'un site en un seul coup.

Méthode 2 – CSS dans une balise <style>

Dans la partie « head » de la page entre les balises « style » :

```
<!DOCTYPE html>
<html>
  <head>
    <title>Titre</title>
    <style type="text/css">
.....Code CSS ici.....
    </style>
  </head>
  <body>
    ...
  </body>
</html>
```

Conclusion

Cette méthode n'est pas plus recommandée que la première étant donné que le code CSS inscrit entre les balises « style » sera local à la page et ne pourra pas servir dans les autres pages du site.

Méthode 3 – CSS avec @import

Dans un fichier externe, nommée « style.css ».

Ce ou ces fichiers seront appelés dans l'en-tête du fichier HTML (toujours dans la partie HEAD) sous cette forme :

```
<!DOCTYPE html>
<html >
  <head>
    <title>Titre</title>

    <style type="text/css">
      <!--
        @import url(../../styles/styles.css);
        @import url(page.css);
      -- >
    </style>
    </head>
    <body>
    ...
    </body>
  </html>
```

Conclusion

Cette méthode est intéressante car elle permet de décentraliser la mise en forme de la structure de la page.

Elle appelle effectivement une feuille de style (CSS) externe au fichier HTML.

Suite

Méthode 4 – CSS avec la balise LINK

En ajoutant une ligne entre les balises «head » qui va permettre d'indiquer au fichier HTML quel fichier CSS il se doit de charger :

```
<!D OCTYPE html >
<html >
  <head>
    <title>Titre</title>
    <link rel="stylesheet" media="screen" type="text/css" href="style.css" />
  </head>
  <body>
...
  </body>
</html>
```

Conclusion

Comme la précédente, cette méthode est intéressante car elle permet également de décentraliser la mise en forme de la structure de la page.

Elle appelle effectivement une feuille de style (CSS) externe au fichier HTML.

C'est LA méthode recommandée et utilisée sur la plupart des sites internet.

DIFFÉRENCES : <LINK /> OU @IMPORT ?

Une question fréquemment posée : vaut-il mieux lier sa feuille de style avec l'élément link ou utiliser la règle @import entre des balises styles ?

L'élément HTML link

link est un élément HTML qui n'est pas uniquement prévu pour lier des feuilles de styles.

Nous pouvons également l'utiliser pour faire appel à une ressource externe comme par exemple un flux RSS ou une favicon, etc.

Dans tous les cas, un élément link présente au moins deux attributs : rel (type de relation) et href (ressource liée).

La règle CSS @import

@import est une propriété CSS2 qui doit être suivie de l'URL d'une feuille de styles à importer.

Nous pouvons utiliser @import:

1. entre les balises <style> et <style> dans le head d'une page HTML ;
2. au début d'un fichier .css, pour inclure une ou plusieurs autres feuilles de styles ;

Suite

Cette deuxième possibilité est intéressante car elle permet de créer des feuilles de styles plus évolutives (on lie un seul fichier depuis la page HTML, et on gère l'import des feuilles de styles directement depuis ce fichier CSS racine).

Conclusion

Le fonctionnement d'@import peut légèrement ralentir le chargement des styles et donc de la page, et il est déconseillé dans une démarche d'optimisation des performances client.

De plus certaines versions d'Internet Explorer ne comprennent pas cette syntaxe, et n'importeront pas la feuille de style correspondante.

Il est donc préférable et plus conventionnel d'utiliser la méthode « <link /> ».

ECRIRE DU CODE CSS

Vous l'aurez compris, nous écrirons le code dans le fichier ayant l'extension « .css ».

Dans l'exemple ci-dessous, les fichiers sont placés au même endroit, cependant il arrive parfois que le fichier CSS ne soit pas dans le même dossier que les fichiers HTML.

Dans ce cas, lors de l'appel du fichier CSS dans la balise link du fichier HTML, il faut indiquer le bon chemin («../»).



LES COMMENTAIRES

Les commentaires dans le code d'une feuille de style CSS s'inscriront de cette façon :

```
/* ceci est un commentaire */
```

Les commentaires sont des annotations dans le code ignorées par l'interpréteur.

Syntaxe

FONCTIONNEMENT

La syntaxe des CSS est la suivante:

```
sélecteur { propriété:valeur; }
```

Les sélecteurs : pourront être de différents types et nous permettront de sélectionner l'élément dont nous voulons modifier l'apparence.

Par exemple, si nous voulons modifier l'apparence des titres de page sur notre site <h2>, nous devrons écrire « h2 ».

Les propriétés : sont les "effets de style" que nous souhaitons appliquer à un ou plusieurs élément(s).

Il y a par exemple la propriété « color » qui permet d'indiquer la couleur du texte.

Les valeurs : pour chaque propriété nous devons indiquer une valeur.

Par exemple, pour la couleur, il faudra indiquer le nom de la couleur (en anglais) ou le code couleur (en hexadécimale).

Exemple

```
h2{  
color : blue ;  
color : #336699 ;  
}
```

La présence du point-virgule « ; » est obligatoire après chaque valeur (sauf pour la dernière, cependant il est préférable de tout le temps mettre les points-virgules).

Nous pouvons mettre autant de propriétés que nous souhaitons à l'intérieur des accolades.

Il n'est pas utile d'inscrire les deux lignes présentées ci-dessus, une seule suffit, avec le code hexadécimal nous nous octroyons un choix de couleur plus précis.

Avec ce code, tous les titres dans les balises « <h2> Titre </h2> » seront de couleur bleue.

DIFFERENTS TYPES DE SELECTEURS

Les sélecteurs vont nous permettre d'appliquer un style à différents éléments.

- o Le sélecteur universel est noté : *
- o Une balise HTML : p { color:red; }
- o Une classe indépendante : .texte-rouge { color:red; }
- o Un identificateur : #texte-rouge { color:red; }
- o Une sous-classe : p.texte-rouge { color:red; }
- o Des éléments imbriqués (hiérarchie) : div p { color:red; }
- o Un attribut : p[class=exemple] { color:red; }
- o Un pseudo-élément : a:hover { color:red; }

SELECTEUR DE BALISES

Imaginons que nous souhaitons mettre en rouge tous les titres de page et en bleu tous les textes sur notre site Internet, nous allons devoir sélectionner les balises auxquelles vont s'appliquer ce style.

Exemple CSS

```
h1 { color: red }  
p { color: blue }
```

Cela aura pour but de mettre en couleur rouge les titres contenus entre les balises «<h1></h1>» et en bleu tous les textes contenus entre les balises « <p></p> ».

Exemple HTML

```
<h1> Ce titre est en rouge </h1>  
<p> ce texte est en bleu </p>
```

Selecteur de class .

Une classe permet de cibler un élément HTML en lui donnant un nom, sur l'exemple ci-dessous, nous avons choisi la classe « texte-rouge ».

Exemple HTML

```
<p> Paragraphe n°1 </p>
<p class="texte - rouge" > Paragraphe n°2 </p>
<p> Paragraphe n°3 </p>
```

Exemple CSS

```
.texte - rouge{
color : red ;
}
```

La classe définie dans le HTML peut être appelée via la feuille de style CSS afin de cibler un seul des trois paragraphes.

Sans cette classe, nous aurions été obligés d'appeler la balise « p » et cela aurait eu pour effet de styliser les trois paragraphes en couleur rouge, ce qui n'était pas notre objectif.

Une même classe peut être utilisée à plusieurs endroits dans le fichier HTML.

Convention de nommage

Les noms des classes ne doivent pas comporter de caractère accentué et ne pas commencer par un chiffre.

Chaque déclaration de classe CSS commence par un point « . » avant le nom de celle-ci.

Exemple CSS

```
.maclasse{ color : blue ;}
.maclasse{ color : green ;}
```

Si le code est contradictoire, l'interpréteur gardera la dernière version lue.

Il est possible d'ajouter le mot « !important », cela est utile notamment dans un environnement CMS ou lorsque certains scripts JS sont intégrés, pour forcer le style à être redéfini.

Exemple CSS

```
.maclasse{ color : pink !important;}
```

Selecteur identificateur



Un identificateur nous permet également de cibler et d'isoler un élément HTML en lui donnant un nom.

La différence entre un id et une classe réside dans le fait que ce nom doit être unique et ne pas être utilisé 2 fois sur la même page (règle et convention w3c), il doit également être privilégié pour des zones importantes.

Exemple HTML

```
<div id="conteneur"></div>
```

Exemple CSS

```
#conteneur {  
border: 1px solid ;  
}
```

Informations

- o Les id ont une différence à ne pas négliger concernant la règle d'écriture, ils ne commencent pas par un point « . » mais par le symbole dièse « # » dans la feuille de style CSS.
- o Les class et les id vous permettent de définir un style personnalisé.
- o Au contraire des ID, les classes peuvent être appelées à volonté sur différents éléments.
C'est-à-dire qu'une class « texte-rouge » peut être appelée sur des titres représentés par des balises « <h1 class="texte-rouge"></h1> » mais aussi sur du texte représenté par les balises « <p class="texte-rouge"></p> » par exemple.

Selecteur imbriqué & hiérarchie



Afin d'appliquer une mise en forme, nous pouvons parfois avoir besoin d'atteindre un élément contenu dans un autre en citant tout l'acheminement à parcourir pour le localiser.

Exemple HTML

```
<body>
<div id="conteneur">
    <p class="maclasse">Texte</p>
</div>
</body>
```

Exemple CSS

```
body #conteneur .maclasse{
color : red;
}
```

Information importante

- o Les espaces auront pour effet d'indiquer « à l'intérieur de ».
- o Lorsqu'il n'y a pas d'espace, cela veut dire « l'élément qui possède telle classe ou tel id ».

Nous pouvons le lire de la droite vers la gauche : « Trouve l'élément possédant la classe .maclasse, à l'intérieur de #conteneur, à l'intérieur de body »

SELECTEUR SOUS CLASSE

Si une classe a été utilisée sur des éléments différents, nous pouvons toujours cibler l'une d'entre elles sans difficulté. Cela permet également d'ajouter de la précision.

Exemple HTML

```
<p class="ma    classe">Texte n°1</p>
<div class="maclasse">Texte n°2</div>
<h1 class="maclasse">Texte n°3</h1>
```

Suite

Exemple CSS

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

Nous demandons à la feuille de style de trouver dans les éléments « p », celui qui possède la classe « .maclasse » ainsi, le texte n°2 de la div et le texte n°3 de la balise h1 ne risquent pas d'être touchés. Attention, pour cela il est important de ne pas mettre d'espace afin de ne pas changer le sens de la demande.

- o p.maclasse – aura pour effet de d'appeler la balise « p » possédant la classe « .maclasse ».
- o p .maclasse – aura pour effet d'appeler la classe « .maclasse » à l'intérieur d'une balise « p » et non pas SUR la balise « p ».

SELECTEUR ET REGROUPEMENT

Nous pouvons regrouper plusieurs sélecteurs dans une liste, séparés par des virgules, quand ceux-ci partagent la même mise en forme.

Exemple CSS

```
h1 { font - family: verdana }
h2 { font - family: verdana }
h3 { font - family: verdana }
```

Ceci est équivalent à :

```
h1, h2, h3 { font - family: verdana }
```

Dans cet exemple, nous condensons trois règles qui ont les mêmes déclarations en une seule. Cela attribuera la police « verdana » à nos titres : h1, h2, h3.

SELECTEUR D'ATTRIBUTS

Les sélecteurs d'attributs peuvent trouver une correspondance.

Exemple

```
p[class=exemple] { color: blue; }
```

Dans cet exemple, le sélecteur correspond avec tous les éléments « p » dont la valeur de l'attribut « class » est exactement « exemple » :

```
<p class="exemple"> ce texte sera donc de couleur bleu </p>
```

SELECTEUR PSEUDO-ELEMENT

Nous pouvons sélectionner un élément HTML grâce au child.

Exemple HTML

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

Exemple CSS

```
p:first-child {
  color: red;
}
```

Seul le 1er paragraphe sera coloré en rouge grâce au pseudo-élément first-child (voir aussi nth-child, last-child, etc.).

Ceci permet d'éviter de donner une classe, néanmoins, son utilisation doit rester occasionnelle car si les éléments HTML changent d'ordre, cela continuera de styliser le 1^{er} élément, qui ne sera peut-être plus le même.

Exemple HTML

```
<p class="p1"> Paragraphe n°1 </p>
<p class="p2"> Paragraphe n°2 </p>
<p class="p3"> Paragraphe n°3 </p>
```

Exemple CSS

```
p:not(.p3){font-weight: bold;}
```

Tous les paragraphes sauf le 3e (not) seront en gras.

Propriétés et Valeurs

Voici une liste (non exhaustive) des propriétés et valeurs CSS existantes.

MISE EN FORME DU TEXTE ET DES TITRES – POLICE, TAILLE

Propriété	Valeur	Description
font-family @font-face	font-family: Verdana; (police) @font-face { font-family: 'MyFont'; src: url('myfont.woff') format('woff'); }	Permet de modifier la police utilisée sur le site
font-size	font-size : 15px ; (pixels) font-size : 30% ; (pourcentage) font-size : 20em ; (taille relative)	Taille du texte
font-weight	font-weight : bold; (gras) font-weight : normal; (pas gras)	Epaisseur du texte
font-style	font-style : italic ; (italique) font-style : normal ; (normal)	Italique
text-decoration	text-decoration : underline ; (souligné) text-decoration : overline ; (ligne au-dessus) text-decoration : line-through ; (barré) text-decoration : none ; (normal)	Soulignement et diverses mises en forme
font-variant	font-variant : small-caps ; (petites capitales) font-variant : normal ; (normal)	Petites capitales
text-transform	text-transform : uppercase ; (tout mettre en majuscules) text-transform : lowercase ; (tout mettre en minuscules) text-transform : capitalize ; (début des mots en majuscules) text-transform : none ; (normal)	Minuscule, Majuscule.
word-spacing	word-spacing : 10px ; points (pt), inches (in), centimètres (cm), pixels (px) ou pourcentage (%)	définit l'espace entre les mots.
letter-spacing	Letter-spacing : 5px ;	définit l'espace entre les lettres.
font	font: bold 17px Arial;	Ceci est un raccourci pour les différentes propriétés de police. Cela mettra votre texte en gras, 17 pixels, Arial.
content	- une valeur, entre "", - url(url.gif), une image,	permet de spécifier un contenu à insérer avec les propriétés de feuille de style css

Suite

	<ul style="list-style-type: none">- open-quote, ouvre les guillemets.- no-open-quote, n'ouvre pas les guillemets,- close-quote, ferme les guillemets,- no-close-quote, ne ferme pas les guillemets,- compteur,- inherit, hérite des propriétés de son parent.	after ou before
--	--	-----------------

ALIGNEMENT

Propriété	Valeur	Description
text-align	text-align : left ; (à gauche) text-align : center ; (centré) text-align : right ; (à droite) text-align : justify ; (texte justifié)	Alignement horizontal du texte
vertical-align	vertical-align : top ; (en haut) vertical-align : middle ; (au milieu) vertical-align : bottom ; (en bas)	Alignement vertical. A utiliser uniquement dans des cellules de tableau, ou dans des éléments inline.
line-height	line-height : 30px ; valeur en pixels (px) ou en pourcentage (%)	Hauteur de ligne
text-indent	text-indent : 10px ; valeur en pixels (px) pour définir l'alinéa de vos paragraphes.	Alinéa
white-space	white-space : normal ; le passage à la ligne est automatique white-space : nowrap ; pas de passage à la ligne automatique white-space : pre ; le passage à la ligne se fait tel que le texte a été saisi dans le code source	Césure
text-shadow	text-shadow : 1px 1px 6px;	Ombre du texte : (horizontale, verticale, fondu, couleur)

DIMENSIONS

Propriété	Valeur	Description
width	width : 900px ;	Largeur (par défaut, la largeur dépendra du texte à l'intérieur) Valeur en px, %, ou encore "auto"
min-width	min-width : 900px ;	Largeur minimum - en pixels
max-width	max-width : 900px ;	Largeur maximum - en pixels
height	height : 900px ;	Hauteur (par défaut, la largeur dépendra du texte à l'intérieur) - Valeur en px, %, ou encore "auto"
min-height	min-height : 900px ;	Hauteur minimum - en pixels
max-height	max-height : 900px ;	Hauteur maximum - en pixels

Couleurs et fond

Propriété	Valeur	Description
color	color : red ; color : #ff0000 ;	Indique une couleur nom de la couleur en anglais, ou, code couleur en hexadécimal (#CC48A1), ou, en RGB : rgb (128, 255, 0)
background-color	background-color : #336699 ; background-color : blue;	Couleur de fond nom de la couleur en anglais (black, blue, green, white, red...). code couleur en hexadécimal (#CC48A1) couleur en RGB : rgb (128, 255, 0)
background-image	background-image:url(images/fond.png);	Image de fond, arrière plan.
background-attachment	background-attachment : fixed ; background-attachment : scroll ;	Comportement du fond en arrière plan le fond reste fixe quand l'utilisateur descend plus bas sur la page ou défile avec le texte.
background-repeat	background-repeat : repeat ; background-repeat : repeat-x; background-repeat : repeat-y; background-repeat : no-repeat;	Répétition de l'image de fond : Repeat : le fond se répète repeat-x: le fond ne se répète que sur une ligne, horizontalement repeat-y : le fond ne se répète que sur une colonne, verticalement no-repeat : le fond ne se répète pas, il n'est affiché qu'une fois
background-position	background-position:50px 200px; background-position : bottom right;	Position du fond
background-size	background-size: cover;	Permet de couvrir tout l'écran, peu importe la taille de l'image utilisée et la résolution d'écran de l'internaute visitant le site
background-linear-gradient	background: linear-gradient(#f6eadd, #f9dac3);	Permet d'avoir une couleur de fond en dégradé
background	background: #fff url("images/fond.png") no-repeat fixed top left;	Ceci est un raccourci pour les différentes propriétés de fond. Indiquer une ou plusieurs valeurs issues des propriétés background-color, background-image, background-repeat, background-attachment, background-position.
opacity	0.5	Transparence

MARGES EXTERIEURES

Propriété	Valeur	Description
margin-top	margin-top : 10px ;	Marge extérieure en haut - pixel, pourcentage, em, etc.
margin-left	margin-left: 10px ;	Marge extérieure à gauche - pixel, pourcentage, em, etc.
margin-right	margin-right: 10px ;	Marge extérieure à droite - pixel, pourcentage, em, etc.
margin-bottom	margin-bottom: 10px ;	Marge extérieure en bas - pixel, pourcentage, em, etc.
margin	margin : 30px 25px 20px 15px ; <ul style="list-style-type: none"> o 1 valeur : ce sera la marge pour le haut, le bas, la gauche et la droite o 2 valeurs : la première correspond à la marge pour le haut et le bas, la seconde pour la gauche et la droite o 3 valeurs : la première correspond à la marge du haut, la seconde aux marges de gauche et de droite, la troisième à la marge du bas o 4 valeurs : respectivement la marge du haut, de la droite, du bas, de la gauche. 	Ceci est un raccourci pour les différentes propriétés de marges extérieures. Indiquez de 1 à 4 valeurs à la suite. Selon le nombre de valeurs que vous mettez, la signification change.

MARGES INTERIEURES

Propriété	Valeur	Description
padding-top	padding-top : 10px ;	Marge intérieure en haut - pixel, pourcentage, em, etc.
padding-left	padding-left: 10px ;	Marge intérieure à gauche - pixel, pourcentage, em, etc.
padding-right	padding-right: 10px ;	Marge intérieure à droite - pixel, pourcentage, em, etc.
padding-bottom	padding-bottom: 10px ;	Marge intérieure en bas - pixel, pourcentage, em, etc.
padding	<p>padding : 30px 25px 20px 15px;</p> <ul style="list-style-type: none"> o 1 valeur : ce sera la marge pour le haut, le bas, la gauche et la droite o 2 valeurs : la première correspond à la marge pour le haut et le bas, la seconde pour la gauche et la droite o 3 valeurs : la première correspond à la marge du haut, la seconde aux marges de gauche et de droite, la troisième à la marge du bas o 4 valeurs : respectivement la marge du haut, de la droite, du bas, de la gauche. 	<p>Ceci est un raccourci pour les différentes propriétés de marges intérieures.</p> <p>Indiquez de 1 à 4 valeurs à la suite.</p> <p>Selon le nombre de valeurs que vous mettez, la signification change.</p>

BORDURES

Propriété	Valeur	Description
border-width	border-width : 3px ;	Epaisseur de la ligne de bordure.
border-color	border-color : #ff0000 ;	Couleur de la ligne de bordure <ul style="list-style-type: none"> o nom de la couleur en anglais (black, blue, green, white, red...). o code couleur en hexadécimal (#CC48A1) o couleur en RGB : rgb (128, 255, 0)
border-style	border-style : none ; (pas de bordure) border-style : hidden ; (bordure cachée) border-style : solid ; (ligne pleine) border-style : double ; (ligne double) border-style : dashed ; (en tirets) border-style : dotted ; (en pointillés) border-style : inset ; (effet 3D "enfoncé") border-style : outset ; (effet 3D "surélevé") border-style : ridge; (autre effet 3D)	Type de bordure
border-left (top, right, bottom)	border-left: 2px inset blue;	Bordure à gauche ou encore (en haut, à droite, en bas). Nous pouvons indiquer la couleur, l'épaisseur et le type de bordure :
border-radius	Border-radius: 10px;	Bordure arrondie
Box-shadow	Box-shadow: 1px 1px 5px;	Ombre sur un conteneur (ex : div)
border	Indiquer la couleur, l'épaisseur et le type de bordure :	Concernera l'apparence des bordures en haut, à droite, en bas et à gauche.

Suite

POSITIONNEMENT

Propriété	Valeur	Description
float	float : left ; (flottant à gauche) float : right ; (flottant à droite) float : none ; (pas de flottant)	Cela permet aux « zones » de s'afficher en colonne.
clear	clear : left ; (supprime l'effet d'un flottant à gauche précédent) clear : right ; (supprime l'effet d'un flottant à droite précédent) clear : both ; (supprime l'effet d'un flottant précédent, qu'il soit à gauche ou à droite) clear : none ; (pas de suppression de l'effet du flottant)	Stopper un effet flottant
position	position : absolute ; (position absolue par rapport au coin en haut à gauche) position : fixed ; (position fixe: L'élément reste à sa position même quand on descend plus bas dans la page). position : relative ; (position relative, par rapport à la position "normale" de l'élément) position : static ; (positionnement normal)	Type de positionnement
top (ou encore: bottom, left, right)	top : 10px ;	Position par rapport au haut (ou encore : bas, gauche, droite). Valeur en px, %, em... A utiliser pour un positionnement absolu, fixe ou relatif.

AFFICHAGE

Propriété	Valeur	Description
display	<p>display : none ; (l'élément ne sera pas affiché)</p> <p>display : block ; (l'élément devient de type "block", étendue sur toute la largeur)</p> <p>display : inline ; (l'élément devient de type "inline", en ligne, il ne prend que la place dont il a besoin)</p> <p>display : list-item ; (l'élément devient de type "élément de liste à puces")</p>	Type d'élément
visibility	<p>visibility : hidden ; (l'élément sera masqué)</p> <p>visibility : visible ; (l'élément sera visible)</p>	Affichage
clip	clip: rect(valeur1, valeur2, valeur3, valeur4);	Affiche seulement une partie d'un élément
overflow	<p>overflow : visible ; (tout l'élément sera affiché).</p> <p>overflow : hidden ; (l'élément sera coupé s'il dépasse les limites définies par height et width. On ne pourra pas voir la partie du texte coupé).</p> <p>overflow : scroll ; (tout comme hidden, l'élément sera coupé s'il dépasse les limites. Toutefois, cette fois le navigateur ajoutera des barres de défilement pour que l'on puisse voir la suite du texte).</p> <p>overflow : auto ; (c'est le navigateur qui décide d'ajouter des barres de défilement ou pas en fonction des cas. Bien souvent, utiliser cette valeur revient à utiliser la valeur "scroll").</p>	Limite les dimensions
z-index	<p>z-index : 100 ;</p> <p>Indiquer un nombre.</p>	Ordre d'affichage. En cas de positionnement absolu par exemple, si 2 éléments se chevauchent, z-index permet d'indiquer quel élément doit être affiché au-dessus de l'autre.

LES TABLEAUX

Propriété	Valeur	Description
border-collapse	border-collapse : collapse ; (les bordures du tableau et des cellules sont mélangées.) border-collapse : separate ; (les bordures du tableau et des cellules sont séparées)	Type de bordure
empty-cells	empty-cells : show ; (les bordures des cellules vides sont affichées). empty-cells : collapse ; (les cellules vides sont masquées)	Cellules vides
caption-side	via la balise <caption> : caption-side : top ; (en haut du tableau) caption-side : bottom ; (en bas du tableau) caption-side : left ; (à gauche du tableau) caption-side : right ; (à droite du tableau)	Position du titre

Suite

LES LISTES

Propriété	Valeur	Description
list-style-type	<p>Pour les listes non ordonnées () :</p> <ul style="list-style-type: none"> list-style-type : disc ; (un disque noir). list-style-type : circle ; (un cercle). list-style-type : square ; (un carré). list-style-type : none ; (aucune puce ne sera utilisée). <p>Pour les listes ordonnées () :</p> <ul style="list-style-type: none"> list-style-type : decimal ; (des nombres 1, 2, 3, 4, 5... (par défaut)) list-style-type : decimal-leading-zero ; (des nombres commençant par zéro (01, 02, 03, 04, 05...)). list-style-type : upper-roman ; (numérotation romaine, en majuscules (I, II, III, IV, V...)) list-style-type : lower-roman ; (numérotation romaine, en minuscules (i, ii, iii, iv, v...)) list-style-type : upper-alpha ; (numérotation alphabétique, en majuscules (A, B, C, D, E...)) list-style-type : lower-alpha ; (numérotation alphabétique, en minuscules (a, b, c, d, e...)) list-style-type : lower-greek ; (numérotation grecque). 	Type de liste
list-style-position	list-style-position : inside ; (sans retrait) list-style-position : outside ; (avec retrait)	Position en retrait
list-style-image	Indiquer l'url de l'image qui servira de puce : list-style-image: url(images/puce.png);	Puce personnalisée
list-style	Vous pouvez réunir les valeurs de list-style-type, list-style-position et list-style-image.	Raccourci pour les différentes propriétés de style de liste.

EFFETS

Propriété	Valeur	Description
box-reflect	Exemple pour google chrome : -webkit-box-reflect: below 1px -webkit-gradient(linear, 0 0%, 0 100%, from(transparent), color-stop(0.7, transparent), to(white));	Cette propriété permet d'ajouter un reflet à une image
transition	transition: all 0.5s ease;	Produit un effet de transition
transition-duration	transition-duration: .8s;	Durée de transition
animation	frame-animation 1.5s steps(10) infinite;	Permet de "jouer" une animation
transform	transform: rotate(-360deg);	Permet d'effectuer une rotation d'un élément (par exemple une image)
@keyframes		Déroulement de votre animation
@media	@media screen and (max-width:950px)	Permet d'adapter le site à différentes résolution (Responsive Design)

La version 3 du langage CSS permet beaucoup d'évolutions notamment sur différents effets.

Autres propriétés

Propriété	Valeur	Description
cursor	<p>cursor : auto ; (curseur automatique)</p> <p>cursor : default ; (curseur standard)</p> <p>cursor : pointer ; (curseur en forme de main, comme quand on pointe sur un lien)</p> <p>cursor : text ; (curseur utilisé quand on pointe sur du texte)</p> <p>cursor : wait ; (curseur utilisé pour indiquer une attente (sablier))</p> <p>cursor : progress ; (curseur utilisé pour indiquer une tâche de fond (curseur avec sablier))</p> <p>cursor : help ; (curseur en forme de point d'interrogation, indiquant une aide)</p> <p>cursor : move ; (curseur en forme de croix, indiquant un déplacement possible)</p> <p>n-resize : flèche vers le nord</p> <p>ne-resize : flèche vers le nord-est</p> <p>e-resize : flèche vers l'est</p> <p>se-resize : flèche vers le sud-est</p> <p>s-resize : flèche vers le sud</p> <p>sw-resize : flèche vers le sud-ouest</p> <p>w-resize : flèche vers l'ouest</p> <p>nw-resize : flèche vers le nord-ouest</p> <p>url : curseur personnalisé, de type .cur ou .ani.</p>	Curseur de souris
column-rule	column-rule:4px outset #ff00ff;	column-rule : groove (rainurée) ridge (relief) inset (relief intérieur) outset (relief extérieur)
Column-gap	column-gap:40px;	espace entre les colonnes
column-count	column-count:3;	nombre de colonne
Resize	resize: horizontal;	Permet d'autoriser l'internaute à

		<p>redimensionner une balise textarea (par exemple) horizontalement.</p>
--	--	--

Impression

Propriété	Valeur	Description
page-break-before	auto, always (toujours insérer un saut de page avant l'élément), avoid (jamais), left (force un ou deux sauts de page pour que la page suivante imprimée soit une page gauche), right (force un ou deux sauts de page pour que la page suivante imprimée soit une page gauche)	Permet de créer / d'empêcher un saut de page avant un élément
page-break-after	auto, always (toujours insérer un saut de page après l'élément), avoid (jamais), left (force un ou deux sauts de page pour que la page suivante imprimée soit une page gauche), right (force un ou deux sauts de page pour que la page suivante imprimée soit une page gauche)	Permet de créer / d'empêcher un saut de page après un élément
page-break-inside	auto, avoid (ne jamais insérer un saut de page dans l'élément), left (force un ou deux sauts de page pour que la page suivante imprimée soit une page gauche), right (force un ou deux sauts de page pour que la page suivante imprimée soit une page gauche)	Permet de créer / empêcher un saut de page dans un élément
orphans	nombre entier positif non-nul	Permet d'indiquer le nombre minimum de lignes de l'élément devant être imprimées sur une même page avant de passer à la page suivante
windows	nombre entier positif non-nul	Permet d'indiquer le nombre maximum de lignes de l'élément devant être imprimées sur une même page avant de passer à la page suivante
size	auto, une largeur et une longueur, landscape (orientation paysage) ou portrait (orientation portrait)	Permet de définir la taille de la page d'impression et / ou le format d'impression
marks	none (par défaut), crop (traits de coupe), cross (repères de montage)	Permet de définir les traits de coupes et les repères de montage

Pseudo éléments

Il existe des pseudo attributs qui peuvent venir s'ajouter au nom d'un sélecteur.

Propriété	Valeur	Description
:link :visited :hover :active	Fonctionne sur un lien et la balise « a »	:link (lien par défaut) :visited (lien déjà visité) :hover (lien au survol) :active (lien actif)
:target	#competences:target #defilement{ left: -700px; }	Permet de créer un comportement CSS en récupérant une information dans l'url
:first-child, :nth-child, :last-child	p:first-child { color: green; } /* 1er élément p - css3 */ p:nth-child(even){ color: blue; } /* éléments pair - css3 */ p:nth-child(odd){ color: brown; } /* éléments impair - css3 */ p:nth-child(5), p:nth-child(6) { color: orange; } /* éléments précis n°5 et n°6 - css3 */ /* p:last-child { color: black ; } dernier élément p - css3 */ div:not(.zoneB){ font-weight: bold; } /* toutes les div sauf la zone B - not() - css3 */	Désigne un élément premier (first), dernier (last), ou particulier (nth)
::selection		Désigne le texte sélectionné et permet notamment de changer la couleur de fond lors d'un surlignement (pour un copier/coller)
:first-Letter		Désigne la première lettre de chaque bloc de texte.
:first-Line		Désigne la première ligne de chaque bloc de texte.
:empty		Désigne les éléments vides. Exemple : TD:Empty désigne les cellules de tableau vide. Un ou plusieurs espaces sont considérés comme du vide mais l'espace fixe () suffit à "remplir" l'élément.

L'ATTRIBUT MEDIA

On utilise le paramètre média (à l'intérieur de la balise style) pour obtenir des présentations différentes, selon le type de média utilisé (ordinateur, synthétiseurs vocaux, ...).

Tableau résumant les valeurs disponibles pour le paramètre média.

- o all Pour tous les types de périphériques
- o screen type classique, pour tous les moniteurs courants
- o tty type télévision (basse résolution)
- o projection type projecteur
- o handheld type PC de poche (Psion, Windows CE, Pilot ...)
- o print type impression (pour les imprimantes)
- o braille Pour les dispositifs de représentation en braille
- o aural Pour les synthétiseurs vocaux

L'héritage des CSS est fondé sur le modèle Parent-Enfant(s) : chaque élément Enfant reçoit en héritage tous les styles de son élément Parent. Par exemple, la balise <html> est parent de <body>, et <table> est parent de <tr> qui lui-même est parent de <td>.

Cet héritage est très pratique et permet d'éviter beaucoup de répétitions inutiles : en effet, en attribuant une propriété à un parent (par exemple font-family: Tahoma), elle sera transmise à tous ses enfants, mais aussi aux enfants de ses enfants, etc.

Précision : l'élément enfant héritera de toutes les propriétés de l'élément parent uniquement si ces propriétés s'héritent car l'héritage ne fonctionne pas non plus sur toutes les propriétés CSS.

Tel que : margin, padding et autres propriétés de bloc.

Exemple CSS

```
body{  
color : red ;  
}
```

Tous les textes seront de couleur rouge quelles que soient leurs balises, car tous les éléments sont des enfants ou petits-enfants de body.

Les éléments enfants sont affectés tant que cet élément enfant n'a pas une valeur propre « d'affecté » pour la propriété.

Exemple CSS

```
h1{  
color : blue ;  
}
```

La propriété est changée en cours de route et nous déclarons que les h1 seront de couleur bleue.

Exemple HTML

```
<body>  
    <h1>Mon titre </h1>  
</body>
```

Suite

Le titre h1 hérite de la couleur d'écriture rouge, mais entre-temps nous lui avons attribué une autre valeur (color : blue) il perd donc son héritage (couleur : red) provenant de body au profit de la nouvelle valeur.

HERITAGE D'ORIGINE

Par exemple, la balise h1 hérite d'une taille d'écriture particulière, de marges, etc.

Cela est implicite, cependant, il est possible d'intervenir en lui attribuant de nouvelles marges et une nouvelle taille d'écriture.

DESHERITAGE

Afin de casser l'héritage qui n'est pas toujours souhaité, il suffit d'appeler toutes les balises et de réinitialiser les propriétés ayant été définies autrement par les normes w3c et le navigateur.

Nous appelons cela, un « reset »

Exemple CSS

```
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, 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, fi           gcaption, 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 - decoration: none;
}
```

Exemple CSS

```
* {  
    margin: 0;  
    padding: 0;  
    border: 0;  
    font-size: 100%;  
    font: inherit;  
    vertical-align: baseline;  
    text-decoration: none;  
}
```

Dans cet exemple, nous pouvons faire la même chose (reset CSS) sans appeler toutes les balises, en passant par le sélecteur universel « * ».

Il est quand même conseillé d'appeler toutes les balises, car si nous devons en retirer une, cela reste possible.

Mise en page et positionnement

TYPE BLOC ET TYPE INLINE

Type Bloc

C'est un élément (ou une zone) qui prend par défaut toute la largeur de la page.

Les balises de type BLOC HTML sont des boites d'éléments qui peuvent contenir d'autres éléments tels que du texte, des images, des liens ou autres... Elles peuvent contenir d'autres balises de type bloc et aussi contenir d'autres balises de type inline.

Par défaut, les blocs se placent toujours l'un en dessous de l'autre.

Quelques balises de type BLOC « block » : <p>, , , <div>, <form>, <input>, <blockquote>, <table>, <h1>...<h6>,...

Type Inline

C'est un élément (ou une zone) qui prend uniquement la place dont il a besoin

Une balise inline ne peut contenir uniquement une (ou plusieurs) balise inline, et n'a pas de dimensions à proprement parler (aucune largeur et aucune hauteur. Elle n'occupe que la place minimum nécessaire à son contenu).

Par défaut, les inline se placent toujours l'un à côté de l'autre.

Quelques exemples de balises de type Inline « en ligne » : <a>, , , , , Description block, inline, etc.

Il est possible de transformer une balise de type inline en type bloc et vice versa.

LA POSITION ABSOLUE

Le positionnement absolu « retire » totalement du flux le contenu concerné : sa position est déterminée par référence aux limites du conteneur. Celui-ci peut être :

- o Une boîte elle-même positionnée (position relative ou absolue) ;
- o Le bloc conteneur initial, à défaut de boîte positionnée, c'est-à-dire en pratique le plus souvent la fenêtre du navigateur.

Code HTML :

```
<div id="zone">
    <p> Bonjour tout le monde </p>
    <p class="orange"> Hello! </p>
</div>
```

Code CSS :

```
p{ margin: 0; }
.orange{ color: #ff9c00; position: absolute; top: 50%; left: 0px; }
#zone{ border: 2px solid #336699; }
```

Résultat :



Inévitablement, le paragraphe en position absolue sort de la zone qui possède les bordures bleues. Avec cette position, le paragraphe prend en référence la fenêtre et se place au milieu verticalement puis à gauche horizontalement.

LA POSITION RELATIVE

Un élément en position relative permet de se placer par rapport aux limites de la zone dans laquelle il se trouve. Sa position est déterminée par référence aux limites du conteneur.

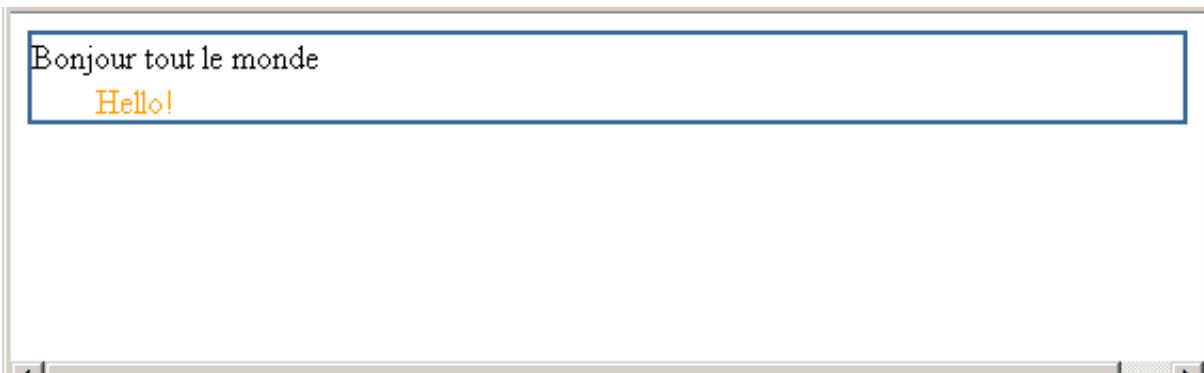
Code HTML :

```
<div id="zone">
    <p> Bonjour tout le monde </p>
    <p class="orange"> Hello! </p>
</div>
```

Code CSS :

```
p{ margin: 0; }
.orange{ color: #ff9c00; position: relative; left: 30px; }
#zone{ border: 2px solid #336699; }
```

Résultat :



Le paragraphe en position relative positionnée de 30px à gauche se décale en prenant le bord du conteneur avec les bordures bleues comme référence et non le bord de la fenêtre.

Avec cette position, le paragraphe prend en point de repère la zone dans laquelle il se trouve.

LA POSITION FIXE

Comme dans un positionnement absolu, le contenu concerné est retiré totalement du flux. Mais il est cette fois positionné uniquement par rapport aux limites de la zone de visualisation, autrement dit la fenêtre du navigateur. Le défilement de la page n'a aucun effet sur un contenu en position fixe.

Code HTML :

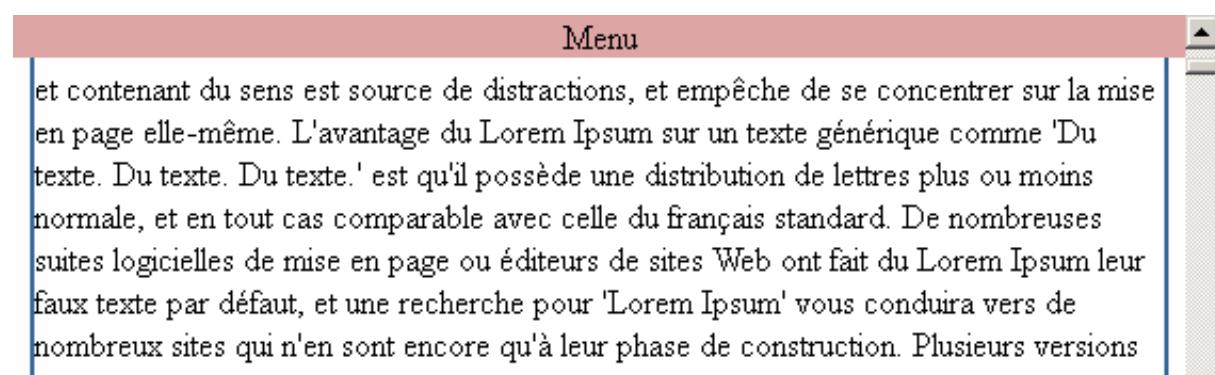
Code CSS:

```
.menufixe {  
background-color: #dea5a5;  
position: fixed;  
text-align: center;  
width: 100%;  
top: 0;  
left: 0;  
}  
.contenu { border: 2px solid #336699; margin: 30px 0 0 0; }
```

Résultat :

Pendant que nous descendons la page à l'aide de l'ascenseur, la zone du menu ne bouge pas.

Le positionnement CSS donne donc toute liberté à la mise en page.



Suite

LES FLOTTANTS

Une boîte flottante est retirée du flux normal, et placée le plus à droite (float: right) ou le plus à gauche (float: left) possible dans son conteneur.

Cela permet de faire plusieurs colonnes sur une page web.

Le contenu suivant cette boîte flottante se place le long de celle-ci, dans l'espace laissé libre.

Exemple n°1

Code HTML :

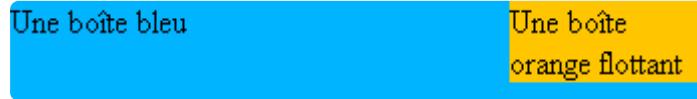
```
<p class="orange">Une boîte orange flottant</p>
<p class="bleu">Une boîte bleu           e</p>
```

Code CSS:

```
.orange {
    background-color: #ffc600;
    float: right;
    width: 100px;
    margin: 0;
}

.bleu {
    background-color: #00b4ff;
    height: 50px;
}
```

Résultat :



Le navigateur place tout d'abord la boîte orange, alignée sur le bord droit de cette partie de notre page, puis reprend le cours normal du flux dans l'espace laissé libre à sa gauche pour afficher la boîte bleue.

Suite

Exemplen°2

Code HTML :

```
<div class="colonne1"> 1 </div>
<div class="colonne2"> 2 </div>
```

Code CSS:

```
.colonne1 {
    float: left;
    width: 70px;
    height: 70px;
    text-align: center;
    border: 1px solid #000000;
}

.colonne2 {
    float: left;
    width: 70px;
    height: 70px;
    text-align: center;
    border: 1px solid #000000;
}
```

Résultat :

1	2
---	---

La combinaison de la position flottante et du flux permet de réaliser aisément une mise en page à deux colonnes.

Le flux doit être arrêté via la propriété clear.

Suite

Exemple n°3

Code HTML :

```
<div class="zone1"> 1 </div>
<div class="zone2"> 2 </div>
<div class="zone3"> 3 </div>
<div class="zone4"> 4 </div>
<div class="zone5"> 5 </div>
```

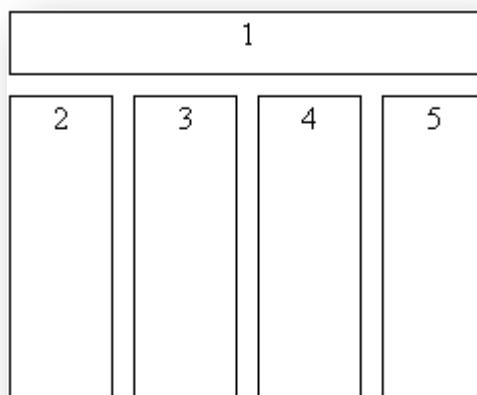
Code CSS:

```
.zone1 {
    float: left;
    width: 236px;
    height: 30px;
    text-align: center;
    border: 1px solid #000000;
}

.zone2, .zone3, .zone4, .zone5{
    float: left;
    margin: 10px 10px 0 0;
    width: 50px;
    height: 150px;
    text-align: center;
    border: 1px solid #000000;
}

.zone2{    clear: left;        }
```

Resultat :



La combinaison de la position flottante et du flux permet de réaliser aisément une mise en page à plusieurs colonnes.

COMMENT CENTRER SON SITE ?

Lors de l'élaboration du design d'un site nous souhaitons souvent centrer les éléments sur la page.

Une méthode préconisée consiste à faire une zone appelée « conteneur » qui sera centrée sur la page et ensuite mettre d'autres zones à l'intérieur de celle-ci qui seront par conséquent centrées aussi.

Voici un 1^{er} exemple

Code HTML:

```
<div id="conteneur">Au Centre!</div>
```

Code CSS:

```
#conteneur{  
width: 960px;  
background: #F4C400;  
margin: 0 auto;  
}
```

Résultat :



LES LARGEURS EN POURCENTAGE

Il peut être intéressant dans certains cas d'utiliser une zone ayant une largeur en pourcentage.

De cette manière la taille de la zone prendra la taille de la fenêtre (cette taille est définie par la résolution de l'utilisateur).

Voici un exemple

Code HTML

```
<div class="taillefixe">Fixe </div>  
<div class="taillevariable">Variable </div>
```

Code CSS

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

Résultat



Afin de se rendre compte du résultat de cet exemple, il faut réduire/agrandir la taille de la fenêtre.

HIERARCHIE : ANCETRE, PARENTS, ENFANTS, FRERES

Comprendre l'imbrication des éléments (boîtes) les uns dans les autres est essentiel.

Chaque document HTML est toujours composé de conteneurs. Ceux-ci peuvent être ancêtres, parents, enfants ou frères. Ces différents éléments composent une hiérarchie d'imbrications.

- o Un élément Ancêtre est un élément qui contient un élément ou une hiérarchie d'éléments.
- o Un bloc Parent est un élément contenant directement un autre bloc. Par exemple, un DIV contenant un paragraphe P.
- o Un bloc contenu directement dans un autre bloc est dit Enfant de cet élément. Par exemple, ici les éléments LI sont enfants de leur conteneur UL.
- o Les éléments ayant le même élément Parent sont appelés Frères.

RESOLUTION D'ECRAN

Avant de commencer la mise en forme d'un site Internet, il est important de prendre en compte la résolution d'écran de l'internaute.

Voici les chiffres officiels du w3c pour janvier 2014 :

Résolution	Pourcentage (%)
800x600	0.5%
1280x768	6%
1280x800	7%
1280x1024	8%
1366x768	31%
1440x900	7%
1680x1050	11%
1920x1080	13%
1920x1200	3%
Autres	13.5%

CONCEPTION D'UN SITE INTERNET

Nous avons vu que le HTML était un langage de balise, le CSS vas permettre de « styliser » ces balises.

Un site Internet est composé d'un ensemble de zones dimensionnées et positionnées permettant ainsi d'obtenir un design.

Lors de la conception d'un site Internet, le langage HTML/CSS est en général confié à un intégrateur.

C'est lui qui reçoit la maquette (image) du designer et qui procède à la découpe d'images.

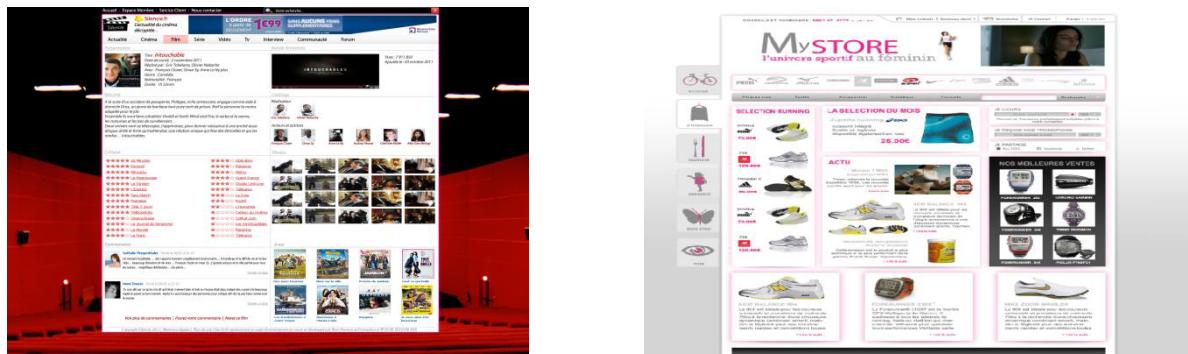
Avant de foncer tête baissée et de rentrer dans le vif du sujet, il est utile de se poser quelques questions :

- o Pour la hauteur et la largeur, le site doit-il avoir une taille fixe ou extensible ?
- o Imaginer le positionnement et le nombre de zones sur la maquette

DE LA CREA GRAPHIQUE A L'INTEGRATION

Lors de la conception d'un site Internet, un designer crée la maquette graphique du site, il s'agit d'une image qui indique de quelle façon doit s'afficher le site Internet.

Exemple



Le rôle de l'intégrateur est d'évaluer les zones à mettre en place au vu de cette maquette graphique et de découper les images dont il a besoin pour commencer l'intégration avec les langages HTML/CSS.

INTRODUCTION

Avec la démocratisation des Smart phones, il devient parfois nécessaire de penser en concevant un site à sa version sur petit écran.

Responsive Design ou Version Mobile

Une version spéciale Smart phone (petit écran) est une étape qui peut être compliquée à envisager. Heureusement, les standards du Web offrent quelques armes fortes bienvenues et efficaces pour la franchir.

Vous le savez, un smart phone n'a pas un écran ni une résolution plus importante qu'un pc, après tout ce n'est pas grave : présenter un site en 480*320 pixels n'est jamais... qu'un style alternatif.

Afin de savoir si l'internaute se connecte à l'aide d'un pc ou un smart phone il est judicieux de mettre côté serveur un script de détection de l'agent utilisateur. Cela permettrait de le rediriger vers une version adaptée.

Par exemple :

Si l'utilisateur qui tente d'accéder à la page navigue via un iPhone, Android, ou autres (à préciser)... alors nous appliquerons style_mob.css sinon nous appliquerons style_pc.css

Les media-queries sont une nouveauté introduite par CSS 3 et permettent d'appliquer des styles selon les paramètres du média (résolution, etc.)

Exemple

```
<link rel="stylesheet" type="text/css" href="style_          mobcss" media="screen" and (max -  
width: 480px)      "/>
```

Notre feuille CSS "style_mob.css" sera donc appliquée si le média est de type screen et si sa largeur maximale est de 480 pixels

QUELQUES CONSEILS

Lors de la conception du style de son site, il va de soi que vous devez tester la répercussion sur votre page dans tous les navigateurs et ce, sans attendre la fin mais à chaque étape.

Il est fortement recommandé de valider son code HTML et CSS par le w3c.

Pour identifier un problème (souvent de positionnement, en CSS) il est utile de changer la couleur de fond d'une zone afin de la démarquer du reste du contenu et observer si elle a le comportement attendu.

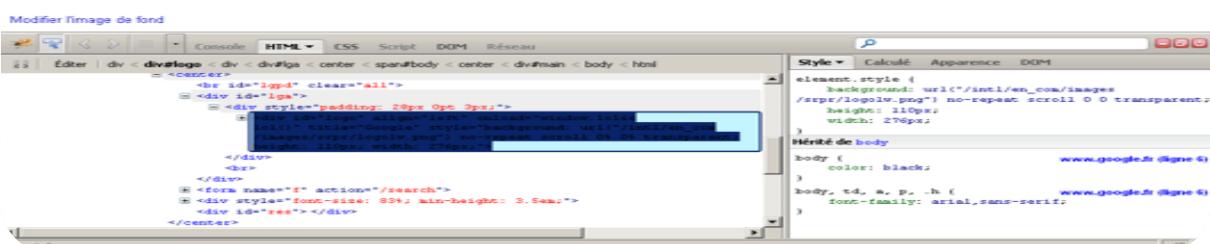
En cas de dernier recours, une solution serait de dupliquer la feuille de style et de retirer des éléments un à un afin de remonter jusqu'à l'origine du problème rencontré.

DES OUTILS INDISPENSABLES : WEBDEVELOPPER ET FIREBUG

Firebug

Firebug est une extension pour Mozilla Firefox qui permet de déboguer, éditer et modifier le HTML, le CSS et le JavaScript d'une page web.

Firebug a été développé par Joe Hewitt, un des créateurs de Firefox.



Cette extension est très intéressante car elle permet de visualiser le CSS de chaque zone sur un site web et de pouvoir opérer des modifications pour observer le changement en direct.

Webdevelopper

Web Developer est une autre extension pour Firefox permettant d'analyser la structure de la page en cours, ses feuilles de style CSS, ses formulaires, ses scripts java, ses images, etc.

Vous pouvez modifier la feuille CSS actuelle et voir les effets en temps réel, charger votre propre feuille de style ou désactiver des styles.

D'ailleurs, en décrivant des tableaux, des DIV ou des cadres, vous pouvez avoir de nombreuses options pour visualiser les éléments CSS et HTML de la page en cours.

En outre, il vous autorise à inclure des liens de validations de code et même à redimensionner la fenêtre du navigateur.

Doté d'une visionneuse et des syntaxes colorées, cette extension va vous aider à concevoir un design unique pour votre site web personnel.



D'autres outils similaires existent pour les autres navigateurs et sont parfois inclus par défaut (c'est le cas de Google Chrome ou Internet Explorer).