



KompoZer

Un éditeur de pages Web

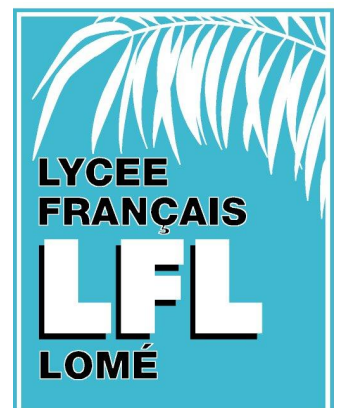


Table des matières

Introduction	5
Présentation de KompoZer	6
Première page	8
Formatage du texte	8
Couleurs du texte	9
Couleurs et fond de page	10
Les paragraphes	11
Les listes	12
Retrait et alignement	12
Propriétés de la page	13
Structure d'un site Internet	13
Introduction	13
Du local au distant	14
Racine du site	14
URL relative et absolue	15
Première approche des éléments HTML	17
Introduction	17
L'onglet "Balises HTML"	18
Les attributs	18
Liste des éléments HTML	19
<body>...</body>	20
...	20
<div>...</div>	20
<a>...	21
	22

Les tableaux	24
Les formulaires	24
Positionnement des éléments HTML	25
Introduction	25
Le positionnement et les dimensions	25
La superposition	26
Les tableaux	27
Insertion d'un tableau	27
Modification du tableau	28
Les formulaires	28
Introduction	28
Principe de fonctionnement	29
Les feuilles de styles	29
Introduction	29
Application directe d'un style à un élément	30
<i>Général</i>	30
<i>Texte</i>	31
<i>Fond</i>	32
<i>Bordures</i>	32
<i>Boîte</i>	32
<i>Listes</i>	34
<i>Audio</i>	34
Les règles de style	34
Affecter des règles aux éléments	36
Gestion des feuilles de styles	37
<i>Introduction</i>	37
<i>Les différentes règles</i>	37
<i>Créer une feuille de style externe</i>	38
<i>Gestion des feuilles de styles et des règles d'une page</i>	38
Javascript	39
Introduction	39

I. Introduction :

Le logiciel KompoZer, gratuit et libre de droit, est un éditeur de pages Web : il permet de modifier et de créer des pages Web aussi facilement que Word permet d'éditer un document texte. Nous allons à travers ce logiciel apprendre à créer des pages Web et construire un site Web

Son design et son utilisation sont assez proches de DreamWeaver, le plus célèbre des éditeurs de pages Web : ce dernier est largement plus abouti mais en contre partie, il est payant et n'existe que sous Windows ; KompoZer est quant à lui disponible sous toutes les plates-formes (*Windows, Linux, MacOS*).

Une page Web est généralement constituée d'images, de liens et du texte ; derrière cette simplicité se trouve le langage informatique HTML. C'est lui qui indique au navigateur comment afficher les différents éléments de la page. Le langage HTML est facile à apprendre : sa syntaxe est réduite et son vocabulaire est pauvre.

Les éditeurs de pages Web, tel que KompoZer ou Dreamweaver permettent, de construire des pages Web sans aucune connaissance de ce langage : à chaque action de l'utilisateur, l'éditeur transforme cela par du code HTML représentant cette action. On dit que ces éditeurs sont WYSIWYG.



WYSIWYG est l'acronyme de "*What you see is what you get*", c'est à dire que vous vous contenterez de cliquer, de déplacer des objets, d'utiliser des commandes dans la barre de menu ; le logiciel se chargera de créer le code correspondant à vos manipulations.

Par contre, KompoZer se révélera insuffisant pour la construction de pages Web dynamiques (*avec le langage **JavaScript***) ou de sites de grande tailles (*avec le langage **PHP** et une base de données*). L'apprentissage du langage HTML, en premier lieu, est alors indispensable.

Le langage HTML a ses débuts n'était pas standardisé ; les deux compagnies Netscape et Microsoft se sont livrés une guerre au travers de leurs navigateurs cherchant, chacun, à imposer leur norme. Ainsi, une page Web peut s'afficher sur "*Netscape Navigator*" et "*Internet Explorer*" avec des différences, s'avérant parfois, gênantes.

Un organisme de standardisation, le W3C (*Word Wide Web Consortium*), a vu le jour pour imposer des normes dans les langages Web. Actuellement, le langage HTML ne présente plus de différences sur les différents navigateurs les plus célèbre "*Internet Explorer*", "*Opera*" et "*Firefox*", mais des différences persistent sur l'utilisation des feuilles de styles CSS et le langage **JavaScript**.

L'apprentissage de KompoZer nous permettra de nous familiariser avec la construction de

pages Web et à la structure d'un site Web. Nous aurons l'occasion de présenter également les feuilles de style CSS (*langage contrôlant l'affichage et le positionnement des éléments d'une page Web*); nous terminerons cette première phase de la formation en mettant notre site en ligne sur Internet.

⚠ Tout au long de cette formation, on notera l'accès à une commande de la barre des menus de KompoZer de la manière suivante :

Insertion ~> Formulaire ~> Zone de texte...

Cet encadré indique la commande “Zone de texte” se trouvant dans le menu *Insertion* et dans le sous-menu *Formulaire*.

Voici quelques définitions et remarques qui vous permettront de lire plus facilement la suite de ce fascicule :

- On appelle “navigateur” les logiciels affichant les pages Webs tels que “Internet Explorer”, “Firefox” ; à la différence des éditeurs qui permettent l'affichage et, surtout, la modification des pages.

- On appelle “client” à la fois la personne et l'ordinateur visualisant une page Web ; à l'opposée, se trouve le serveur qui lui contient les pages Webs et les envoie aux clients.

- Une page Web est composé d'*éléments* HTML.

Dans les manuels de la formation, ils seront indiqués de la manière suivantes **element**. Chacun d'eux (*ou presque*) est composé, dans le langage HTML, d'une balise d'ouverture et une balise de fermeture s'écrivant sous la forme suivante :

`<element>...</element>`.

On appelle contenu d'un élément HTML, tout ce qui se trouve entre ces deux balises. Chaque élément possède une caractéristique (*mise en gras, augmentation de la taille de la fonte...*) dont son contenu sera affecté.

Par exemple l'élément **b**, défini par les deux balises `...`, permet de mettre en gras du texte.

- Il est très fréquent de voir un élément contenant un autre élément. Il est toujours bon de comprendre qui est contenu par qui pour arriver à réussir une mise en page correcte.

Voici la présentation du code HTML d'une page HTML ne contenant que le mot “Bonjour”. Vous pouvez repérer la présence de nombreux éléments HTML. Aucune étude ne sera faite dans cette partie de la formation sur le langage HTML :

Vous pouvez vous amuser à repérer le contenu des éléments suivant **html**, **head**, **title** et **body**.

```

1. <!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01//EN" "http://www.w3.org/TR/html4/strict.dtd">
2. <html>
3. <head>
4.   <meta content="text/html; charset=ISO-8859-1"
5.     http-equiv="content-type">
6.   <title></title>
7. </head>
8. <body>
9.   bonjour
10. </body>
11. </html>

```

Figure 1 : “Présentation du code HTML d’une page”

II. Présentation de KompoZer :

L’image (Figure 2) donne un aperçu de la fenêtre principale de KompoZer :

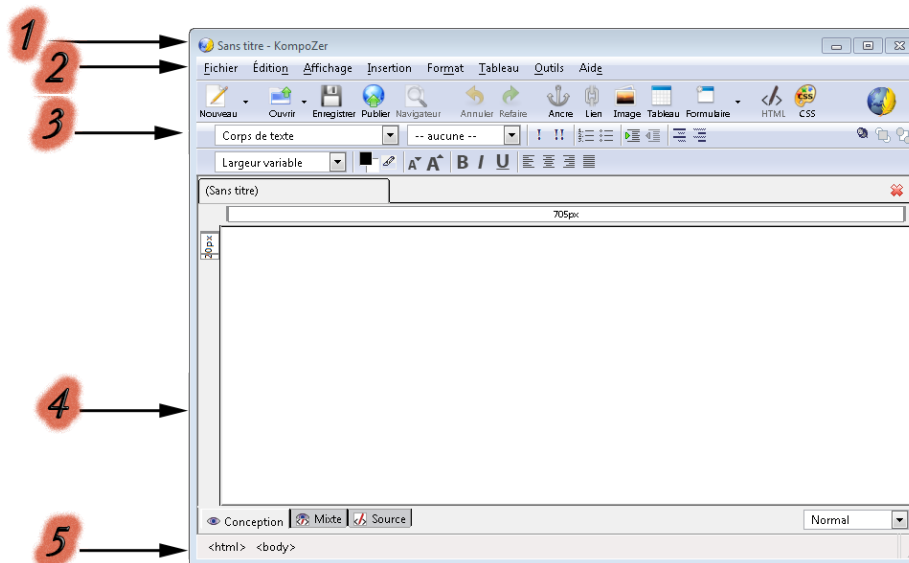


Figure 2 : “Fenêtre principale de composer”

Voici les différentes parties de la fenêtre de KompoZer :

1. **barre des titres** : elle affiche le nom du fichier courant, ou le titre de la page s’il est défini.
2. **barre des menus** : elle permet d’accéder à l’ensemble des commandes de KompoZer.
3. **barre d’outils** : ces boutons permettent un accès rapide vers les commandes les plus utilisées.
4. **espace de travail** : dans cet espace, KompoZer affiche la page en cours d’édition.
5. **barre d’état** : très utile pour affiner sa présentation, elle indique l’élément HTML sélectionné. La compréhension de cet affichage est important pour comprendre la structure de la page Web éditée.

Au dessus de la barre d’état, KompoZer propose trois manières de présenter la page en cours d’édition (voir figure (Figure 3)).



Figure 3 : “Options d’affichage”

- le mode “*Conception*” : la page est affichée comme avec un navigateur.

- le mode “*Mixte*” : la fenêtre principale de KompoZer se partage en deux : la partie supérieure affiche la page et la partie inférieure affiche le code source de celle-ci. Ce mode est intéressant lorsque, avec une légère connaissance du HTML, on souhaite déboguer une page s’affichant incorrectement.
- le mode “*Source*” affiche le code HTML source de la page courante.

On utilisera essentiellement le mode normal pour construire nos pages Web.

III. Première page :

Dans ce paragraphe, nous allons voir les différentes commandes de KompoZer permettant de formater dans un premier temps notre page Internet (*au sens de la mise en place des éléments*).

Le menu **Format** contient l’ensemble des commandes présentées ici. Pour mettre en forme votre page, KompoZer utilise deux technologies :

- La plus ancienne est l’utilisation d’éléments HTML ; chaque élément affecte l’affichage de son contenu. Tout texte placé dans l’élément **i** s’affiche avec une police de caractère en italique.
- Les feuilles de styles CSS, plus couramment utilisées, est un langage informatique (*de description*) permettant d’affecter des styles visuels aux éléments de notre page.

A. Formatage du texte :

Après avoir sélectionné une partie du texte de votre page Web, vous pouvez utiliser les commandes suivantes pour modifier la mise en forme du texte :

- Le changement de police s’effectue à l’aide de la commande :

Format ~> **Police** ▶

Aucune difficulté, cela se passe comme dans les autres logiciels de bureautique.

- En HTML, la taille de la police se mesure de 1 à 7. Le menu ci-contre accessible via :

Format ~> **Taille** ▶

affiche les commandes permettant d’augmenter ou réduire la taille du texte (*voir ci-contre*).

Plus petite	Ctrl+-
Plus grande	Ctrl++

Très petite	
Petite	
• Moyenne	
Grande	
Très grande	
Extra-grande	

- Les deux premiers choix permettent d’augmenter ou de baisser d’une unité la taille du texte sélectionné. Elles utilisent les éléments HTML **big** et **small**).
- Les cinq autres choix modifient la taille de la sélection relativement à la taille courante de la page, à l’aide de l’élément **font**.

Actuellement, ces éléments HTML ne sont plus utilisés ; ils sont remplacés par l'utilisation des feuilles de styles CSS.

- Voici pour les styles **Format ~> style de texte ▶**

⇒ La première partie est l'ensemble des styles communément utilisés.

La “*largeur fixe*” implique un changement de police où l'ensemble des caractères ont une largeur fixe.

La plupart de ces commandes réalisent leurs formatages à l'aide des feuilles de style.

⇒ La seconde partie des commandes rassemble un ensemble de commandes faisant intervenir des anciens éléments HTML, actuellement peu utilisés, permettant d'harmoniser la présentation d'un document (*même formatage, par exemple, pour les abréviations d'un document*).

Gras	Ctrl+B
Italique	Ctrl+I
Souligné	Ctrl+U
Barré	
Exposant	
Indice	
Largeur fixe	
Insécable	
<hr/>	
Mise en évidence	
Mise en évidence prononcée	
Citation	
Abréviation	
Acronyme	
Code	
Exemple de sortie d'un programme	
Variable	

B. Couleurs du texte :

Après avoir sélectionné la partie du texte à colorier, utilisons la commande :

Format ~> Couleur du texte...

La boîte (Figure 4) s'affiche et permet de sélectionner la couleur à utiliser :

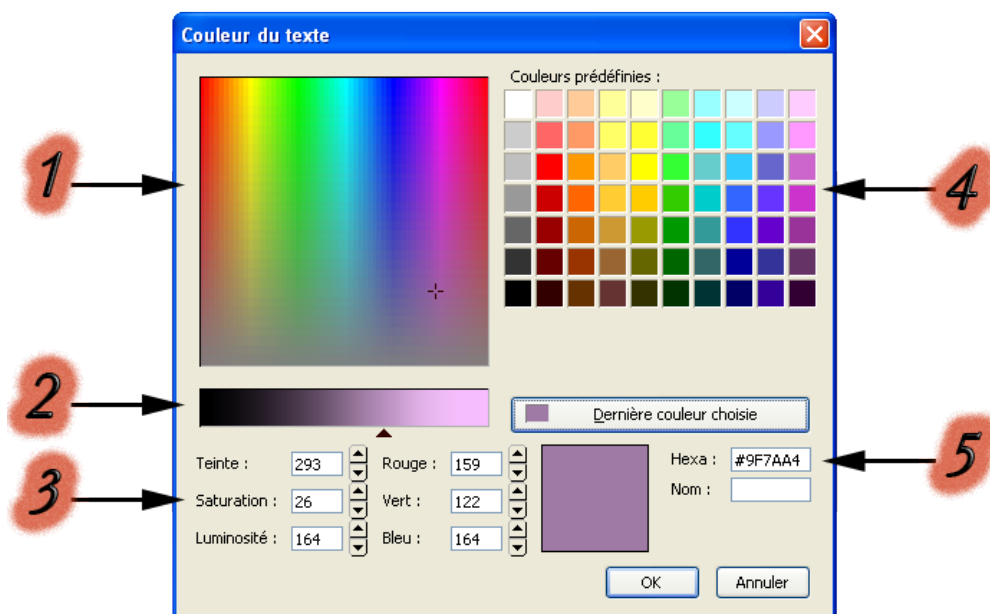


Figure 4 : “Tablette des couleurs”

Vous choisirez rapidement une couleur dans le cadre des couleurs prédéfinies 4.

Pour expliquer les parties 3. et 5. de cette boîte à dialogue, il est nécessaire d'exposer la conception des couleurs dans le monde informatique :

En informatique, tout n'est que nombre, une couleur peut être définie par :

- le modèle RGB représentant une couleur par l'addition de ses trois composantes *rouge*, *vert* et *bleu*. Les valeurs prises par chacune d'elle est comprise entre 0 et 255.
- le modèle TSL (*teinte*, *saturation*, *luminosité* - *HSB en anglais*) permet également de représenter une couleur (*il ne sera pas étudié ici*)

Ainsi, la partie **3.** indique les différentes valeurs prises par ces composantes pour la couleur sélectionnée.

La partie **5.** expose l'expression de cette couleur dans le modèle RGB à l'aide d'un nombre en base hexadécimal : Les deux premiers caractères représentent le *rouge*, les deux suivant le *vert*, les deux derniers le *bleu*.



Voici une petite explication de l'écriture d'un nombre en base hexadécimel.

En informatique, les données sont représentées en mode binaire (*que des 0 et des 1 appelés bits*). Une succession de 4 bits représente un nombre ayant une valeur de 0 à 15 : on représentera un tel nombre par un chiffre hexadécimal représenté par les 16 caractères :

0 ; 1 ; 2 ; 3 ; ... ; 9 ; A ; B ; C ; D ; E ; F

Un octet est une succession de 8 bits : il est donc représenté par 2 chiffres hexadécimal.

La valeur représentée par un octet a une valeur comprise entre 0 et 255.

Dans le modèle RGB de couleurs utilisés dans le standart Web, chaque composante est codée sur 1 octet : ce modèle est le RGB-24bits (*3 composantes × 8 bits*)

C. Couleurs et fond de page :

On accède à la gestion de la couleur par défaut de l'ensemble du texte de la page et des liens à travers la commande :

Format ↪ *Couleurs et fond de page...*

La boîte à dialogue ouverte par cette commande permet également d'incruster une image en fond de page.

Après avoir actionné cette commande, la boîte à dialogue (*Figure 5*) s'affiche :

Voici les différentes options proposées :

1. Cette case représente la couleur par défaut pour l'ensemble du texte de la page.
2. Les liens occupent un status spécial parmi l'ensemble du texte de votre page. Trois états, associés à trois couleurs, caractérisent un lien : le lien normal, le lien déjà visité et le lien lorsque le client est en train de cliquer dessus.
3. Cette case représente la couleur de fond de votre page
4. Ce champ de texte contiendra l'emplacement d'un fichier-image qui sera affichée en fond

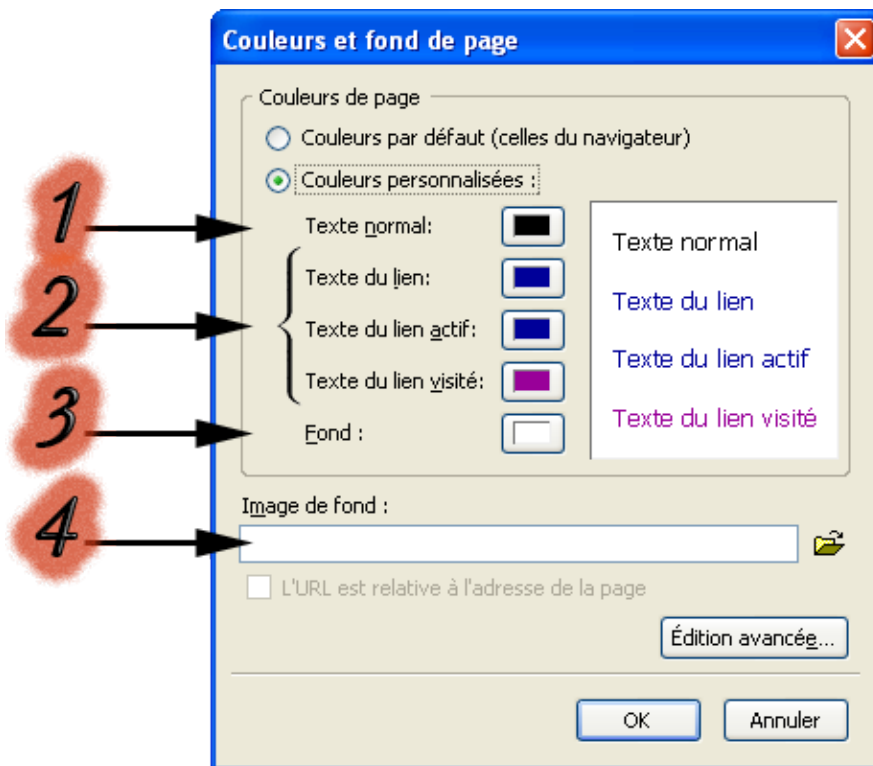


Figure 5 : “Couleurs et fond de page”

de page.

Les trois premières options proposent un bouton représentant la couleur actuellement sélectionnée; en cliquant dessus, la boîte précédemment s’affiche permettant de sélectionner une nouvelle couleur.

⚠ Nous verrons plus loin comment les feuilles de styles permettent de modifier la façon dont l’image sera positionnée en fond de page.

Pour utiliser correctement les images dans un site Web, il est préférable d’avoir une petite connaissance sur les URL relatives et absolues (*voir prochain paragraphe*).

D. Les paragraphes :

Les commandes relatives à la structure de paragraphes dans une page Web (*présentées ci-contre*) se trouve à l’emplacement suivant :

Format ↔ Paragraphe ▶

Elles utilisent des éléments HTML pour effectuer créer des blocs horizontaux représentant les paragraphes de notre page Web :

Trois groupes de commandes se distinctent :

- Par défaut, KompoZer se trouve en mode “*en corps de texte*”; KompoZer insère, dans ce mode, directement le texte saisi dans l’élément parent (*généralement **body***).

Lors du passage en mode “*Paragraphe*”, la ligne courante va être insérée à l’intérieur de l’élément HTML **p**; cet élément crée, à l’aide de son contenu, un bloc horizontal et l’espaçant avec les éléments l’entourant donnant l’effet de paragraphes écartés entre eux.


- Corps de texte
- Paragraphe
- Titre 1
- Titre 2
- Titre 3
- Titre 4
- Titre 5
- Titre 6
- Adresse
- Préformaté

Conteneur générique (div)

Lors d'un nouveau passage en mode "en corps de texte", l'élément **p** est retiré et son contenu est reversé dans le **body** (ou l'élément parent).

- Les modes de "Titre 1" à "Préformaté" utilisent des éléments HTML pour apporter un formatage prédéfini à la ligne courante en l'insérant également dans un bloc ; elles permettent d'uniformiser ainsi la présentation des pages. Actuellement, Ils sont délaissés en faveur des feuilles de styles.
- Le mode **conteneur générique (div)** est le plus important, il permet d'inclure la ligne courante dans un bloc sans lui apporter aucun formatage ; son effet est uniquement de créer un bloc horizontal en utilisant la ligne courante.

N'apportant aucun formatage, il est l'outil préféré des feuilles de styles pour apporter à plusieurs lignes et/ou à plusieurs éléments HTML le même formatage ; il est également utile en **JavaScript** pour faire référence à une partie de la page en même temps.

 Voici un petit défaut de KompoZer : si un de ces éléments se trouve en fin de page (ou en début), il est alors difficile de placer le curseur sous cet élément pour ajouter du contenu à la page.

Une des solutions consiste à placer son curseur dans cet élément et de passer en mode d'affichage "Mixte" ; seul l'élément considéré et son contenu est affichée dans la partie du code : il suffit alors de ajouter manuellement le contenu souhaité à la suite de la balise fermante (`</p>`, `</div>`, `</h1>`...) puis de revenir en mode conception.

E. Les listes :

Les listes permettent de présenter une énumération d'éléments.

La commande **Format ~> Liste ▶** permet d'introduire une liste d'énumération dans la page. On accède au sous-menu ci-contre :

- Aucune
- Puces
- Numérotation
- Terme
- Définition
- Propriétés de la liste...

- Les listes ordinales (défini par l'élément **ol**) : chaque élément de la listes est précédé d'un numéro arabe ou romain, ou alors d'une lettre de l'alphabet.
- Les listes non-ordinales (défini par l'élément **ul**) : un symbole (carrés, rond) précède le début de chaque élément de la liste pour le distinguer.

Il est possible de modifier le format de la numérotation et les puces servant à marquer un élément de la liste à partir de la commande :

Format ~> Liste ~> Propriété de la liste...

Les commandes *Terme* et *Définition* sont devenus obsolètes ; elles étaient utilisées pour faciliter le formatage de définitions et autres. Il est bon de rappeler que le réseau Internet ainsi que les pages Web ont été développés dans le milieu scientifique pour la diffusion de documentation

informatique.

F. Retrait et alignement :

Pour mettre en évidence un paragraphe, il est possible de le décaler vers la droite (*appelé retrait positif*), on utilise alors la commande :

Format ~> Augmenter le retrait

Pour ramener un paragraphe vers la gauche, on utilise la commande :

Format ~> Diminuer le retrait

La commande **Format ~> Aligner ►** permet d'aligner à droite, à gauche, centrer ou justifier une ligne de texte à l'intérieur d'un bloc.

Si le mode courant de KompoZer est "en corps de texte" lors de l'utilisation de l'une de ces commandes, le texte de la ligne courante sera placé automatiquement dans un élément **div** : des pointillés rouges viendront signaler l'apparition de l'élément **div**.

G. Propriétés de la page :

La commande **Format ~> Titre et propriété de la page...** permet :

- de donner un titre à votre page ; celui-ci est affiché dans la barre des titres du navigateur et sert également au référencement de la page sur les moteurs de recherche.
- *Auteur, Description, Langue* : ces informations servent essentiellement aux moteurs de recherche pour le référencement de votre page.
- de spécifier l'encodage des caractères choisi par le navigateur pour afficher votre texte (*un lecteur asiatique verra afficher les caractères latins même si son navigateur est configuré pour un autre affichage*).



Un caractère est représenté à l'intérieur de l'ordinateur par un nombre. Il existe donc pour chaque alphabet une table de correspondance entre un caractère et un nombre. Cette table de correspondance n'est pas la même pour une langue européenne et une langue asiatique. Ce choix s'appelle le type d'encodage de caractères.

IV. Structure d'un site Internet :

A. Introduction :

Les pages Web sont de simples fichiers texte contenant du texte et le code HTML donnant le formatage à votre page. Généralement, ces fichiers portent l'extension ".html" ou ".htm" ; ils

peuvent être ouvert avec un simple éditeur de texte : le code, comme apparaissant en affichage “*Source*” apparaîtra.

Les images, les sons... sont, en fait, des fichiers externes aux pages Webs !

Ainsi, un site Internet est composé des fichiers contenant le code HTML (*fichier texte d’extension .html*) et d’autres fichiers qui seront incorporés dans la page tel que les images, les vidéo, les animations...

Il est préférable d’organiser toujours un site à l’aide de dossiers et sous-dossiers. Mettre tous les fichiers dans un même endroit gêne lorsque votre site s’agrandit ou s’il doit être mis à jour. Plusieurs structures d’organisation sont possibles :

- Disposer les images dans un dossier, les animations dans un autre ...
- Chaque page et les fichiers associés sont disposés dans un dossier séparé.

A vous d’imaginer votre structure, mais sachez que ceci influencera la facilité avec laquelle vous allez mettre votre site à jour.

B. Du local au distant :

Pour qu’un site soit accessible tout le temps et de partout dans le monde, il est nécessaire de trouver un hébergeur qui accueillera votre site et le rendra accessible au monde entier.

Il existe deux grands temps dans la construction d’un site internet :

- La partie la plus longue de développement du site sur un ordinateur personnel (*présent physiquement*) : on dira qu’on développe notre site en **local**.
- Ensuite, il faudra trouver un hébergeur et y transférer l’ensemble des fichiers caractérisant notre site ; on parlera alors de serveur **distant**. Ce transfert de fichiers s’effectue à l’aide d’un logiciel FTP.

Votre hébergeur vous donnera une adresse du type suivant pour visiter votre site :

`http://hebergeur.com/monIdentifiant`

Voici le problème que nous allons apprendre à gérer dans ce paragraphe. Supposons que l’image `im.jpg` fasse partie de votre site ; voici sa localisation sur votre disque dur :

`C:\Documents and Settings\maSession\Mes documents\monSite\im.jpg`

alors que vous ne saurez pas par avance l’emplacement que votre hébergeur vous a réservé ; par exemple :

`C:\hebergement\monIdentifiant\im.jpg`

Nous allons voir dans ce paragraphe comment gérer cet inconvénient lors de la mise en ligne de notre document.

C. Racine du site :

- Localement, pour construire un site, on crée un répertoire dédié au site : ce dossier contiendra tous les fichiers nécessaires au site.
- L'hébergeur vous offrira également un dossier dans lequel vous placerez tous vos fichiers (*page Web, images, sous-dossiers...*)

Dans les deux cas (*local ou distant*), on appelle ce dossier la “*racine du site*”.

D. URL relative et absolue :

On appelle URL (*Uniform Resource Locator*) d'un fichier ou d'un site l'adresse permettant d'accéder à un fichier. La forme d'une URL dépend du système d'exploitation ou de communication utilisé ; voici des exemples d'adresses permettant de localiser le fichier `dessin.png` :

- Avec un système d'exploitation Windows :
`C:\Program Files\EasyPHP1-8\www\dessin.png`
- Avec un système d'exploitation Linux :
`/utilisateurs/home/eleve/publicHtml/dessin.png`
- Sur le réseau Internet :
`http://www.monsite.fr/dossier/dessin.png`

Ces adresses s'appellent des **URL absolues** : elles indiquent l'adresse complète du fichier.

Voici l'exemple d'une URL absolue d'un serveur Web :

<u>http://</u>	<u>www.yahoo.com</u>	/	<u>france/actualite/aujourd'hui.html</u>
Protocole	Hôte + Nom de domaine		Chemin d'accès du fichier

- Le protocole indique la manière dont l'information est envoyée via internet. On utilisera principalement deux protocoles : le HTTP pour recevoir et visualiser des pages Web dans un navigateur et le FTP pour transférer des fichiers via Internet.
- Le nom de domaine identifie le serveur contenant l'information recherchée ; l'hôte permet d'identifier le site (*dans le cas où le serveur héberge plusieurs sites*).
- Le chemin d'accès est le chemin à parcourir en partant de la racine du site jusqu'au fichier demandé (*ici aujourd'hui.html*).

Voici une étude de cas illustrant le problème d'adressage se posant lors du passage du local au distant :

Supposons l'existence d'une image `dessin.png` utilisée par une page Web, supposons que cette image a pour URL :

`C:\site\dessin.png`

Mais une fois chez l'hébergeur, elle sera accessible par une adresse de la forme :

`http://monSite/dessin.png`

L'élément HTML `img` permet l'affichage des images dans une page Web ; quel adresse doit-on lui fournir afin qu'il puisse afficher cette image ?

- ⇒ Si on transmet la première, l'image ne s'affichera localement mais pas lorsqu'un client ira chez l'hébergeur pour consulter votre page.
- ⇒ Si on lui soumet la seconde, localement, l'image ne s'affichera pas.

Les **URL relatives** résolvent ce problème : on indique uniquement le chemin à parcourir du fichier courant jusqu'au fichier cible. La structure du site étant la même localement que sur l'hébergeur distant, le chemin de parcour restera le même : l'image pourra s'afficher normalement.

Pour écrire une URL relative, on utilise les deux symboles suivant :

- Le point “.” représente le dossier courant (*ce symbole est en fait facultatif*).
- Le double point “..” représente le dossier parent du dossier courant (*le dossier “juste au dessus”*).

La figure suivante (Figure 6) donne l'exemple d'une arborescence de fichiers :

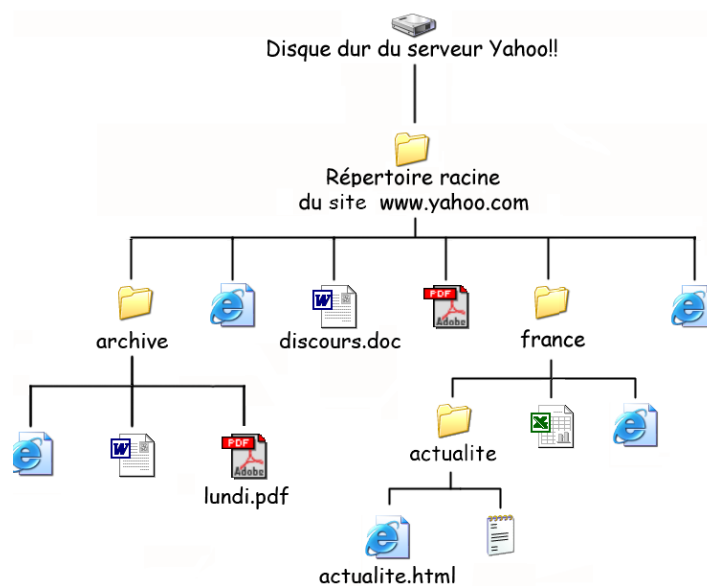


Figure 6 : “Exemple d'arborescence de fichiers”

Cette représentation, d'un arbre renversé, doit vous faire comprendre ce que signifie “descendre dans un répertoire” et “remonter dans le dossier parent”. Nous supposons ici qu'on étudie le site yahoo.com ; le dossier racine du site www.yahoo.com est représenté sur le graphique ci-dessus. Voici les URL absolus de certains fichiers sur le serveur Web de yahoo.com :

- Le fichier Word “discours.doc” :
`http://www.yahoo.com/discours.doc`
- La page HTML “actualite.html” :
`http://www.yahoo.com/france/actualite/actualite.html`
- Le fichier PDF “lundi.pdf” :
`http://www.yahoo.com/archive/lundi.pdf`

Voici quelques exemples d'URL relatives tirées cette étude de cas. Pour définir une URL relative, il faut spécifier le fichier courant faisant référence à un fichier cible : l'URL relative est la description du chemin de l'un vers l'autre.

- De “discours.doc” vers “actualite.html” :

Le fichier “discours.doc” se trouve à la racine du site, pour accéder au fichier “actualite.html”, il faut descendre dans le dossier “france”, descendre encore dans “actualité” pour accéder au fichier “actualite.html”. Voici l'URL relative de “discours.doc” vers “actualite.html” :

```
./france/actualite/actualite.html
```

- de “discours.doc” vers “actualite.html” :


Il faut remonter de deux dossiers pour accéder au fichier Word :

```
../../discours.doc
```

- De “actualite.html” relativement à “lundi.pdf” :

```
../france/actualite/actualite.html
```

KompoZer gère lui-même les URL relatives, ainsi il peut paraître superflue de maîtriser ces notions mais lors de comportements inattendus apparaissent (*non-affichage d'une image, absence du formatage des styles CSS*) il est important de vérifier l'ensemble des URL relatives de manière manuelle.

 Généralement, les URL absolues ne sont utilisés que pour indiquer la direction de fichiers se trouvant à l'extérieur du site en construction.

V. Première approche des éléments HTML :

A. Introduction :

Nous avons déjà vu qu'une page est constituée d'éléments HTML permettant d'indiquer au navigateur le formatage à appliquer à cette page. La plupart des éléments sont composés d'une balise d'ouverture et de fermeture :

L'élément **p**, déclarant un nouveau paragraphe dans la page, est défini par les balises :

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

Les éléments **img** et **br**, insérant respectivement une image ou un saut de ligne, ne sont définis que par leur balise d'ouverture.

Les éléments HTML peuvent contenir d'autres éléments HTML : les éléments HTML s'imbriquent les uns dans les autres. Il est très important de connaître l'imbrication de ces éléments afin de maîtriser l'affichage d'une page.

Voici quelques exemples d'affichages de la barre d'état commentés :

1. `<body>` `<div>`

2. `<body>` `<table>` `<tbody>` `<tr>` `<td>`

1. Le curseur est placé actuellement dans un conteneur `div` placé directement dans l'élément `body`.

2. Ici, le curseur est placé dans une cellule d'un tableau ce qui explique la succession des imbrications :

- `table` définit un tableau
- `tbody` définit le corps du tableau.
- `tr` définit une ligne du tableau
- `td` définit une cellule de la ligne courante



Voici quelques remarque sur cette barre d'état :

- ➔ En cliquant sur un élément de la barre d'état, KompoZer sélectionne automatiquement le contenu de cet élément.
- ➔ L'élément `body` est présent dans toutes les pages Web ; tous les éléments HTML affichés à l'écran doivent se placer dans cet élément. `body` représente le corps du document.
- ➔ On remarque dans l'exemple 2. que certaines balises, représentant les images HTML, sont présentés en italique ; ceci indique que ces éléments sont porteurs de style CSS. Ceux-ci seront étudiés plus loin mais on remarque encore l'importance de l'affichage présent dans la barre d'état.

B. L'affichage "Balises HTML" :

La commande `Affichage ~> Balises HTML`, présente aussi dans le menu déroulant en bas à droite de la fenêtre de KompoZer, affiche à l'écran la plupart des éléments HTML présents dans la page.

Cette affichage déforme le positionnement des éléments dans la page mais présente l'avantage de visualiser rapidement l'emplacement des éléments HTML dans la page mais seul la barre d'état vous permettra de connaître précisément l'imbrication des éléments.

La capture d'écran (Figure 7) montre la présence de deux éléments :

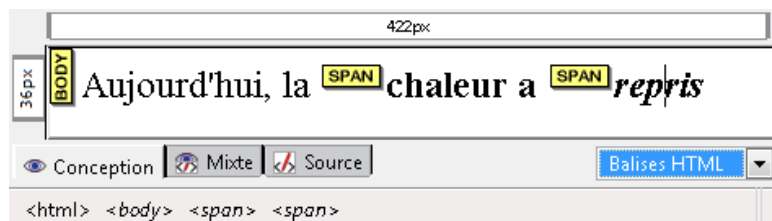


Figure 7 : "Présence des éléments HTML"

On aurait pû penser que ces deux éléments `span` se trouve côte à côte ; or, le curseur se trouve dans le second `span` et l'observation de la barre d'état montre que le second `span` est inclu dans le premier. On aurait également pû appuyer sur le premier `span` pour voir si le second

aurait également été sélectionné par KompoZer.

C. Les attributs :

Chaque élément HTML admet un certain nombre de paramètres permettant de contrôler son comportement, son action sur la page. Dans ce paragraphe, nous ne rentrerons pas dans le détail des attributs qui sont davantage utiles lors de la programmation HTML.

L'élément `div` admet l'attribut *align* permettant de définir l'alignement du texte qui le contient ; avec pour valeur *center*, le contenu de `div` sera centré sur la largeur de la page courante. Dans le langage HTML, on incorpore cet attribut dans la balise d'ouverture :

```
<div align="center">...</div>
```

KompoZer gère lui-même les attributs. Cette utilisation est transparente pour l'utilisateur : pas besoin de rentrer dans le code HTML. Mais KompoZer va nous donner des indications sur la présence de certains attributs (*notamment ceux liés aux feuilles de styles*).

Considérons un élément `div` directement contenu dans `body` :

```
<body> <div>
```

En appliquant un style CSS sur le `div` (*par exemple changement d'alignement*), KompoZer va utiliser les feuilles de styles et cette utilisation sera indiquée par un changement de police dans la barre d'état :

```
<body> <div>
```

 Une balise représentée en italique indique que l'élément utilise les styles internes CSS

Comme nous le verrons dans le chapitre sur les feuilles de styles dans un prochain paragraphe, il est possible d'affecter un style à un élément via un fichier externe. Pour désigner l'élément ciblé, on utilisera l'attribut *id* ; la barre d'état affichera la balise de l'élément sous la forme suivante :

```
<body> <div id="maPhoto">
```

Ici, l'élément `div` a pour identifiant "*maPhoto*".

Si on souhaite qu'un certain nombre d'éléments d'une même page reçoivent le même formatage, on utilisera l'attribut *class* :

```
<body> <div class="chapitre">
```

D'où l'importance d'information que contient la barre d'état.

On peut également tout combiner pour voir afficher la barre d'état de KompoZer sous la forme :

```
<body> <div id="maPhoto" class="chapitre">
```

Pour en savoir davantage sur les règles de styles, reporter en fin de cette formation sur le

VI. Liste des éléments HTML :

Dans ce paragraphe, sont présentés les éléments HTML rencontrés le plus fréquemment dans KompoZer.

A. `<body>...</body>` :

L'élément `body` contient l'ensemble des caractères et des éléments HTML composant l'affichage d'une page Web : il définit le corps de la page.

C'est à travers cet élément qu'on définit les valeurs par défaut de la taille du texte, la fonte utilisée, la couleur d'arrière plan de la page...

Certains de ces paramétrages sont accessibles via :

Format ↪ *Couleurs et fond de la page*

les autres se feront par l'intervention des feuilles de styles CSS.

B. `...` :

Cet élément est un conteneur : son contenu ne subit aucune modification ni aucun affichage particulier ; les feuilles de styles sont alors utilisées pour apporter un même formatage à l'ensemble de son contenu ; un tel conteneur peut également permettre à un script **JavaScript** de contrôler en même temps l'ensemble de son contenu.

C'est un élément dit "*en ligne*", il ne peut contenir aucun saut de lignes, ni aucun élément déclenchant un saut de ligne tel que `p`, `div`, `table`...

KompoZer crée un tel élément lorsqu'on applique un changement de taille de caractère, de polices [...] à une partie du texte.

 Voici quelques astuces liées à l'élément `span` :

- Pour supprimer le formatage d'une partie du texte, sélectionnez-la, puis actionnez la commande :

Format ↪ *Supprimer tous les styles de texte*

Si vous avez sélectionné l'ensemble du contenu d'un élément `span`, celui-ci disparaîtra.

- Pour supprimer un élément `span` (ou en fait n'importe quel élément HTML), mais pas son contenu, effectuez un clic droit sur la balise représentant l'élément considéré puis actionnez la commande :

menu contextuel ↪ *Supprimer la balise*

C. `<div>...</div>` :

L'élément `div` est également un conteneur ; il n'apporte aucun formatage à son contenu mais permet de gérer l'ensemble du contenu comme un seul élément (*utile pour les feuilles de styles ou les scripts*) mais il est de type "bloc" : il peut contenir des sauts de lignes et tout type d'élément HTML.

KompoZer fait apparaître cette balise dès l'utilisation de certaines commandes commandes tel qu'un changement d'alignement, un retrait du texte... Pour insérer manuellement un élément `div` dans la page, on utilise la commande suivante :

Format ~> *Paragraphe* ~> *Conteneur générique (div)*

⚠ L'élément `div` peut contenir des éléments `span` et `div`.

L'élément `...`, quant à lui, ne peut contenir de saut de ligne : ainsi il ne peut contenir d'élément `div`.

D. <a>... :

L'existence de liens est la caractéristique principale distinguant une page Internet d'un simple document texte : elle donne accès au client de naviguer de pages en pages en lui laissant accès à une zone cliquable.

L'élément définissant les liens est `a`.

⚠ Pour utiliser cet élément, il est préférable d'avoir un idée de la structure (*emplacement de la racine, dossiers et sous-dossier*) ; en effet, lors de l'insertion d'un lien, KompoZer insère dans l'élément `a` l'adresse relative du fichier courant vers le fichier cible ; si la structure de votre site change, l'URL sera invalide et il faudra modifier l'adresse de destination afin que le lien redevienne valide.

Pour créer un lien texte, commencez à sélectionner la partie du texte le représentant. Puis, utilisez la commande :

Insertion ~> *Lien*

La boîte à dialogue (*Figure 8*) des propriétés du lien s'ouvre :

1. Ce champ donne un aperçu du texte sélectionné et qui servira de zone cliquable pour le lien.
2. Ce champ texte doit contenir l'adresse du fichier cible :
 - Pour un fichier se trouvant sur un autre site, il faut saisir l'URL absolue de la page cible : `http://.....`
 - Pour un fichier appartenant au site, il faut utiliser l'URL relative de la page courante vers la page cible. On peut la saisir au clavier ou laisser KompoZer faire en utilisant le bouton en forme de dossier à droite de ce champ.
3. Si le lien est destiné à envoyer un mail, cochez cette case. Le client, en cliquant sur ce lien, verra son logiciel de messagerie par défaut (*généralement Outlook Express*) s'ouvrir

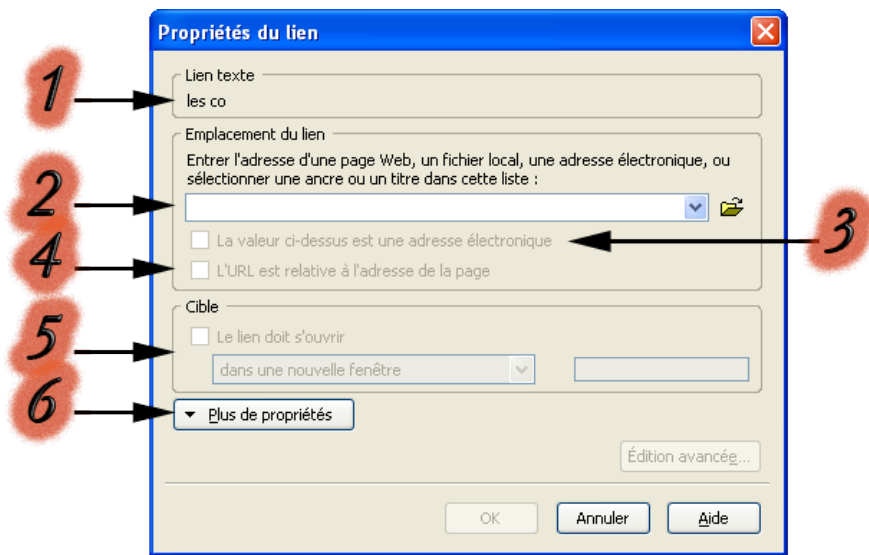


Figure 8 : “Propriété d’un lien”

proposant un nouveau message destiné à l’adresse saisie en 2.

4. Cette option indique que l’URL rentrée est relative. En pratique, lorsqu’on utilisera le bouton à droite du 2., KompoZer cochera automatiquement cette case pour nous.
5. par défaut, en cliquant sur un lien, la page cible remplacera la page courante dans la même fenêtre du navigateur, donnant l’impression au client de naviguer de page en page. En cochant cette case et en choisissant la seconde option “*Dans une nouvelle fenêtre*”, on modifie en conséquence ce comportement ; les autres options sont actuellement désuètes (*voir remarque suivante*).

⚠ Une ancienne conception de site était de partager la fenêtre d’affichage en plusieurs pages appelées des cadres (*frames en anglais*). Cette technique, introduite dans la fin des années 90, permet lors de la redirection du client de ne recharger qu’une partie de la page (*gagnant ainsi du temps et de la bande passante*).

Cette technique a été très décriée et à tendance actuellement à disparaître ; la technologie AJAX permet d’obtenir un bien meilleur rendu lorsqu’on souhaite changer dynamiquement une partie de la page (*mais ceci est un autre problème ; intégration conjointe de PHP et de Javascript*)

On préfère actuellement une construction avec des conteneurs `div` positionnés à l’aide des feuilles de styles.

E. `` :

Une image s’insère dans une page Web à l’aide de l’élément `img`. On insère une image avec KompoZer à l’aide de la commande :

Insertion ↔ Image...

La boîte à dialogue (Figure) suivante s’ouvrira :

1. Ce champ indique l’URL (*absolue ou relative*) localisant l’image à insérer. Le bouton à

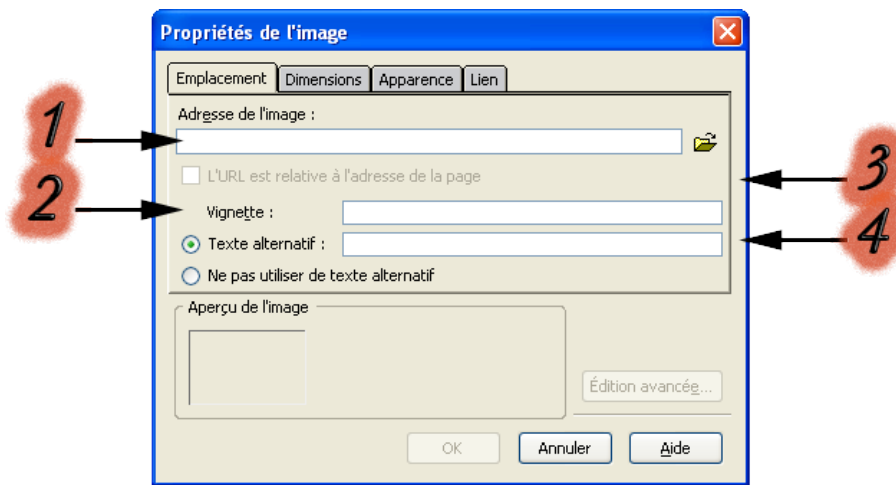


Figure 9 : “Propriété d’une image”

droite du champ (*en forme de dossier*) permet de parcourir le disque dur local à la recherche de l’image.

⚠ Pour ne pas avoir de problème lors du transfert du site chez l’hébergeur, l’image doit appartenir à la racine du site ou un de ses sous-dossiers.

2. La vignette est le petit bandeau jaune contenant un texte apparaissant lorsque le client laisse son curseur immobile au dessus de l’image ; vous pouvez saisir un descriptif de l’image, une date. . .
3. Comme pour les liens, cette option précise si l’adresse de l’image (*son URL*) est relative ou absolue. En parcourant la racine du site, KompoZer insère des URL relatives : il cochera automatiquement cette case.
4. Certains navigateurs, pour les mal-voyants notamment, affichent un texte à la place de l’image : remplissez ici un court descriptif de l’image ou cochez la case “*Ne pas utiliser de texte alternatif*”.

Nous venons de passer en revue l’onglet “*Emplacement*”. Voici un descriptif rapide des autres onglets de cette boîte à dialogue :

- *Dimensions* : cet onglet affiche la taille de l’image que vous pouvez modifier ; attention, en augmentant la taille de l’image, à un effet de “*pixellisation*” ; attention de ne pas déformer l’image en décochant l’option “*Conserver les proportions*”.
- *Apparence* : cet onglet permet d’affiner l’affichage de l’image en :
 - ⇒ Plaçant un espace horizontal (*à droite et à gauche*) autour de l’image ainsi que vertical.
 - ⇒ Plaçant une bordure autour de l’image pour l’encadrer
 - ⇒ “*Placement du texte*” : cette option rend l’image flottante ; le texte va se placer à gauche ou droite de l’image en prenant plusieurs lignes.
 - ⇒ On peut découper une image en différentes parties cliquables : ce sont les “*Image Map*”. KompoZer permet uniquement de supprimer une telle information dans le cas d’une importation de pages Webs. Les Images Maps ne seront pas développées ici.

- **Lien** : cet onglet permet de rendre l'image cliquable ; cette partie s'apparente à celle rencontrée pour les liens (*reportez vous au paragraphe précédent*).

⚠ Une image numérisée est un ensemble de points de couleurs, appelés les pixels ; la **dimension d'une image** est le nombre de ces points, horizontalement (*par ligne*) et verticalement (*par colonne*), constituant l'image.

La **taille de l'image**, elle, dépendra de la résolution avec laquelle, on affiche ou on imprime l'image : c'est à dire le nombre de pixels placés horizontalement sur une longueur d'un pouce ; l'unité de mesure de la résolution est le DPI (*en anglais dot per inch*).

On peut dire que la résolution est le "concentration" de pixel par unité de longueur.

F. Les tableaux :

La construction de tableaux avec KompoZer sera développée dans un prochain paragraphe ; voici néanmoins les différents éléments HTML rencontrés lors de la présence d'un tableau :

- **table** : définit la zone de déclaration d'un tableau
- **tr** : définit une ligne du tableau.
- **td** : définit une cellule du tableau.

G. Les formulaires :

Les formulaires ne seront pas étudiés ici. Car pour que les données recueillies par un formulaire soit utilisable, il est nécessaire d'utiliser un langage de programmation :

- **JavaScript** peut permettre, mais avec de lourdes peines, de traiter les données ici d'un formulaire mais sans grand succès.
- Il faudra utiliser un langage de programmation côté serveur (*implémenté sur le serveur, genre PHP ou ASP*) afin de traiter efficacement les données renvoyées par un formulaire.

KompoZer prend en charge l'ensemble des éléments associés à une page Web. On trouve l'ensemble des contrôles disponibles pour un formulaire à l'emplacement suivant :

Insertion ↔ Formulaire ►

Voici un descriptif court des commandes proposées :

- "Définir un formulaire. . ." : cette commande permet de déclarer un formulaire dans la page. On attribue un nom au formulaire et notamment la page Web qui recevra et traitement les informations contenues par le formulaire. Un formulaire est défini par l'élément **form**.
- "Champ de formulaire. . ." : cette commande permet d'insérer un élément **input** ; cet élément définit la plupart des contrôle d'un formulaire. Il peut être une case à cocher, un champ texte à saisir, le bouton de soumission du formulaire. . . C'est un attribut HTML de cet élément qui définit la nature du contrôle.

- “Image de formulaire” : cette commande permet d’insérer une image cliquable (*définie également par l’élément `input`*).
- “Zone de texte...” : cette commande insère un champ de texte sur plusieurs lignes. Elle est définie par l’élément `textarea`.
- “Liste de sélection...” : cette commande crée une liste de sélection généralement présentée sous la forme d’un menu déroulant. La liste est défini par l’élément `select`.

VII. Positionnement des éléments HTML :

A. Introduction :

Normalement, un élément HTML est positionné dans le flux courant du texte : cela signifie que les éléments (*caractères, images...*) sont affichés un par un par le navigateur pour former les lignes ; le navigateur se charge d’insérer automatiquement, lorsque toute la largeur de la page est utilisée, des retours à la ligne.

Les feuilles de styles ont apportées de nombreux effets de formatage améliorant l’apparence des pages Internet. Elles ont permis également de modifier ce comportement : on peut définir précisément l’emplacement d’un élément le sortant ainsi du flux courant du texte. On parlera alors de **positionnement absolu** : on choisira l’emplacement exact de notre élément et ses dimensions ne seront plus assujettis à la taille d’affichage du client (*de la fenêtre du navigateur*).

Bien qu’il soit possible de modifier le placement de tous les éléments, on préfère habituellement travailler sur les éléments `div`. La fenêtre (*Figure 10*) de KompoZer où sont marquées quelques indications utiles pour modifier le positionnement d’un élément de la page :

B. Le positionnement et les dimensions :

Voici la procédure pour positionner un élément dans la page en positionnement absolu :

- Sélectionner un élément à l’aide de la barre d’état **4.** Ici, un élément `div` ;
- Cliquer sur le bouton **1.** : l’élément passe en positionnement absolu. Il apparaît dans un rectangle entouré par les points cardinaux (*voir (Figure 11)*).
- Les points cardinaux **8.** du rectangle permettent de redimensionner l’élément. Ses dimensions peuvent être observées en **6.**
- Le point d’ancrage **7.** permet de déplacer, à l’aide du curseur, l’élément dans la page. Ce déplacement se mesure habituellement à partir du point supérieur gauche de la page : Les ressorts en **5.** symbolise l’espace séparant le coin supérieur gauche de notre élément à celui de la page.

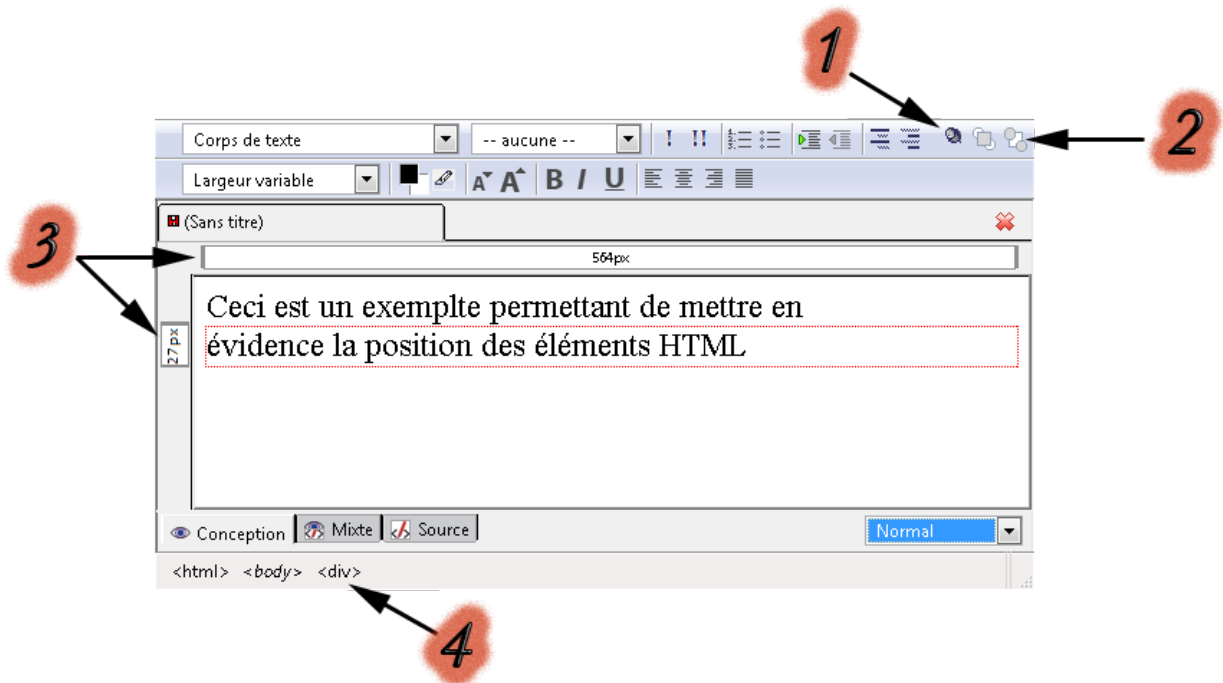


Figure 10 : “Elements pour le positionnement absolu”

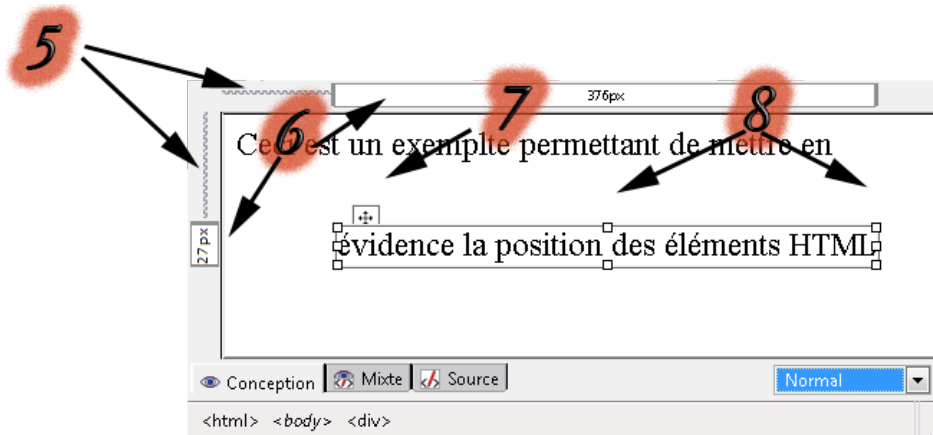


Figure 11 : “Apparition d’un élément en positionnement absolu”

Pour supprimer le positionnement absolu d’un élément, il faut, après l’avoir sélectionné, appuyer sur le bouton **1** : la punaise est renfoncée dans la page ; le document retrouve sa place normale dans le flux du texte.

Il est toujours préférable lorsqu’on souhaite utiliser un élément en positionnement absolue d’utiliser l’élément **div** et que celui-ci soit placé directement dans l’élément **body**.



KompoZer pose des problèmes pour positionner le curseur en fin de document lorsque celui-ci se termine par un élément **div** ; un commentaire dans ce sens a déjà été traité (voir page 12)

C. La superposition :

Plusieurs éléments en positionnement absolu peuvent se superposer, comme le montre l’image (Figure 12) :

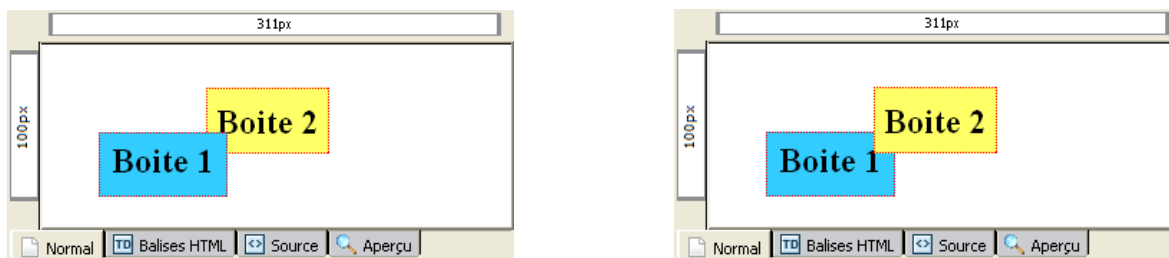


Figure 12 : “Superposition d’éléments en positionnement absolu”

Nous avons déjà vu que ce sont les feuilles de styles qui permettent le positionnement absolu des éléments HTML ; de plus, pour chaque élément, un compteur nommé *z-index* est attribué, dans le cas d’une superposition d’éléments, celui qui a le plus grand *z-index* sera placé au dessus des autres...

On modifie la valeur du *z-index* sélectionnée en cliquant sur les deux boutons indiqués en 2.

VIII. Les tableaux :

L’ensemble des commandes relatives aux tableaux se trouvent dans le menu **Tableau**.

A. Insertion d’un tableau :

On insère un tableau dans une page Web à l’aide d’une des deux commandes ci-dessous :

⇒ **Insertion ~> Tableau...**

⇒ **Tableau ~> Insérer ~> Tableau...**

Ces commandes actionnent l’apparition de la boîte à dialogue ci-dessous permettant de définir les caractéristiques du tableau. Voici les différents onglets proposés par cette boîte à dialogue (Figure 13).



1.

2.

3.

Figure 13 : “Fenêtre de configuration d’un tableau”

Voici le descriptif de chacun de ces onglets :

1. Cet onglet permet de définir très rapidement la taille du tableau (*ici limité à 6 lignes et 6 colonnes*), mais aucune option n’est applicable.
2. Cet onglet permet de saisir au clavier le nombre de lignes et de colonnes composant le tableau. On peut également fixer la largeur totale du tableau en pixels ou la définir en pourcentage relativement à la largeur totale de la page. On définit également la largeur de

la bordure en pixel.

3. Cet onglet permet d'affiner l'affichage des cellules :

- L'alignement horizontal ou vertical par défaut de chaque cellule ; la valeur “*non spécifié*” laissera le navigateur appliquer ses paramètres par défaut.
- Suivant la largeur attribuée au tableau et, de fait, à chaque cellule, le navigateur insère, par défaut, des retours à la ligne lorsque le contenu d'une cellule est trop grand. L'option “*Retour automatique à la ligne*” permet d'annuler ce comportement avec la valeur “*non*”, mais la largeur du tableau risque alors de s'allonger et dépasser la taille d'affichage de la page.
- “*Espacement dans les cellules*” caractérise l'espace laissé entre les cellules (*horizontalement et verticalement*).
- “*Marges dans les cellules*” permet d'éloigner le contenu de la cellule avec la bordure de cette même cellule. Permettant d'aérer le contenu de la cellule.

B. Modification du tableau :

- Les sous-menu suivant et leurs commandes se passent de commentaire. Certaines d'entre elles agissent en fonction de l'emplacement du curseur dans le tableau.

⇒ **Tableau ~> Insérer**

⇒ **Tableau ~> Sélectionner**

⇒ **Tableau ~> Supprimer**

- Plusieurs cellules peuvent être fusionner en une seule cellule à l'aide de la commande :

Tableau ~> Fusionner avec la cellule suivante

Pour annuler la fusion de cellules, sélectionner la cellule fusionnée et utiliser la commande :

Tableau ~> Scinder la cellule

- La couleur de fond d'un tableau ou d'une cellule est déclarée ou modifiée à l'aide de la commande :

Tableau ~> Couleur de fond de tableau ou de cellule...

La boîte à dialogue classique vous permet de sélectionner la couleur. Remarquez l'option se rajoutant en haut de la boîte à dialogue :

Fond pour : Tableau Cellule(s)

précisant si la couleur de fond sera appliquée à l'ensemble du tableau ou à la cellule courante.

- Après création d'un tableau, vous pouvez toujours modifier les caractéristiques du tableau en ouvrant la boîte à dialogue :

Tableau ~> Propriétés du tableau...

IX. Les formulaires :

A. Introduction :

L'usage des formulaires n'est pas étudié dans cette formation. En effet, le principe du formulaire est de laisser le client saisir des valeurs dans un champs texte, cochez des cases... puis envoyer ces informations vers un serveur.

Celui-ci récupère les données, il peut modifier son comportement suivant les paramètres envoyés (*notamment renvoyé tel ou tel page*) ou stocker les données dans une base de donnée. Il faut alors pouvoir coder (*généralement à l'aide des langages PHP et ASP*) ce genre de comportement.

Ces langages de programmation étant plus compliqué, ils ne seront pas étudié dans la formation.

Les éléments HTML relatifs aux formulaires ont été présentés dans le paragraphe "*Liste des éléments HTML*". Dans la suite de ce paragraphe, je n'expose que les principes de base pouvant vous permettre de mieux appréhender certains comportement observés sur votre navigateur.

B. Principe de fonctionnement :

En cliquant sur le bouton de soumission (*nom du bouton validant l'envoi du formulaire*), le client demande l'affichage d'une nouvelle page. Lors de cette envoi, les données saisies par le client sont envoyés en même temps que cette demande au serveur.

Ces données sont transmises au serveur sous la forme de couple (nom du contrôle ; valeur).

On peut observer dans l'URL la transmission des valeurs ayant pour nom *fr*, *ei* et *p* :

```
http://yahoo.com/news?fr=FP-tab-news-t&ei=ISO-8859-1&p=france
```

Ce mode de transfert d'informations au travers l'URL s'appelle la méthode GET.

Il existe également la méthode POST qui cache les données (*mais attention aucun cryptage n'est employé*) dans l'entête de la demande. Ce point est un peu difficile à comprendre et fait parti du protocole HTTP (*qui ne sera pas développé ici*).

La méthode POST est la plus appréciée pour les raisons suivantes :

- La méthode GET impose une taille maximale pour une URL (*généralement 256 caractères*), limitant ainsi le nombre d'information transportée.
- Même si les données ne sont pas cryptées, elles deviennent alors inaccessible à la grande majorité des utilisateurs d'Internet.

X. Les feuilles de styles :

A. Introduction :

Les feuilles de styles en cascades (*CSS*) est un langage de description permettant d'obtenir une meilleure mise en page des éléments HTML. Elles permettent également d'homogénéiser la présentation d'un site en centralisant les règles de styles.

Bien que ce langage soit facile d'apprentissage, KompoZer nous facilite l'utilisation des feuilles de styles au travers de commandes et de boîtes de dialogues associées.

Pour utiliser les feuilles de styles, il est préférable d'introduire le contenu à formater à l'intérieur :

- d'un conteneur `span` pour quelques mots ;
- d'un conteneur `div` pour un paragraphe.

pour y appliquer le formatage voulu ; ces deux éléments ayant l'avantage de n'apporter aucun formatage.

Il existe principalement deux manières d'appliquer un style à un élément HTML :

- On place les propriétés de styles directement dans l'élément (*plus précisément dans la balise d'ouverture*) : on parle alors de “*style interne*”.
- On place les propriétés dans des règles placées à l'extérieur de l'élément (*soit dans l'entête de la page, soit dans un fichier externe*) : on parle alors de “*règle de styles*”.

Ces deux manières sont liées à la présentation des attributs des élément HTML dans la barre d'état (*voir page 19*)

Un style est un ensemble de déclarations de style.

Chaque déclaration de style est la donnée d'une propriété et sa valeur : par exemple, “*taille du texte*” et “*12 points*” formant le couple propriété/valeur d'une telle déclaration.

B. Application directe d'un style à un élément :

Pour appliquer un style directement à un élément HTML, sélectionnez celui-ci ou placez votre curseur dans le contenu de cet élément, puis repérez cet élément dans la barre d'état et faites apparaître le menu contextuel avec le clic droit ; le menu (*Figure 14*) apparaît :

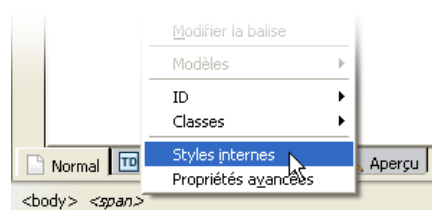


Figure 14 : “Menu des styles internes”

La commande **Styles internes** fait apparaître la boîte à dialogue gérant les feuilles de styles. L'examen des onglets de cette boîte permet de faire le tour des options de formatage qu'offrent les feuilles de styles.

1. Général :

La fenêtre (Figure 15) est l'interface de gestion des styles CSS internes à un élément HTML ; mais comme on le verra plus loin, elle gère également les feuilles de styles CSS :

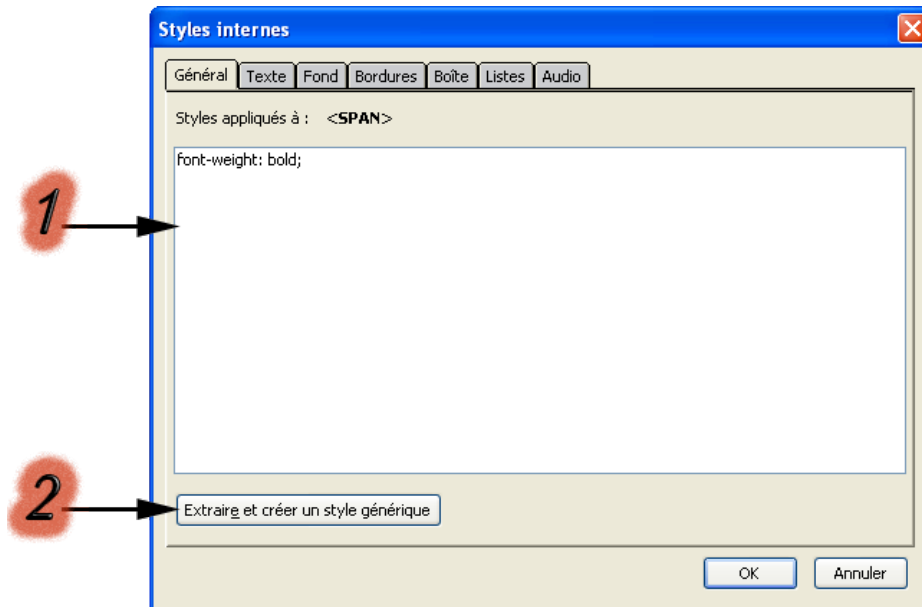


Figure 15 : “Fenêtre de gestion des styles internes”

Cet onglet présente en **1.** les déclarations de style affectés à l'élément courant dans le langage CSS : le code représentant votre style.

Le bouton **2.** permet de récupérer le style affecté à l'élément et d'en créer une règle pour être exporter dans une feuille internet ou externe (*voir les règles de styles - page 35*)

2. Texte :

Cet onglet (Figure 16) présente les propriétés courantes de formatage du texte :

Seul deux précisions sont à apporter :

1. Dans le choix de la police, la valeur “*Non défini*” laisse le navigateur utilisée sa police par défaut.

KompoZer propose des valeurs prédéfinies de police que tout ordinateur doit habituellement posséder ; un choix comporte plusieurs polices séparées par des virgules : la première rencontrée sur l'ordinateur du client est utilisée.

On peut personnaliser son choix également. Attention d'utiliser des polices suffisamment diffusées.

2. La propriété “*casse du texte*” permet de transformer tous les caractères de l'élément courant en majuscules, en minuscules, de les laisser inchanger ou de changer que la première lettre en majuscule.

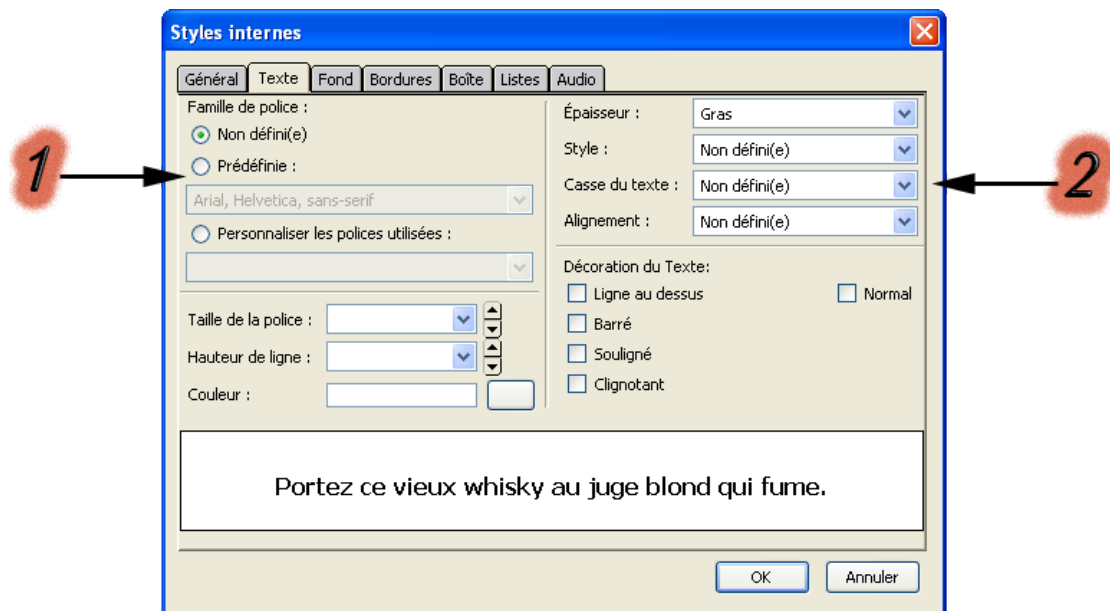


Figure 16 : “L’onglet Texte”

3. Fond :

Cet onglet (Figure 17) permet d’insérer une couleur ou une image de fond à l’élément courant qui peut être le corps du document `body` ou tout autre élément HTML.

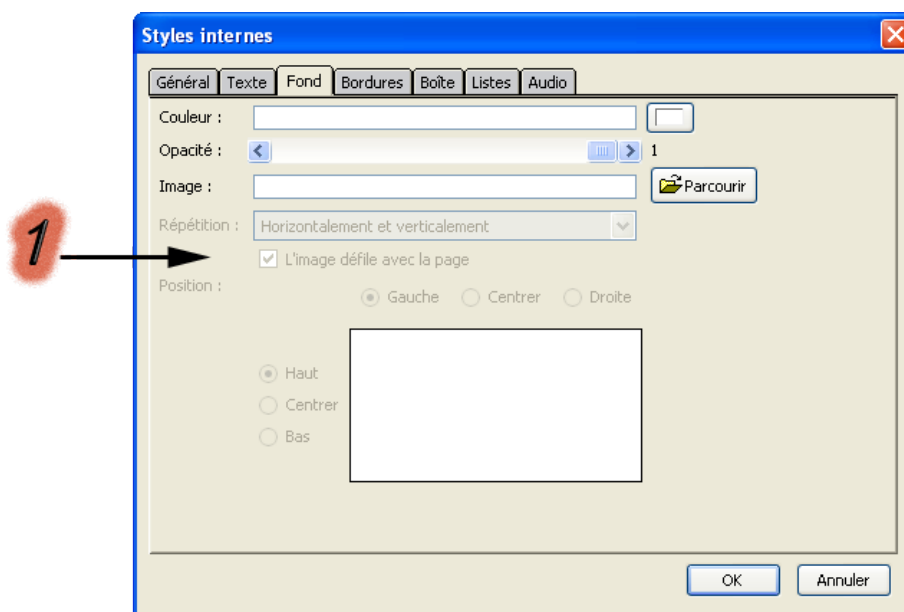


Figure 17 : “L’onglet Fond”

Voici deux commentaires quant à l’insertion des images :

- Lorsque la page ne peut pas s’afficher entière dans le navigateur, celui-ci affiche les barres de défilement : l’option 1. permet soit de laisser l’image fixe pendant que le client défile dans la page, soit de faire défiler l’image en même temps que la page.
- L’image de fond peut être insérée une seule fois dans l’élément ou être répétée plusieurs fois permettant le remplissage complet de l’élément (voir l’option “répétition”).

4. Bordures :

Cet onglet (Figure 18) permet de faire apparaître une bordure à l’élément courant.

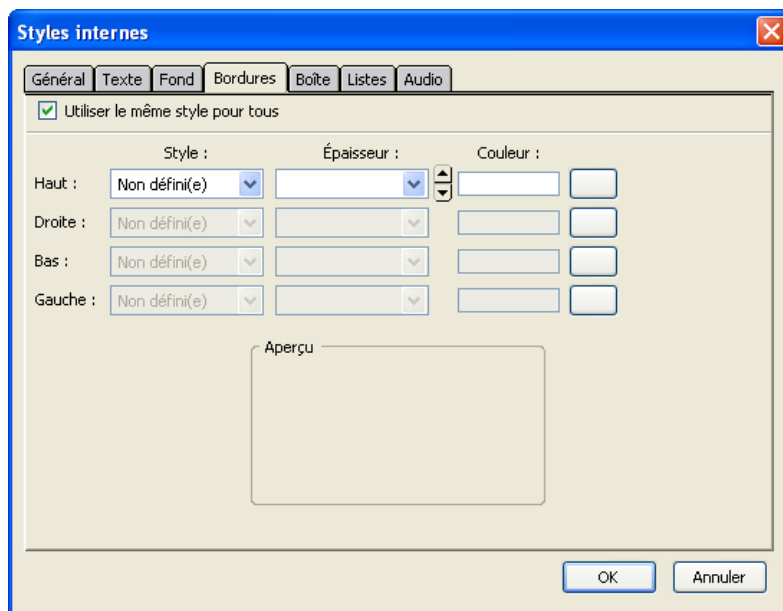


Figure 18 : “L’onglet Bordure”

En décochant l’option “*Utiliser le même style pour tous*”, vous pouvez choisir des bordures différentes pour les quatres côtés de l’élément.

⚠ Si vous trouvez la bordure trop proche du contenu de l’élément, vous devez rajouter de l’espace intérieur à l’élément (*padding en anglais*) à votre élément : voir dans l’onglet “*Boîtes*”.

5. Boîte :

Voici l’onglet (Figure 19) le plus important, le plus utile et le plus compliqué des feuilles de styles. Il permet de gérer :

- le positionnement des éléments dans la page
- la taille des éléments
- les marges entourant l’élément...

Voici la description des différents champs de contrôle de cette boîte à dialogue :

1. Flottante : cette option permet, utilisée par exemple avec une image, de placer l’élément sur le bord droit ou gauche de la page ; le texte se placera d’un côté ou de l’autre.
2. Position : seul deux options nous intéressent ici
 - statique : l’élément est placé dans le flux du texte.
 - absolue : l’élément est sortie du flux normal et peut être placé à un endroit précis de la page (*voir positionnement absolue - page 26*). Cette valeur est équivalent à l’utilisation du bouton se trouvant en haut à droite de toutes pages de KompoZer :



3. Index-z : deux éléments en positionnement absolue peuvent se superposer. Cette propriété prend pour valeur un nombre entier. Le nombre ayant la valeur de “*Z-index*” est placé au

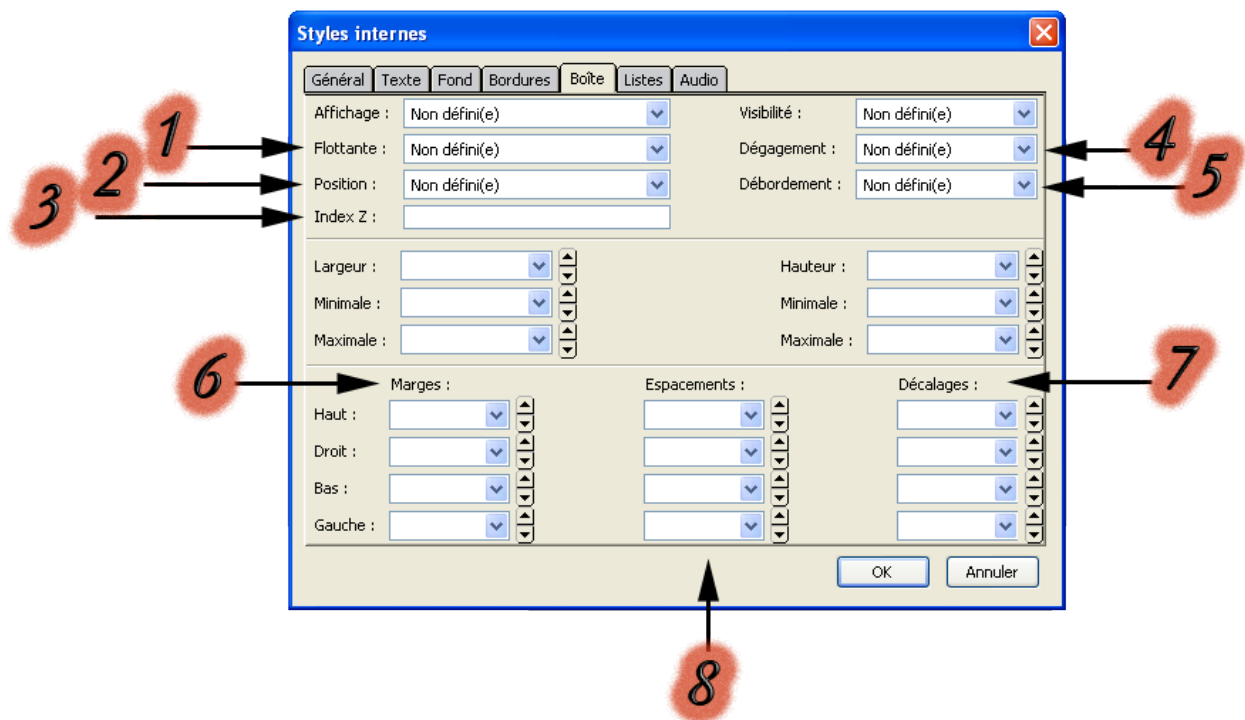


Figure 19 : “L’onglet Boîte”

premier plan (*voir page 26*).

4. Dégagement : cette option s’utilise conjointement à la propriété “*Flottante*”. Lorsque le texte s’écrit sur le côté de l’image et que le navigateur rencontre un élément ayant cette propriété (*généralement l’élément `br`*) alors le flux du texte se poursuit sous l’élément.
5. débordement : lorsque l’élément courant a une dimension imposée et que son contenu est plus grand, le navigateur peut, avec cette propriété, réagir de différentes manières :
 - visible : le contenu débordant de l’élément est quand même affiché.
 - masquer : le contenu débordant est caché.
 - défiler : l’élément se verra doter de barres de défilement.

Un élément est constitué d’une bordure (*même si l’épaisseur de celle-ci est nulle*) et d’un contenu :

- La marge 6. est l’espace séparant la bordure de l’extérieur : on parle de marge extérieure.
- L’espacement 8. est l’espace séparant la bordure de son contenu : on parle de marge interne.
- Lorsque l’élément est en positionnement absolu, son emplacement est déterminé à l’aide des options “*Décalages*” 7. .

⚠ Pour bien comprendre la différence entre “*marge*” et “*espacement*”, faites apparaître une bordure à un élément de la page et modifier ces deux valeurs.

6. Listes :

Le langage CSS permet d'ajouter de nouvelle puce et aussi de gérer la position de la puce ; l'onglet (Figure 21) permet de gérer l'apparence des listes.

7. Audio :

Pour les personnes mal-voyantes, le texte inscrit dans un élément peut être lû par le navigateur. A ma connaissance, seul Opera prend en compte des fonctionnalités pour l'instant. L'onglet (Figure 20) permet de gérer ces propriétés.

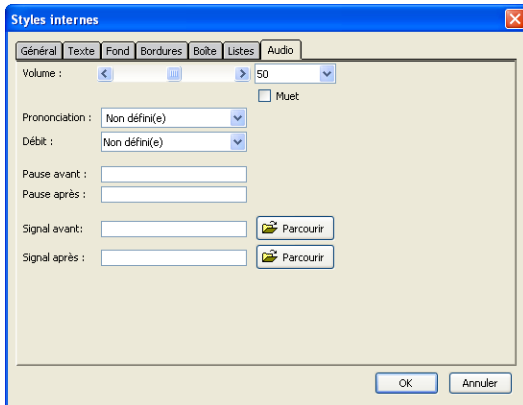


Figure 20 : "L'onglet Audio"

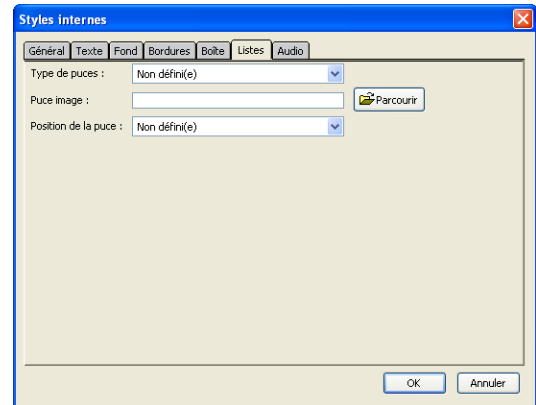


Figure 21 : "L'onglet Listes"

C. Les règles de style :

Dans les paragraphes précédent, nous avons vu comment affecter des déclarations de styles à un élément. Ces déclarations sont des styles internes inscrites directement dans la balise d'ouverture de l'élément (voir page 19).

On reconnaîtra de tels éléments en observant la représentation dans la barre d'état de leur balise d'ouverture en italique :

`<body>` `<div>` (comme ici l'élément `div`)

Il est également possible de placer ces définitions extérieurement à l'élément concerné. On les place dans l'entête du document ou dans une feuille de styles externe. Voici quelques avantages de cette technique :

- Plusieurs éléments HTML d'une page recevront facilement le même formatage (*titre, chapitre, commentaire...*) : facilitant ainsi la lecture d'une page.
- En inscrivant ces définitions dans une feuille externe, elles peuvent être utilisées conjointement par plusieurs pages : permettant l'homogénéisation d'un site.

On parle alors de "règles de style" qui s'appliqueront à des éléments de la page en fonction de leurs attributs d'identification.

Pour créer de telles règles de styles, facilement avec KompoZer, on part d'un style interne d'un élément et on l'exporte dans une nouvelle règle. Pour cela, on utilise le bouton :

"Extraire et créer un style générique" de l'onglet "Général".

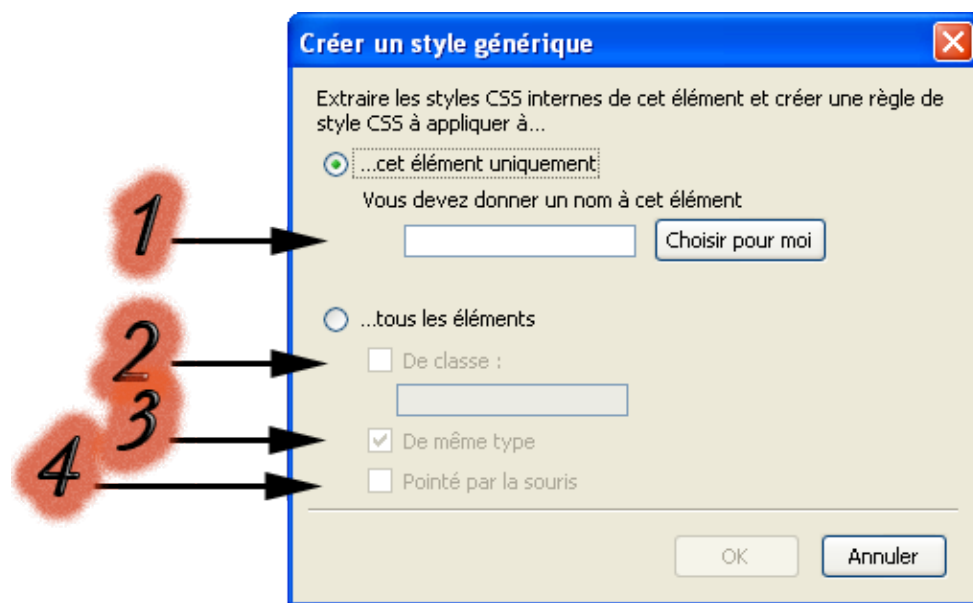


Figure 22 : “Création d’un style générique”

La fenêtre (Figure 22) apparaît.

Ces déclarations sont alors transformées en une règle. Une **règle** est la donnée d’un sélecteur et d’un ensemble de déclaration de style. Le sélecteur est le moyen de désigner les éléments qui récupéreront ces déclarations de styles pour leur affichage.

On s’intéressera ici à trois sélecteurs différents :

- Le sélecteur de classe **2.** :
Il permet d’affecter le style à tous les éléments qui auront pour attribut *class* la même valeur le nom du sélecteur de la règle.
- Le sélecteur d’identifiant **1.** :
Il permet d’affecter les déclaration de style qu’à un seul élément de la page.
- Le sélecteur du type d’élément **3.** :
Il permet de modifier les déclarations de style de tous les éléments de la page du même type que l’élément courant.

Les deux premiers sont les plus utilisés dans une page Web ; choisissez vos noms afin qu’ils soient suffisamment explicites sur les éléments qu’ils doivent pointés (*cela vous permettra une maintenance plus aisée de votre site*)

D. Affecter des règles aux éléments :

Après avoir extrait des styles et avoir créé des règles ayant un sélecteur de classe ou d’identifiant, voici comment lier une règle de style avec les éléments de la page en fonction de la nature de la règle

- Une règle de style appliquée par un identifiant :
Un seul élément de la page recevra les déclarations de style de cette règle. Après avoir

sélectionné cet élément, cliquez sur la balise lui correspondant pour actionner la commande :

menu contextuel \rightsquigarrow **ID** \rightsquigarrow **nom de l'identifiant**

Un élément affecté d'un identifiant unique verra sa barre d'état affichée sous la forme suivante :

```
<body> <div id="maPhoto">
```

- Une règle de style appliquée à une classe d'éléments :

Il faudra reproduire cette séquence sur tous les éléments de la classe. En sélectionnant chacun des éléments, actionnez la commande suivante à partir de la barre d'état :

menu contextuel \rightsquigarrow **Classes** \rightsquigarrow **nom de la classe**

Un élément affecté d'une appartenance à une classe verra sa barre d'état sous la forme suivante :

```
<body> <div class="chapitre">
```

E. Gestion des feuilles de styles :

1. Introduction :

Les règles sont stockés dans les feuilles de styles ; celles-ci peuvent être internes ou externes :

- Les “*feuilles internes*” sont stockés dans l'en-tête de la page Web ; aucun fichier supplémentaire n'est créé et ses règles ne peuvent pas être partagées avec les autres pages Web du site (*utilisation donc assez limitée*).
- Les “*feuilles externes*” sont des fichiers externes (*généralement avec l'extension css*) à la page utilisée pour stocker les règles de style ; l'utilité de telles feuilles et de pouvoir partager les définitions des règles entre plusieurs pages et ainsi d'homogénéiser la présentation d'un site.

La commande suivante permet d'ouvrir la boîte à dialogue (*Figure 23*) :

Outils \rightsquigarrow **Editeur CSS**

Ici, on observe que la feuille de style interne comprend une règle se nommant **adresse** et qu'une feuille externe de nom “*mesStyles.css*” possède quatre règles.

2. Les différentes règles :

Les règles contenues, dans chaque feuilles de style, peuvent être modifiées directement à l'aide des onglets se trouvant dans la partie droite de la boîte à dialogue.

En rentrant un peu dans la syntaxe des feuilles de styles CSS, les noms de règles peuvent être précédés d'un caractère. Voici les possibilités rencontrées avec KompoZer et leur explication :

- un point : la règle est liée à tous les éléments ayant pour attribut **class** le nom de la règle.

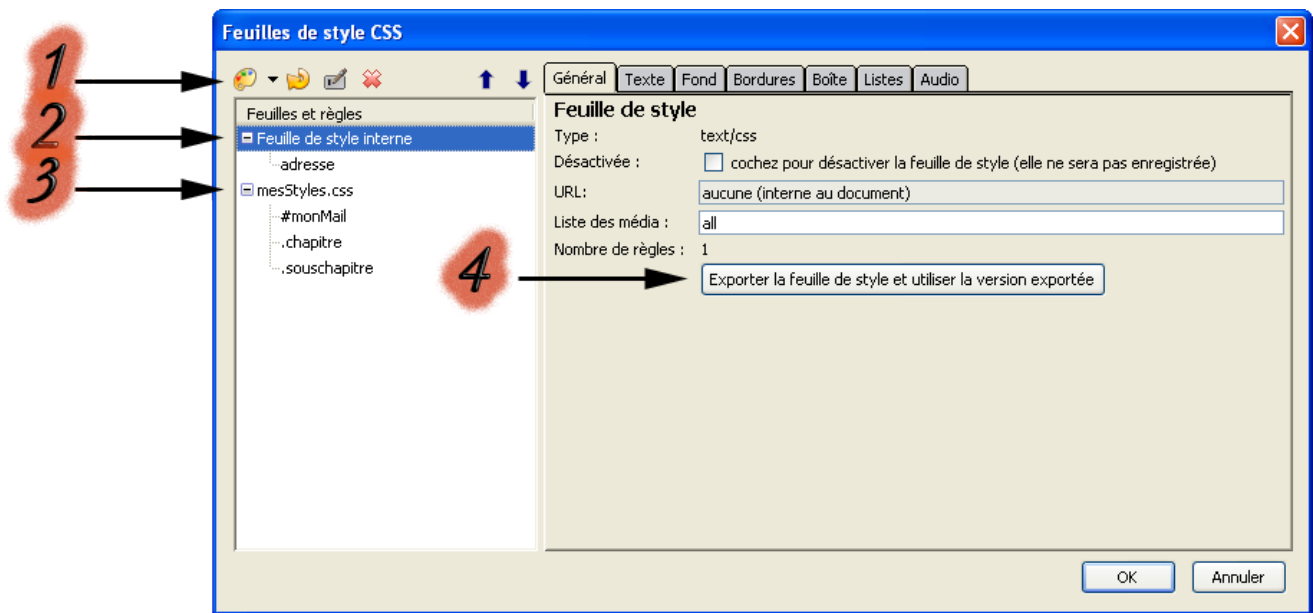


Figure 23 : “Gestion des feuilles de styles”

- un dièse : la règle est affectée à l’unique élément de la page (*s’il existe*) ayant la valeur de l’attribut *id* le nom de cette règle.
- si aucun caractère est en préfixe du nom de la règle, celle-ci s’appelle à tous les éléments ayant pour nom de balise, le nom de cette règle.

3. Créer une feuille de style externe :

Avec KompoZer, on crée des déclarations de style sur des éléments, puis on crée des règles qui sont placés, automatiquement, dans la feuille de style interne de la page courante.

Pour exporter ces règles dans un fichier externe et ainsi les rendre utilisables par d’autres pages, il suffit de sélectionner la feuille de style interne désirée et de cliquer sur le bouton “*Exporter la feuille de style et utiliser la version exportée*” dans le panneau de droite.

Pour cela, effectuer les opérations suivantes :

- Cliquez sur le bouton ;
- Dans la boîte à dialogue, dirigez-vous vers l’endroit où sera enregistré ce fichier ;
- Rentrez-le nom choisi pour ce fichier (*préférée l’extension “.html”*) et enregistrez.

Remarquez alors qu’un nouveau fichier vient d’être créé.

4. Gestion des feuilles de styles et des règles d’une page :

Dans ce paragraphe, nous allons voir comment créer directement des feuilles de styles vierges et comment créer directement ses propres règles sans repasser par les styles internes d’un élément ; Voici le bouton qu’on utilise (*Figure 24*) :

et ces différentes commandes proposées :

- “@import” : elle permet d’inclure l’importation d’une feuille de style externe ; en pratique

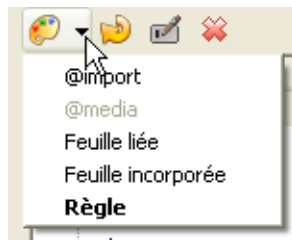


Figure 24 : “Bouton de création de feuilles et règles”

cette option n’est pas utilisée.

- “feuille liée” : cette commande lie la page courante à une feuille de style externe en indiquant son URL (*pensez aux URL relatives*) ; si celle-ci n’existe pas, elle sera créée.
- “feuille incorporée” : cette commande crée une nouvelle feuille de styles interne ; généralement une seule par document est suffisante.
- “règle” : cette commande est l’action par défaut de ce bouton ; elle permet de créer une nouvelle règle dans la feuille de styles courantes. En cliquant dessus apparaît dans le panneau de gauche l’intitulé “Nouvelle règle de style” et il est proposé quatre modes pour la création d’une nouvelle règle (*dans la feuille sélectionnée*)

⇒ une règle s’appliquant à tous les éléments d’un même type.

⇒ une règle s’appliquant à tous les éléments ayant le même nom de classe (*remarquer le point précédant le nom de la classe*).

⇒ une règle s’appliquant à l’élément portant le même ID (*remarquer le dièse séparant le nom de l’ID*).

⇒ ou manuellement pour l’utilisateur expérimenté.

⚠ N’oubliez pas de valider votre choix en appuyant sur le bouton “Créer la règle de style”.

Cette boîte de dialogue permet également de modifier toute règle liée par une feuille à cette page.

Le bouton de droite en forme de croix permet d’effacer une règle ou une feuille de style.

XI. Javascript :

A. Introduction :

JavaScript est un langage de programmation utilisé dans les pages Web. Il permet de créer des pages dynamiques : des pages qui peuvent réagir aux actions du client. La page peut :

- détecter le déplacement de la souris du client au dessus d’un élément.

- Repérer l'enfoncement d'une touche.
- Effectuer une tâche à intervalle régulier (*changement de la couleur du fond de la page*)

Le langage **JavaScript** n'est pas développé dans cette partie de la formation. Son apprentissage permet une bonne introduction aux langages informatiques par sa simplicité et son utilité dans les pages Web.

KompoZer permet modestement de placer des bouts de code à quelques endroits dans le code HTML mais ceci ne satisfera nullement un programmeur en **JavaScript** qui préférera toujours saisir son programme directement dans le code HTML de la page.

B. Utilisation avec KompoZer :

Voici un exemple d'utilisation de Javascript au travers de KompoZer :

1. Créez un conteneur **div** et saisissez un peu de contenu.
2. Dans la barre d'état, cliquez avec le bouton droit sur l'élément `<div>`. Dans le menu contextuel, choisissez la commande **Propriétés avancées**
3. Dans la boîte à dialogue sélectionner l'onglet "Événements Javascript".
4. Sélectionnez l'attribut "onmouseover" avec la valeur :
`this.style.backgroundColor="#EEEEEE"`
5. Sélectionnez l'attribut "onmouseout" avec la valeur :
`this.style.backgroundColor="#EEFFAA"`
6. Validez vos changements.

Ainsi, dès le passage de la souris au dessus de votre élément, l'élément **div** verra sa couleur de fond se modifier.

Voici le résumé de vos changements dans la fenêtre ci-dessous :

