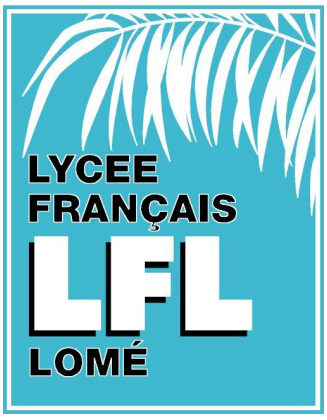
HTML

&

CSS

Travaux dirigés



Sommaire :

I. Introduction :

KompoZer se charge pour nous de remplacer nos actions sur les boutons ou avec la souris en langage HTML et intègre directement les styles CSS dans les éléments dans notre page ; danc cette partie de la formation, nous allons apprendre à itomoser directement le langage HTML et le langage CSS.

Le langage HTML, ainsi que le langage CSS, ne sont pas des langages de programmation mais plutôt de description. De plus, ils ne permettent de produire que de simples pages HTML et les fonctionnalités les plus intéressantes des pages Webs telles que viennent de l'utilisation d'autres langages de programmation :

- Le langage **JavaScript** permet de relier l'action du client *(pression d'une touche ou d'un bouton de la souris, déplacement de la souris)* avec les éléments HTML en les cachant ou les affichant, en modifiant leurs couleurs...
- Le langage PHP permet, sur le serveur, de contrôler les informations envoyées par un client *(par exemple par un formulaire)* et de lui renvoyer l'information adéquate. Il est possible de demander au PHP de créer dynamiquement la page qu'il renvoye au client.
- Une base de donnée MySQL permet de stocker les informations reçues par un formulaire ; si votre site contient des milliers de photos, il est plus facile de les classer et de les ordonner dans une base de donnée.

L'apprentissage, bien que très simple, du langage HTML nous permmet de rentrer dans le coeur de la programmation informatique. Nous allons comprendre comment l'ordinateur fonctionne, comment un navigagteur effectue l'affichage d'une page Web. De plus, sa connaissance est indispensable dans la programmation en **JavaScript** et en PHP.

Ainsi, nous devons considérer le langage HTML comme une étape vers d'autres langage ; comme un premier apprentissage dans le monde de la programmation et vers une compréhension plus grande de l'informatique : savoir ce qui se passe derrière le simple fait de cliquer sur des boutons ou déplacer des fenêtres.

Le langage HTML n'est pas un langage de programmation mais plutôt un langage de descriptions : on indique au navigateur comment afficher des éléments.

Pour connaître le code source d'une page Web, il suffit d'ouvrir le fichier "*.html*" avec un simple éditeur de texte :

menu contextuel ⇝ Ouvrir avec ... ⇝ notepad, wordpad ou emacs

Attention, Word est un traitement de texte *(pas un simple éditeur de texte)*. Ainsi, si vous ouvrez un fichier "*.html*" dans Word, celui-ci tentera de l'afficher à la manière d'un navigateur : un traitement est effectué avant l'affichage.

Le code d'une page Web est constituée :

- du texte contenu dans la page ;
- d'élément HTML qui indiqueront au navigateur le formatage qu'il doit apporter au contenu de la page.

Un élément HTML est habituellement constitué de deux balises ; ces deux balises permettent de délimiter le champ d'action de l'élément HTML. Certains éléments HTML ne possédant pas de contenu sont définis seulement par une balise ouvrante. Chaque élément de type `p` est constitué :

- d'une balise la balise ouvrante `<p>` ;
- d'une balise fermante `</p>`.

Le contenu d'un élément de type `p` est affichée par les navigateurs comme un paragraphe : un espace au dessus et en dessous sont insérés. Nous allons parcourir dans cette formation les éléments HTML les plus couramment utilisés.

Exercice 1

Dans la suite de la formation, nous créons entièrement nos pages Webs ; il est alors plus facile de partir d'une page Web. C'est pourquoi nous partons toujours d'un fichier texte vide qu'on transforme en une page Web :

1. Créez sur le bureau un fichier texte :

   menu contextuel ⇝ Nouveau ... ⇝ Document texte

2. Editez ce fichier à l'aide de NotePad *(double-cliquez sur le fichier)* et recopier le texte ci-dessous :

```
1 Bonjour <div>monsieur</div>,
2
3 Quel est votre <b>nom</b>? Votre <i>pr\'enom</i>
```

Vous devez vous assurer que Windows affiche les extensions des fichiers ; si ce n'est pas le cas, vous pouvez vous reporter à l'annexe du TD de KompoZer qui indique la démarche à suivre.

3. Renommez ce fichier avec l'extension "*.html*", puis affichez-le dans un navigateur.

4. Observez l'affichage de votre fichier afin de répondre aux questions suivantes :

   a. Quel effet a l'élément `div` défini par les deux balises `<div>` et `</div>` sur son contenu?

.....

.....

b. Comment est formaté le contenu de l'élément `b` ?

.....

c. Quel formatage apporte l'élément `i` a son contenu?

.....

.....

d. Quelles autres remarques pouvez-vous apporter sur la différence entre le code et la représentation de celui-ci dans une page Web?

.....

.....

.....

II. Installation des logiciels requis :

Dans cette partie, nous allons installer les différents logiciels nécessaire à la suite de la formation :

- FireFox : ce navigateur possède de nombreux pluggins facilitant le diagnostique d'erreurs de syntaxe.
  Il est également, avec Opéra, l'un des navigateurs respectant au mieux les standards HTML, CSS, **JavaScript**préconisé par le consortium W3C.

- Le module "*HTML Validator*" permet d'inspecter le code source d'une page et reporte les erreurs de codage de la page.

- L'éditeur de texte Emacs est utilisé pour saisir le code source de nos pages Web ; en coloriant automatiquement les mots comportant une signification pour le langage HTML, il facilite la saisie du code HTML.

D'autres éditeurs de texte sont également bien adaptés à l'utilisation mais Emacs est lui-même entièrement programmable à l'aide du langage LISP ; il a été configuré pour cette formation pour enregistrer et afficher automatiquement votre source dans un navigateur en lorsque vous actionnez la touche "*F1*" ; cela nous permettra de vérifier plus fréquemment l'effet d'un changement sur notre code.

Exercice 2

Installons l'éditeur de texte Emacs sur l'ordinateur. Plutôt que de télécharger l'ensemble des applications et modules le constituant, nous allons utiliser une version "*pré-installée*" présente dans le CD d'installation.

1. Copiez le dossier "*b-emacs*" présent dans le CD de la formation sur votre bureau.

a. Décompressez le fichier "*emacs.rar*" à l'aide de la commande *(Winrar doit être installé)* :

menu contextuel ⇝ Extraire vers emacs/

Déplacer le dossier "*emacs*" à l'emplacement suivant :

`C:\Program Files\`

b. Déplacer le fichier "*.emacs*" à la racine de votre disque dur `C:`.

c. Actionnez le fichier "*ouvrirAvecEmacs.reg*" ; modifiant la base de registre, ce fichier permet d'ouvrir plus facilement les fichiers "*.html*" avec Emacs ; un clic droit sur un fichier "*.html*" permet d'accéder à la commande :

menu contextuel ⇝ Ouvrir avec... ⇝ GNU Emacs

Exercice 3

Nous allons créer une page Web toute simple et l'éditer avec Emacs pour commencer à se familiariser avec lui :

1. Sur le bureau, on part d'un fichier vide :

   menu contextuel ⇝ Nouveau... ⇝ Document texte

   Puis, renommez-le avec l'extension "*.html*" ; on obtient une page Web vide.

2. A l'aide du clic-droit, ouvrez le fichier :

   menu contextuel ⇝ Ouvrir avec... ⇝ GNU Emacs

3. Saisissez le texte suivant :

```
<div align="center">Bonjour</div>
<b>Vous allez bien?</b> Je l'<i>espère</i>.
```

4. Au fur et à mesure que vous tapiez votre texte, Emacs a réagi au texte saisi en mettant des parties du texte saisi en évidence ; précisez :

.....

.....

.....

.....

.....

5. Actionnez maintenant la touche *F1* pour que Emacs affiche la page Web dans le navigateur FireFox.

Emacs rajoute à l'affichage des couleurs et modifie la fonte de certaines parties du code source en fonction des mots clefs utilisés :

- Les chaînes de caractères, entourées de guillemets, sont affichées en gris ;
- Le type des balises est affiché en bleu ;
- Le contenu de l'élément `b` est affiché en gras ;
- Le contenu de l'élément `i` est affiché en italique.

Attention, le fichier que vous enregistrez ne contiend que ce que vous saisissez ; les changements de couleurs ou de fontes sont effectués uniquement par Emacs ; aucune trace de ce formatage n'est conservé dans le fichier final.

Exercice 4

Firefox intègre une boîte à dialogue présentant les erreurs CSS ou **JavaScript** présentent dans une page Web ; pour obtenir des informations sur l'intégrité de notre code, nous allons utilisé le modue "*tidy_firefox*" :

1. Dans FireFox, ouvrez la fenêtre des modules :
   Outils ⇝ Modules complémentaires
2. Récupérer le fichier "*tidy_firefox_win_0853.xpi*" se trouvant dans le dossier de la formation dans le dossier `c-firefox`, puis faîtes le glisser dans la fenêtre des modules de FireFox.
3. Après avoir accepté l'installation, vous devez redémarrer Firefox pour activer le module.
4. Au redémarrage de FireFox, le module "*HTML Validator*" vous proposera de choisir le type de vérification. Pour l'apprentissage, choisissez "*HTML tidy*" ; ce mode nous apporte une aide plus confortable que l'autre algorithme.

Le module est maintenant installé.

Exercice 5

Nous allons voir ce que nous apporte ce nouveau module :

1. Ouvrez le fichier "*.html*" précédemment créé avec FireFox.
2. Vous devez voir l'icône suivant en bas à droite de votre page 
3. En cliquant deux fois dessus, la fenêtre de "*HTML Validator*" ouvre une boîte à dialogue *(Figure 1)* :

Voici une présentation de la fenêtre de "*HTML Validator*" :
- La partie supérieure présente le code HTML de la page.

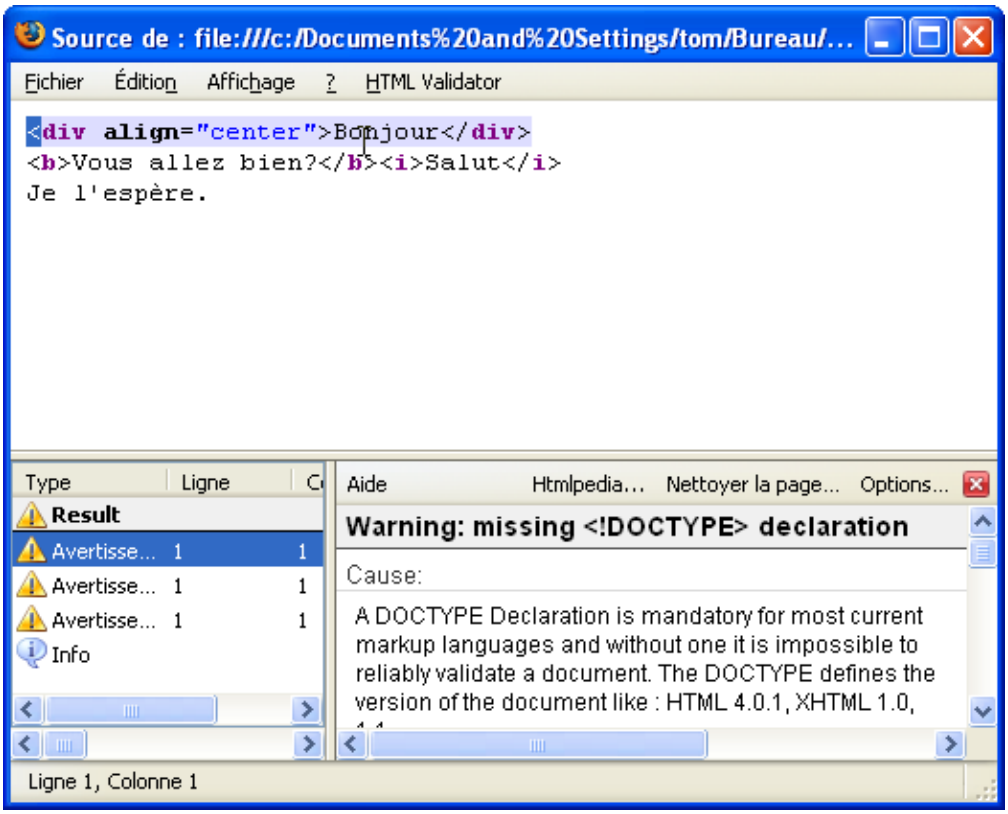


Figure 1: "*Fenêtre de HTML Validator*"

- La partie inférieure gauche présente les différentes erreurs relevées par "*HTML Validator*". Un avertissement est une erreur que le module a corrigé. Ce module présente comme erreur tout ce qu'il n'a pû corrigé automatiquement.
- Pour chaque avertissement ou erreur, "*HTML validator*" propose son interprétation dans la partie inférieure droite.

4. "*HTML Validator*" vous propose de corriger les erreurs présentes dans votre page. Pour cela, cliquez sur le bouton :

   Nettoyer la page...

Voici le code proposée par "*HTML Validator*" :

```
1  <!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01//EN">
2  <html>
3   <head>
4    <title></title>
5   </head>
6   <body>
7    <div align="center">Bonjour</div>
8    <b>Vous allez bien?</b><i>Salut</i>Je l'esp&egrave;re.
9   </body>
10 </html>
```

Voici une première explication du code proposé :

- La structure minimale d'une page *(les éléments* `html` , `body` , `head` *)* a été complétée par "*HTML Validator*"
- L'élément `doctype` indique la version du HTML utilisée *(voir page* **??** *sur le tutorial HTML).*
- L'élément `title` doit être présent dans l'entête du document *(l'élément* `head` *)*. Il indique le titre de la page.

III. Les attributs :

Voici quelques exemples de fonctions qu'un élément HTML apporte à une page Web :

- Il apporte un formatage particulier à son contenu.
- Il permet d'insérer une image ;
- Il indique la présence d'un lien hypertexte...

Chaque élément HTML possède un certain nombre d'attribut que le programmeur peut utiliser pour modifier le comportement de l'élément. Un attribut s'inscrit dans la balise ouvrante de l'élément. Voici un exemple générique où l'élément de type `elt` s'est vu rajouter deux attributs :

```
<elt attr1="valeur" attr2="valeur">.........</elt>
```

Remarquez la structure *attribut*=*valeur* d'écriture des attributs en HTML.

En fin du manuel sur le HTML, vous trouverez :

- La liste des éléments HTML avec les attributs acceptés par chacun d'eux.
- La liste des attributs avec pour chaque attribut les éléments HTML acceptant l'attribut.

Exercice 6

Nous allons voir l'utilisation de quelques atributs :

1. Commencez par copier, à partir du CD de la formation, le dossier "*d-exerciceHtmlCss*" sur votre bureau.
2. Editez le fichier "*_exerciceA.html*" à l'aide de Emacs.
3. Complétez le code de votre fichier afin qu'il soit identique au code ci-dessous ; pensez à

chaque changement à appuyer sur la touche “*F1*” afin d'observer les modificiations apportées lors de l'affichage de la page :

```
<div align="center" title="G. Brassens">
La prière</div>

Par le petit garçon qui
<font size=5>meurt près de sa mère</font><br>
Tandis que
<font color="red">des enfants</font>
 s'amusent au parterre<br>
Et par l'<span style="background-color:yellow">
oiseau blessé</span> qui ne sait pas comment
Son aile tout à coup <ul style="margin:0px">
<li type="circle">s'ensanglante
<li type="square">et descend
</ul>
Par <a href="http://www.google.fr">
la soif et la faim</a> et le délire ardent
 Je vous salue, Marie.
```

4. A chaque changement, utilisez la touche “*F1*” pour observer le changement apporté à l'élément. Complétez alors les lignes vides ci-dessous en marquant vos remarques :

- `<div align="center" title="G. Brassens">`

. . . . . .

. . . . . .

- `<font size="5">`

. . . . . .

- `<font color="red">`

. . . . . .

- `<span style="background-color:yellow">`

. . . . . .

- `<ul style="margin:0px">`

. . . . . .

. . . . . .

- `<li type="circle">`

. . . . . .

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

. . . . . .

Exercice 7 *(Mise en évidence des éléments et de leurs attributs)*

On considère le code ci-dessous :

```
1 <p id="titre" style="font-size:15pt;color:red">
2 Mon titre</p>
3
4 <div id="chap1">
5 <span class="chap">chapitre 1</span>
6 Texte du chapitre 1</div>
7
8 <div><span class="chap">chapitre 2</span>
9    Texte du chapitre 2</div>
```

1. Combien d'élément HTML comptez-vous dans cette page?

2. L'élément `p` possède combien d'attributs? Donnez la valeur de chacun de ses attributs

3. Ecrire le contenu de l'élément possédant l'attribut `id` avec la valeur *"chap1"*?

4. Combien d'éléments ont la valeur *chap* pour l'attribut `class`?

IV. Un peu de syntaxe :

La syntaxe d'une langue est l'ensemble des règles qui régissent l'arrangement des mots, la construction des phrases et les rapports entre elles.

En général, dans un langage informatique, la syntaxe est stricte :

- un espace de trop peu provoquer une erreur,
- une minuscule remplacée par une majuscule peut empêcher le langage de reconnaître un mot.

Pour le langage HTML, la syntaxe est assez simple et assez lâche. Mais quelques règles sont encore à respecter. Les exercices ci-dessous sont conçus pour faire le tour de quelques unes de ces règles.

Exercice 8 *(Les espaces et les sauts de lignes)*

1. Créez un nouveau fichier "*.html*" vide et saisissez-y le code ci-dessous :

```
Bonjour , monsieur       comment
    allez -vous?

Bonne journée.
```

2. Afficher cette page dans votre navigateur, en actionnant la touche "*F1*". Quel différences remarquez-vous entre le code et son affichage dans un navigateur?

.....

.....

.....

Voici quelques règles syntaxiques du langage HTML :

- Plusieurs espaces saisis consécutivement ne sont interprêtés que comme un seul espace.
- Les sauts de lignes présents dans le code sont interprêtés à l'affichage comme étant des espaces.

Voici quelques astuces pour forcer un navigateur à afficher des espaces consécutifs ou des sauts de lignes :

- La séquence "` `" représente un espace insécable *(abbévation anglaise de Non Breaking Space)*. Plusieurs de ces espaces inscrits consécutivement seront affichés.
- L'élément br, défini par sa seule balise ouvrante, permet l'affichage d'un saut de ligne.
- L'élément p, défini par les deux balises et , permet de spécifier un paragraphe. Le navigateur insérerun espace avant et après cet élément pour renforcer l'apparence des paragraphes.

3. Utiliser les éléments HTML présentés dans l'encadré ci-dessous pour saisir un code dans votre fichier tel que l'affichage dans un navigateur fasse apparaître des espaces consécutifs, un saut de ligne et un saut de paragraphe *(à l'image du code présenté ci-dessus)*.

Exercice 9 *(L'écriture des balises et des attributs)*

- Un élément HTML est défini par une balise ouvrante et la plupart du temps par une balise fermante.

- La balise ouvrante est défini par les deux symboles < et >. Le nom de l'élément doit être directement précédé par le signe < ; un espace entre < et le nom de l'élément empêche le navigateur de reconnaître l'élément HTML. Le codage suivant est incorrect :

  `< b>Mon titre</b>`

- Il en va de même pour la balise fermante dont le nom de l'élément doit directement suivre </ et être suivi de >.

- Le nom des balises est insensible à la casse : le langage HTML ne fait pas de distinction entre les majuscules et les minuscules. Ainsi, les différentes balises ouvrantes suivantes définissent le même élément :

  `<table>` ; `<TABLE>` ; `<taBLE>`

- Normalement, la valeur d'un attributs est une chaîne de caractère, elle doit être entourée de guillemets, même si dans de nombreux cas ceux-ci sont facultatifs, dans certains cas ils sont obligatoires :

  `<font face=Comic Sans MS>` est incorrect

  et doit se coder ainsi, pour récupérer la police "*Comic Sans MS*" :

  `<font face="Comic Sans MS">`

1. Dans un fichier "*.html*" vide, saisissez le code ci-dessous, en respectant tout ce qui est donnée *(espace supplémentaire ou autre)* et tout ce qui n'est pas donné *(abscence d'espace ou de guillemet)*

```
1 < b > Bonjour <i> Charles< /i>.</ B>Ca va?
2 <div style=margin:10px 50px;color:red>Que veux-tu ↩
    manger?</div>
```

2. Affichez ce fichier dans un navigateur. Qu'observez-vous?

.....

.....

3. Modifiez le code afin que toutes les balises soient reconnues par le navigateur et que l'attribut `style` ait pour valeur *"margin:10px 50px;color:red"*. Puis affichez cette page dans un navigateur pour observer la différence.

Exercice 10 *(L'imbrication des éléments)*

Les éléments HTML peuvent s'imbriquer les uns dans les autres mais ne peuvent pas se chevaucher : un élément peut être contenu dans un autre élément mais alors il doit y être entièrement contenu.

1. Saisissez le code suivant dans un fichier "*.html*" vide.

```
1 <b>Bonjour <i>Jean-Paul</b>. Comment allez-vous?</i>
```

2. Affichez la page dans un navigateur et indiquez les caractéristiques de la fonte utilisée pour chacune des parties suivantes :
   - "*Bonjour*" :

.....

   - "*Jean-Paul.*" :

.....

   - "*Comment allez-vous?*" :

.....

3. En ouvrant "*HTML Validator*" en cliquant sur l'icône  en bas à droite de votre navigateur, vous verrez les erreurs HTML signalées dans ce module.

   "*HTML Validator*" vous proposera un code correct en cliquant sur le bouton "*Nettoyer la page...*", notez ce code et prêtez une attention particulière à l'imbrication des éléments :

.....

.....

.....

Tel qu'on a écrit le code :

- La balise ouvrante `<i>` est inclu dans l'élément b, ainsi la présence de la `</i>` est obligatoire avan la fermeture de l'élément b.
- La balise fermante `</i>` est présente en fin de ligne ; la séquence "*Comment allez-vous?*" doit être affichée en italique : elle doit être contenue i.

Ce genre d'erreur de syntaxe est bien gérée par les navigateurs, mais il est toujours préférable de coder dans les règles. Cela peut posséder de conséquence lorsqu'on utilise **JavaScript** :

```
1 <b>Bonjour <i id=ital>Jean-Paul</b>. Comment ↩
    allez-vous?</i>
```

Quel est alors l'élément ayant *ital* pour identifiant?

V. Quelques éléments HTML :

Dans ce paragraphe, nous passons en revue quelques une des balises les plus fréquemment rencontrées dans une page Web ainsi que leurs attributs les plus souvent utilisées.

De nombreuses balises supplémentaires existent ainsi que d'attributs pouvant s'y rattacher mais les feuilles de styles CSS ont réduit l'importance de ceux-ci ; reportez-vous à l'annexe du manuel sur le HTML pour connaître toutes les balises et tous les attributs ; reportez vous au site du W3C pour avoir leurs significations.

A. ... ; ... :

Exercice 11

1. Saisissez le code ci-dessous :

```
1 <div> Ce lundi</div> il faut que j'<span>aille</span> ↩
    à Yaoundé
```

2. Affichez cette page dans un navigateur. Quelles semblent être les fonctions des éléments div et span ?

.....

.....

Les éléments div et span n'ont pour seul fonction que de contenir d'autres éléments :

- div : affiche son contenu sur une ou plusieurs lignes ; on parle d'éléments en bloc.
- span : affiche son contenu sur la ligne courante ; on parle d'éléments en ligne.

Leurs intérêts résident dans le fait de contenir un ensemble d'éléments *(son contenu)* dans un seul élément. Ainsi, on peut facilement affecter un même style CSS à plusieurs éléments de la page ; on peut contrôler via **JavaScript** plusieurs éléments en une fois.

3. a. Dans le code, rajoutez les attributs suivants :
   - A l'élément div, rajoutez l'attribut ***align*** avec la valeur *center*.
   - A l'élément span, rajoutez l'attribut ***style*** avec la valeur *color:red*.

   b. Affichez votre page dans un navigateur pour observer les différences :

.....

.....

B.
 ; <p> :

Exercice 12

1. Dans un fichier "*.html*" vide, saisissez le code ci-dessous :

```
L'algèbre élémentaire<br>est
<p>la partie des mathématiques</p>
<p>qui étudie les grandeurs</p> en substituant
<br>des<br> lettres aux valeurs numétiques
```

2. En affichant cette page dans un navigateur, déterminez la fonction de chacun de ces deux éléments :
   - L'élément br

.....

   - L'élément p

.....

.....

3. Supprimez les balises fermantes dans votre source. Remarquez-vous des changements :

.....

L'élément br indique un saut de ligne.

L'élément p, par ses deux balises ouvrantes et fermantes, délimite un paragraphe ; le navigateur rajoute des espaces verticaux au dessus et en dessous de l'élément pour le mettre en évidence.

La balise fermante de l'élément p est optionnelle ; mais, attention, Internet Explorer et FireFox ne réagisse pas de la même manière à la rencontre d'un élément p dont la balise fermante n'a pas été explicitement saisie. Les dernières questions sont là pour le faire remarquer.

4. a. Compléter votre code par cette dernière ligne :

   Le mot Algèbre vient de l'arabe Al-Jabr

   Puis, observez l'affichage avec Internet Explorer "*F2*" et avec FireFox "*F1*".

   b. Quelle différence notez-vous?

   c. Fermez, dans votre code, explicitement l'élément p avant la balise ouvrante ; puis, affichez la nouvelle page dans les deux navigateurs pour observer s'il subsiste des différences d'affichage.

C. <a>... :

Les liens hyper-textes permettent aux clients d'un site de naviguer de pages en pages

; elles sont définies par l'élément a. Il est définit par les deux balises <a> et et le contenu de ses deux balises définit la zone cliquable représentant le lien.

L'attribut ***href*** prend pour valeur l'URL vers laquelle est redirigé le client lors de l'actionnement du lien.

Exercice 13

1. Dans le dossier "*d-exerciceHtmlCss*" qui doit se trouver sur le bureau, éditer le fichier "*_exerciceB.html*" avec Emacs.

2. a. En lisant attentivement l'encadré ci-dessus, utilisez l'élément a pour créer un lien redirigeant liant le mot "*Google*" avec l'adresse :

   `http://www.google.fr`

   b. Afficher la page dans votre navigateur et testez le lien.

   c. Dans le code source, ajoutez, à l'élément a, l'attribut ***target*** avec la valeur *"_blank"*.

   d. Dans votre navigateur, rechargez votre page avec la touche "*F5*" et cliquez plusieurs fois sur le lien. Que remarquez-vous?

.....

.....

3. Toujours dans le code de cette page, dans le second paragraphe, faîtes en sorte que le mot "*ici*" soit un lien redirigeant le client vers le fichier "*_exerciceA.html*" présent à la racine du dossier de ces exercices.
   Testez votre lien dans un navigateur.

4. a. Modifiez le code source de cette page pour que le mot "*Cameroun*" devienne un lien redirigeant le client vers le fichier "*cameroun.html*" présent dans le dossier "*ressource*".
   Testez votre lien.

   b. Ajoutez à ce lien l'attribut ***target*** avec la valeur *"nvlFenetre"*.

   c. Testez ce lien dans un navigateur, puis cliquez plusieurs fois sur ce lien que se passe-t-il?

.....

.....

L'élément a définit un lien dans une page Web et l'attribut ***href*** définit l'adresse de redirection.

L'attribut ***target*** permet de définir la fenêtre de navigation dans laquelle sera ouverte le lien.

- Pour ouvrir une nouvelle fenêtre à chaque action du lien, on utilise la valeur *target*

:

```
<a href="..." target="_blank">...</a>
```

- L'action par défaut des navigateurs est d'ouvrir le lien dans la fenêtre où celui-ci se trouve. On retrouve ce comportement avec la valeur *"_self"* de l'attribut **target**.
- Les valeurs *"_top"* et *"_parent"* étaient utilisées pour l'utilisation des cadres. Ceux-ci sont déconseillés à l'utilisation et de plus en plus abandonnés.
- Toutes autres valeurs ouvrent le lien dans une nouvelle fenêtre et celle-ci est identifiée par la valeur de **target** permettant l'ouverture d'autres liens dans cette nouvelle fenêtre.

D. <img> :

L'élément img permet de d'inclure des images dans la page Web. C'est à travers l'attribut **src** qu'on précise l'adresse absolue ou relative de l'image à insérer. Cet élément n'est défini que par sa balise ouvrante :

```
<img src="...">
```

Exercice 14

1. Créez une page Web vierge à la racine du dossier des exercices et éditez-le à l'aide d'Emacs.
2. a. Insérez un élément img affichant l'image "*drapeaucameroun.gif*" présent à la racine du dossier.

   b. Ajoutez à cet élément l'attribut **border** avec la valeur *"2"*. Que remarquez-vous?

. . . . . .

3. a. Insérez dans votre document l'image "*wouri.jpg*" présente dans le dossier "*image*".

   b. Dans la balise ouvrante, ajouter les attributs **width** et **height** ayant pour valeurs respectives *"200"* et *"45"*.

E. ... ; ... ; :

Le langage HTML propose des éléments pour créer facilement des listes. Voici les éléments HTML utilisés dans la construction de liste :

- ul définit une liste non-ordonnée : chaque élément de la liste est mis en évidence par une puce graphique le précédant.

- ol définit une liste ordonnée : le marqueur d'un élément est un entier décimal, un nombre romain ou un caractère s'incrémentant pour tout nouveau élément de la liste.
- Quelque soit le type de la liste, un élément d'une liste est défini par l'élément li.

1. Dans un fichier "*.html*" vide, saisissez le code suivant :

```
1 <ol>
2  <li> 1
3  <li> 2
4   <ul>
5    <li> 1
6    <li> 2
7   </ul>
8  <li> 3
9 </ol>
```

2. Dans un navigateur, affichez ce fichier. Observez les deux types de listes différents et leur emplacement dans le code.

3. Rajoutez à l'élément ol l'attribut ***start*** avec la valeur *"4"*. A près avoir affiché la page dans un navigateur, écrivez vos remarques :

.....

4. Dans la balise ouvrante des éléments li des lignes 2 et 6, inscrire l'attribut ***type*** avec respectivement les valeurs *"a"* et *"square"*. Que remarquez-vous comme changement dans l'affichage de la page?

.....

.....

F. <table>...</table> ; <tr>...</tr> ; <td>...</td> :

Ici, nous présentons les trois éléments HMTL fondamentaux dans la construction d'un tableau.

- L'élément table permet de définir un tableau. Il est défini par ses deux balises ; son contenu doit définir les lignes et les cellules du tableau.

En HTML, un tableau est défini par ses lignes ; puis chaque ligne par les cellules qu'elle contient :

- Chaque ligne est définie par un élément tr.

Dans une ligne, chaque cellule est définie par l'élément td.

Tous ces éléments sont définies par leurs deux balises ouvrantes et fermantes.

Exercice 15

On considère le code suivant :

```
1 <table>
2  <tr><td>1</td><td>2</td><td>3</td></tr>
3  <tr><td>2</td><td>3  3  3</td><td>4<br>4<br>4</td><↩
     /tr>
4 </table>
```

1. Répondez aux questions suivantes :
   a. Combien de lignes contient ce tableau?

.....

   b. La première ligne contient combien de cellules?

.....

   c. Quel est le contenu HTML de la seconde cellule de la seconde ligne?

.....

Les balises fermantes des éléments tr et td sont optionnelles.

Il est donc possible de les omettre dans l'écriture du code afin d'aléger celui-ci.

2. En utilisant la remarque ci-dessus, ré-écrivez le code dans un nouveu fichier "*.html*" en omettant les balises fermantes </tr> et </td>.

3. Ci-dessous est présentée une série d'attribut et leurs valeurs associées. Ecrivez-les un-à-un dans la balise ouvrante de l'élément table ; pensez à chaque fois à afficher votre page dans un navigateur afin de noter les les modifications apportées.
   a. *align*="*center*"

.....

   b. *border*="*1*"

.....

   c. *width*="*50%*"

.....

   d. *cellspacing*="*3*"

.....

e. *cellpadding="10"*

.....

L'attribut *align* permet de définir l'alignement du tableau dans la page ; sa valeur est à choisir parmi :

left ; *center* ; *right*

Les attributs *border*, *cellspacing* et *cellpadding* prennent pour valeur des entiers représentant une mesure en pixels. *cellspacing* permet de contrôler l'espacement des cellules entre elles alors que l'attribut *cellpadding* contrôle l'espacement entre la bordure d'une cellule et son contenu.

L'attribut *width* représente la largeur souhaitée par l'auteur du tableau. Le navigateur s'adapte à cette demande mais également à la taille du contenu de chaque cellule. Sa valeur est soit un entier exprimant une taille en pixel, soit un pourcentage représentant la taille du tableau relativement à l'élément parent *(ici body)*

4. Dans la seconde cellule de la première ligne, ajoutez l'attribut ci-dessous et affichez de nouveau la page dans un navigateur :

   align="right"

5. Dans cette même cellule, ajoutez l'attribut suivant :

   colspan="2"

   Quel est l'effet de cet attribut?

.....

.....

6. Ajoutez à l'élément tr définissant la seconde ligne l'attribut suivant :

   valign=top

   Expliquez l'effet de cet attribut sur la dernière ligne.

.....

.....

7. Ajoutez dans la même cellule, l'attribut suivant :

   height=200

   Que remarquez-vous à l'affichage de cette page dans un navigateur?

.....

- Les attributs *align* et *valign* sont applicables aux éléments tr et td et déterminent la position du contenu d'une cellule dans celle-ci.

- L'attribut *align* prend ses valeurs parmi *left*, *center*, *right*.
- L'attribut *valign* prend ses valeurs parmi *top*, *middle*, *bottom*.

- L'attribut *colspan* définit avec une valeur de *2*, utilisé sur une cellule, permet à un élément td de s'étendre sur deux colonnes.
  Dans l'exemple ci-dessus, cela implique que la première ligne comprend quatre cellule.
  L'attribut *rowspan* existe également pour étendre une cellule sur plusieurs lignes.
- L'attribut *height*, appliqué à une cellule ou à une ligne, définit la hauteur mimimun de celle-ci.

VI. La structure d'une page Web :

Tout au long de cette formation, nous nous sommes contentés d'écrire dans un fichier "*.html*" vide directement le code source de notre page, mais on ne respecter pas la structure de base d'une page Web ; heureusement que les navigateurs complétés par eux-mêmes nos oublis.

Une page Web est composée de deux parties :

- L'entête : il contient toutes les informations destinées au navigateur tel que le nom de l'auteur, la date de création, le codage des caractères utilisés...
  Ces informations ne sont pas affichées dans le navigateur.
  L'entête est définie par l'élément head .
- Le corps du document : il contient tous les éléments HTML affichable à l'écran ; il représente l'espace affiché dans la fenêtre du navigateur.
  Le corps du document est défini par l'élément body .

A l'origine, les concepteurs du Web pensaient que les informations transmises allaient, dans le futur, transporter du HTML mais d'autres types de données ; ainsi il créère l'élément html qui doit contenir tout le code HTML : plus précisemment, l'entête et le corps du document.

Ainsi, voici la structure minimale correcte d'une page Web :

```
<html>
 <head>
 </head>

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

Exercice 16

La balise body ne demande aucune connaissance particulière. Tous les codes HTML que vous avez déjà saisis, auraient dû être incorporés à l'intérieur de cet élément.

Dans cette exercice, nous allons donc préciser quelques utilisations de l'élément head :

1. Récupérez le fichier "*_exerciceC.html*" présent dans le dossier des exercices et éditez-le à l'aide d'Emacs.

2. a. Rajoutez dans l'entête de la source *(l'élément head)* l'élément meta suivant :
   `<meta name="author" content="Wikipédia">`.

   b. Remarquez-vous une différence à l'affichage de votre page dans un navigateur :

.....

   c. Dans FireFox, actionnez la commande :
   Outils ⇝ Informations sur la page
   Retrouvez l'information incorporée précédemment dans l'entête de votre page.

3. Rajoutez la balise suivante dans l'entête de votre page :
   `<meta http-equiv="Content-Type" content="text/html; charset=Shift_JIS">`
   Que remarquez-vous lors de l'affichage de la page dans un navigateur?

.....

.....

4. De même avec la balise suivante :
   `<meta http-equiv="refresh" content="5, http://google.fr">`

.....

.....

L'entête du document peut également contenir :

- L'élément style qui permet, comme nous allons le voir prochainement, d'écrire

des feuiles internes de style CSS.

- L'élément link qui permet de lier à la page courante des feuilles externes de styles CSS ou des scripts externes **JavaScript**.
- L'élément script permettant de définir des scripts incorporés dans la page.

Voici quelques autres exemples d'utilisation de l'élément meta :

- L'élément meta, utilisé comme ci-dessous, aide les moteurs de recherche à référencer votre page Web :

  `<meta name="keyword" content="éditeur,texte,scientifique">`

- On autorise, ci-dessous, les navigateurs à garder cette page jusqu'au 29 Novembre 2009 évitant ainsi au client de recharger la page lors d'une future visite :

  `<meta http-equiv="Expires" content="Thu, 29 Nov 2009 16:18:42 GMT">`

- Pour indiquer au navigateur que les caractères occidentaux doivent être affichés, on insère la balise suivante dans l'entête du document :
  `<meta http-equiv="Content-Type" content='text/html; charset=ISO-8859-1'>`

VII. Les feuilles de styles :

A. Les styles en ligne :

Exercice 17

Nous avons déjà eu l'occasion, avec KompoZer, d'utiliser les styles CSS ; cet exercice va nous permettre de faire rapidement un tour des propriétés de style les plus fréquemments utilisés. Il faut, dès à présent, s'habituer au nom de ces propriétés et aux différentes valeurs autorisées.

1. Ouvrez le fichier "*_exerciceD.html*" avec votre navigateur.
2. Cliquez sur chaque propriété de style pour observer son effet sur les élément de la page.
3. a. Affectez une bordure 'a l'élément contenant "*Les Romains...*".

   b. Modifiez les valeurs des propriétés ***padding*** et ***margin***. Quelle est la différence entre ces deux propriétés?

. . . . . .

. . . . . .

Voici la structure simplifié de cette page :

```
<table>
  ...
</table>

<div id="premier">
 Premier
 <div id="second">
   Les Romains ...
 </div>
</div>
```

La plupart des styles modifient l'élément div ayant pour attribut `id` la valeur *second* ; le fait qu'il soit contenu dans un autre élément div permet de mieux mettre en évidence la différence entre les propriétés ***padding*** et ***margin***.
Les propriétés ***table-layout***, ***table-collapse***, ***border-spacing***, ***empty-cells*** agissent directement sur l'élément table de notre page.
La propriété ***cursor*** influence le curseur de notre souris lorsque celui-ci passe au-dessus de l'élément *second*.

Exercice 18

Tous les éléments HTML, ou presque, acceptent l'attribut `style` ; celui-ci permet d'affecter directement un style CSS à un élément HTML. Sa valeur est une déclaration de styles : elle regroupe plusieurs propriétés de styles affectées de valeurs.

Voici la forme générale d'une déclaration de style :

$$\underbrace{\textbf{\textit{margin-bottom}}}_{\text{Propriété}} : \underbrace{8cm}_{\text{Valeur}} ; \underbrace{\textbf{\textit{font}}}_{\text{Propriété}} : \underbrace{1pt\ Arial}_{\text{Valeur}}$$

Les propriétés sont séparés entre elles par un point virgule "**;**".
Chaque propriété est définie par le couple $^{\text{propriété}}/_{\text{valeur}}$; on sépare la propriété de sa valeur par deux points "**:**".

Voici un exemple d'utilisation de l'attribut `style` dans la balise ouvrante d'un élément :

```
<div style="font-weight:900;background-color:#AAAAAA">
Un exemple de conteneur affecté d'un style
</div>
```

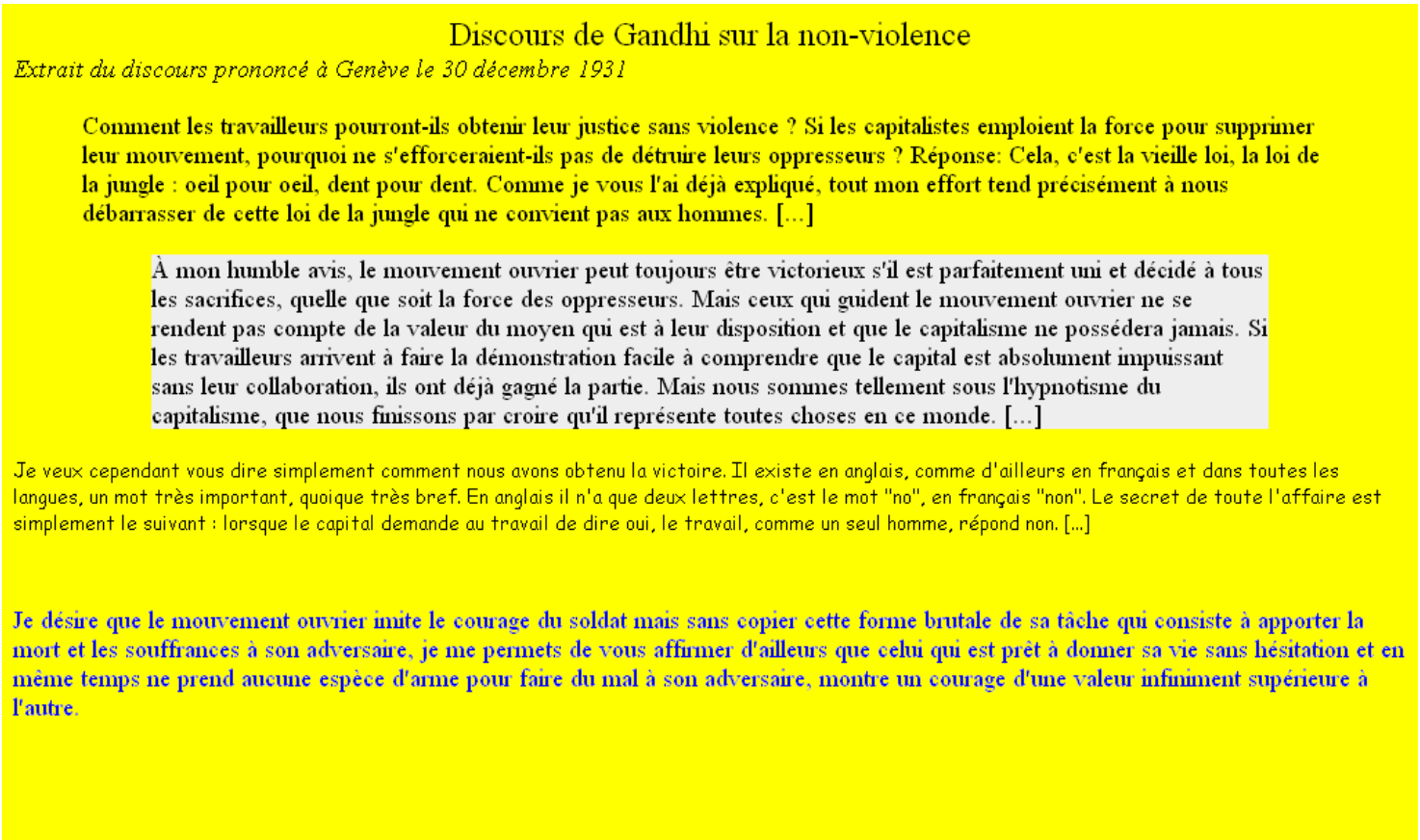
1. Editez le fichier "*_exerciceE.html*" à l'aide de Emacs.

Remarquez que le texte a été découpé en paragraphe à l'aide d'éléments `div` et que chacun d'eux possède un attribut `style` qui pour l'instant ne possède aucune valeur.

Le but de l'exercice est de formater cette page en utilisant uniquement les styles CSS. Voici quelques références à votre cours sur les styles CSS qui vous permettra de formater cette page :

- Les propriétés sur la gestion des polices et des fontes *(page 4)*
- Les propriétés de gestion de paragraphes *(page 7)*
- Les propriétés d'espacement des éléments *(page 9)*
- Les propriétés pour gérer la couleur de fond des éléments *(page 12)*

2. Utilisez l'attribut `style` de chaque élément pour formater cette page à l'image de la capture d'écran ci-dessous :



B. Positionnement et apparence :

Nous allons maintenant regarder les propriétés de positionnement des éléments HTML ; ce sont les propriétés de styles les plus importantes pour formater correctement notre page.

Exercice 19

Le fichier "*_exerciceF.html*" présente un code HTML similaire à celui-ci :

```
1 <div style="background-color:pink">
2  Premier
3  <div style="background-color:yellow">
4   Second
5  </div>
6  Fin Premier
7 </div>
```

Voici la structure de cette page :

- Cette page est composée d'un `div` dont le fond de couleur est rose ; son contenu est délimité par les mots "*Premier*" et "*Fin Premier*".
- Un second élément `div` dont le fond est de couleur jaune est contenu dans le premier.

Ce fichier dispose d'un script **JavaScript** permettant de manipuler facilement les différentes propriétés de style liées à la position de ces deux éléments `div` dans la page.

Vous trouverez référence à ces propriétés de style à la page 13 de votre manuel de cours sur les CSS.

1. Fixez la valeur de la propriété de ***position*** du second éléments `div`, à la valeur *relative*. Notez-vous une différence?

.....

2. a. Modifiez successivement la valeur, pour le second `div`, de la propriété ***top***. Que remarquez-vous?

.....

.....

Fixez la valeur de cette propriéé à *150px*.

b. Modifiez successivement la valeur, pour le premier `div`, de la propriété ***top***. Que remarquez-vous?

.....

c. Fixez la valeur de la propriété ***position*** à la valeur *relative* ; modifiez de nouveau la

valeur de ***top*** de cet élément.

. . . . . .

. . . . . .

Les propriétés ***top***, ***left*** n'ont pas d'effet sur les éléments dont la propriété ***position*** a la valeur *static* ; les éléments ayant pour position *static* sont placés par le navigateur sans interaction avec les styles CSS :

- Un élément de type *block* crée une nouvelle ligne dans le navigateur ;
- Un élément de type *inline* est placé à la suite des autres éléments, de même type sur une même ligne ; le navigateur s'occupe de remplir les lignes et d'insérer automatiquement les retours à la ligne

En position *relative*, le second élément div conserve sa place dans le premier mais les propriétés ***top*** et ***left*** permettent de le déplacer vis-à-vis de sa position initiale.

3. Le premier div doit être en position *relative* et donnez à la propriété ***top*** du second div la valeur *0px*.

a. Mettez la valeur de ***position*** à *absolute*. Que remarquez-vous?

. . . . . .

. . . . . .

b. Donner, successivement, à la propriété ***position*** du premier élément toutes les valeurs. Concentrez-vous alors sur la position du second élément. Que pouvez-vous dire de cette position

En position *relative*, un élément est placé "*naturellement*" dans la page ; sa place initialement prise est conservée mais sa position peut être modifiée avec les propriétés ***top*** et ***left***.

En position *absolute*, aucune place n'est réservée à l'élément, mais il reste à comprendre le principe de positionnement de tels objets ; la question b. permet d'observer le positionnement du second élément div, lui-même en position *absolute* et ayant la valeur *0px* à la propriété ***top***, lorsqu'on modifie le positionnement du premier div ; voici la règle utilisée :

> `La position d'un élément en position "`*absolute*`" est calculé relativement au premier élément-parent ayant un positionnement non-`*static*

- Lorsque le premier élément div est en position *relative* ou *absolute* alors le second élément suit la position du premier.
- Lorsque le premier élément est en position *static*, le second ne possède qu'un seul

élément parent en position non-*static* : c'est l'élément `body` ; expliquant ainsi que le second `div` se trouve tout en haut de la fenêtre d'affichage du navigateur.

4. Appuyez sur la touche "*F5*" pour rafraîchir votre page et revenir aux définitions de base de cette page.

   a. Faîtes varier la valeur de la propriété ***display*** du second élément. Que pouvez-vous dire des trois valeurs possibles de cette propriété :

.....

.....

.....

.....

   b. Modifiez la valeur de la propriété ***width*** respectivement quand le second `div` se trouve en affichage *block* et en affichage *inline*. Quelle conclusion tirez-vous de cette expérimentation?

.....

.....

Un élément HTML possède trois modes principaux d'affichage :

- *none* : l'élément n'est pas affiché et aucune place ne lui est réservée.
- *block* : l'élément est affiché sur toute la largeur de la fenêtre du navigateur.
- *inline* : l'élément est affiché et sa largeur correspond à celle de son contenu. Il est placé dans le flux courant du texte : à la suite du contenu le précédant et sur la ligne courante.

Chaque élément HTML possède initialement un affichage précis dépendant de sa nature :

- `div` `p`, `form` `pre`... sont des objets s'affichant en *block* ;
- `span`, `img`, `b`... ont pour affichage par défaut *inline* ;
- L'élément `table` a un affichage de *table* ;
- L'élément `head` a un affichage par défaut de *none*.

Ces valeurs sont tirées du fichier `html.css` utilisé par FireFox pour définir par défaut le style de chaque élément HTML : c'est sa feuille de style par défaut.

5. Cliquez sur la touche "*F5*" pour rafraîchir l'affichage de votre page et repartir des paramètres par défaut de cette page. Comparez alors la valeur *hidden* de la propriété ***visibility*** et la valeur *none* de ***display*** :

.....

.....

Exercice 20

En position *absolute*, les éléments peuvent se chevaucher ; nous allons voir dans cet exercice, les règles utilisées par un navigateur pour décider quels éléments seront affichés au premier plan ou à l'arrière plan.

1. Editez le fichier "*_exerciceI.html*" avec Emacs et affichez-le également dans un navigateur.

2. a. Recherchez l'élément body ; quel est le premier élément div déclaré dans ce code?

.....

b. Par défaut, quel est l'élément div qui est affiché au premier plan?

.....

3. La propriété ***z-index*** définit la profondeur à laquelle est placé un élément. Un code **JavaScript** permet de modifier, pour chaque élément div, la valeur de cette propriété en cliquant directement sur les éléments.

   Faîtes suffisamment de test et pensez à la touche "*F5*" pour réinitialiser votre affichage, afin de répondre correctement aux questions suivantes :

   a. Si deux éléments positionnés en *absolute* ayant des valeurs de ***z-index*** distinctes se chevauchent, lequel est affiché au premeir plan?

.....

.....

   b. Lorsque deux éléments ayant le même ***z-index*** se chevauchent quel élément est affiché au premier plan?

.....

.....

C. Les feuilles de styles internes et les sélecteurs :

Pour l'instant, nous avons inscrit les déclarations de styles dans la balise ouvrante de l'élément recevant les propriétés de styles ; ce sont les déclarations de *styles internes*.

Nous avons déjà vu dans KompoZer la possibilité d'utiliser les ***feuilles de styles*** pour centraliser les déclarations de styles hors des éléments et permettant ainsi à plusieurs éléments d'une page de recevoir une même déclaration de styles.

Chaque déclaration de style est alors précédée d'un sélecteur qui permet de désigner les éléments recevant les propriétés de styles de cette déclaration : chaque couple $^{\text{sélecteur}}/_{\text{déclaration}}$ s'appelle une ***règle***. Voici un exemple de règle :

$$\underbrace{\underbrace{.\text{chapitre}}_{\text{Sélecteur}}\left\{\underbrace{\textbf{\textit{margin-bottom}} : 8cm; \textbf{\textit{font}} : 1pt\ Arial}_{\text{Déclaration}}\right\}}_{\text{Règle}}$$

Il existe deux manières de créer des feuilles de styles CSS :

- Les ***feuilles de styles internes*** sont insérés dans chaque entête *(l'élément* `head` *)* de page à l'aide de l'élément `style`.
- Les ***feuilles de styles externes*** sont des fichiers externes aux pages Web, généralement possédant une extension "*.css*" contenant des règles de styles. Pour se lier à une feuille de style externe, une page Web utilise l'élément `link`.

Exercice 21

1. Editez le fichier "*_exerciceG.html*" à l'aide d'Emacs. Remarquez que le module "*HTML Validator*" vous signale un certain nombre d'erreurs dans cette page :
   a. Rajoutez les éléments `html`, `head`, `body` afin de redonner une structure correcte à votre page Web.
   b. Sauvegardez ce fichier édité dans Emacs; puis, appuyer dans "*HTML Validator*" sur la touche "*F5*" pour actualiser le contenu de cette fenêtre. Rajoutez le "*DocType*" proposé pour compléter votre page Web.

2. Dans l'entête de cette page *(l'élément* `head` *)*, rajoutez un élément `style` à l'intérieur duquel, vous ajouterez une à une les règles suivantes, sans oublier d'observer l'effet de chacune sur le document :
   a. `body` $\{$ ***background-color*** : *yellow* $\}$

. . . . . .

. . . . . .

   b. `#titre` $\{$ ***text-align*** : *center* ; ***font-size*** : *24pt* ; ***font-style*** : *italic* ; ***margin*** : *20px* $\}$

. . . . . .

. . . . . .

   c. `.chapitre` $\{$ ***font-size*** : *18pt* ; ***text-decoration*** : *underline* ; ***margin*** : *40px 0px 20px* $\}$

. . . . . .

. . . . . .

Voici quelques explications sur les sélecteurs utilisées ci-dessus et leurs liens avec la page Web utilisée :

a. Lorsque le sélecteur d'une règle est le nom d'un élément, tous ces éléments de la page recevront les déclarations de styles de cette règle ; ici, seul l'élément body est affecté par cette règle.

b. Lorsqu'un sélecteur est constitué d'un dièse suivi d'un nom, cette règle vise qu'un élément dans la page : celui qui possède l'attribut `id` ayant ce nom pour valeur ; ici, seul l'élément div en début de document ayant l'attribut ci-dessous est affecté :

`id`=*titre*

c. Lorsqu'un sélecteur est constitué d'un point suivi d'un nom, cette règle appliquera ses déclarations de styles à tous les éléments possédant l'attribut `class` ayant ce nom pour valeur ; ici, tous les éléments de la page ayant l'attribut ci-dessous sont affectés :

`class`=*chapitre*

Servez-vous de la remarque précédente pour définir, dans la feuille interne de cette page, les règles de styles suivantes :

3. a. Cette règle comporte les déclarations de styles suivantes :
margin: *0px 50px* ; ***border***: *6px red groove* ; ***background-color***: *tomato* ; ***padding***: *5px*
Elle doit sélectionner tous les éléments de la page ayant la valeur *boite* pour l'attribut `class`

b. Combien d'éléments cette règle a affecté d'éléments dans la page?

.....

4. a. Cette règle comporte la déclaration de styles suivantes :
background-color: *#CCCCCC*
et elle doit affecter tous les éléments input de la page.

b. Combien d'éléments cette règle a affecté d'éléments dans la page?

.....

5. a. Cette règle comporte la déclaration de styles suivantes :
text-align: *center* ; ***font-size***: *150%*

b. Combien d'éléments cette règle a affecté d'éléments dans la page?

.....

Exercice 22

Il est possible qu'un élément hérite de plusieurs sélecteurs différents de plusieurs valeurs pour une même propriété ; nous allons étudier la manière dont les navigateurs choisissent d'affecter une valeur plutôt qu'une autre à une propriété :

1. Dans un fichier "*.html*" vide, écrivez la structure complète d'une page HTML vide.

2. a. Complétez l'élément `body` de sorte que celui-ci contienne quatre éléments `div` ayant les caractéristiques suivantes :

- Le premier élément `div` ne possède aucun attribut ;
- Le second `div` possède l'attribut *id* avec la valeur *monId* ;
- Le troisième `div` possède l'attribut *class* avec la valeur *maClasse* ;
- Le dernier `div` possède deux attributs : l'attribut *class* avec la valeur *maClasse* et l'attribut *id* avec la valeur *monId2*.

Pensez à ajouter un contenu de votre choix à chaun de ces éléments pour les identifier facilement.

b. Affichez votre page dans un navigateur ; la page doit être blanche et doit comporter quatre lignes.

3. Placer la feuille interne de style suivante dans votre code source :

```
<style type="text/css">
  body{background-color:red}
  div{background-color:blue}
  .maClasse{background-color:green}
  #monId{background-color:pink}
  #monId2{background-color:linen}
</style>
```

Puis, affichez cette page dans votre navigateur.

a. Chaque élément `div` de la page reçoit une ou plusieurs valeurs pour la propriété de style ***background-color*** ; notez, pour chaque `div` le sélecteur qui a imposé sa valeur :

- Premier `div` :

. . . . . .

- Second `div` :

. . . . . .

- Troisième `div` :

. . . . . .

- Quatrième `div` :

. . . . . .

Voici, par ordre décroissant, l'ordre de sélection de la valeur en fonction du sélecteur utilisée :

- Sélecteur à l'aide de l'attribut *id*.
- Sélecteur à l'aide de l'attribut *class*.
- Sélecteur en fonction de la nature de l'élément.

4. a. Ajoutez à la seconde ligne de votre feuille de style interne le mot clef *!important* pour qu'elle devienne :

div{ ***background-color*** : *blue!important* }

b. En affichant la page dans votre navigateur, quelles sont les remarques que vous pouvez apporter :

.....

.....

Le mot clef *!important* infléchit l'ordre naturel de sélection de la valeur d'une propriété par le navigateur.

Le fait qu'une propriété de style puisse recevoir plusieurs valeurs illustre le nom de feuille de style en cascade des styles CSS ; en fait, la variété des sources que peut recevoir un élément est grande *(voir cours sur CSS page 30)* ; mais, toutes ces interactions sont assez transparentes pour le programmeur commun.

Exercice 23

Appliquons maintenant les règles de cascade de styles citées dans l'exercice précédente dans le code ci-dessous :

```
1  <html>
2   <head>
3    <style type="text/css">
4     body{background-color:#EEEEEE;color:black;font-size:10↩
          pt}
5     div{color:red;font-size:14pt}
6     span{color:blue}
7     #monId{color:green}
8     #monId2{color:pink;font-size:24pt}
9    </style>
10  </head>
11  <body>
12   Bonjour tout le monde,
13   <div>
14    On commence ce cours
15     <span>du jeudi</span>
16     <span id="monId">par des maths</span>
17   </div>
```

```
18    Ne vous inquiétez pas
19    <span class="maClasse">Ce sera facile</span>
20    <div class="maClasse" id="monId2">Merci beaucoup</div>
21    A une prochaine
22  </body>
23 </html>
```

1. Déterminez les valeurs des propriétés ***color*** et ***font-size*** héritées pour chacune des parties suivantes du document :
   - "*Bonjour tout le monde,*" :
     color : ..... ***font-size*** : .....
   - "*On commence ce cours*" :
     color : ..... ***font-size*** : .....
   - "*du jeudi*" :
     color : ..... ***font-size*** : .....
   - "*par des maths*" :
     color : ..... ***font-size*** : .....
   - "*Ne vous inquiétez pas*" :
     color : ..... ***font-size*** : .....
   - "*Ce sera facile*" :
     color : ..... ***font-size*** : .....
   - "*Merci beaucoup*" :
     color : ..... ***font-size*** : .....
   - "*A une prochaine.*" :
     color : ..... ***font-size*** : .....
2. Affichez le fichier "*_exerciceH.html*" dans votre navigateur ; il contient le code présenté dans cet exercice. Comparez vos réponses à la question précédente avec l'affichage dans le navigateur.

D. Les feuilles de styles externes :

Les feuilles de styles externes sont de simple fichier texte contenant des règles de styles.

Elles présentent un avantage sur les feuilles de styles internes : elles peuvent partager leurs règles de styles avec plusieurs pages Web ; elles sont très utiles pour homogéniser la présentation d'un site.

Pour lier une feuille de styles externe à une page Web, on ajoute, dans l'entête de cette page, l'élément link *(définit par sa seule balise ouvrante)* paramétrer de la façon suivante :

`<link rel=stylesheet type=text/css href="...">`

où l'attribut ***href*** indique l'URL relative ou absolu de la feuille de style externe à lier.

Exercice 24

1. A partir du CD de la formation et dans le dossier `d-exerciceHtmlCss`, recopier la version originale de "*_exerciceG.html*" ainsi que le fichier `style.css` se trouvant dans ce dossier.
2. Editez le fichier "*_exercuceG.html*" et rajoutez les éléments html , head , body pour reconstruire la structure d'une page Web.
3. Affichez votre page dans un navigateur et observez qu'aucun style CSS n'est appliqué à cette page.
4. Rajoutez dans l'entête de cette page, un élément link liant la feuille de style externe "*style.css*" présente à la racine du dossier des exercices.
5. Affichez votre page dans un navigateur.
6. Modifiez quelque peu les règles se trouvant dans le fichier "*style.css*", puis réactualisez l'affichage du navigateur en appuyant sur la touche "*F5*".