

PRESENTATION

Avec l'apparition du **World Wide Web**, en 1989, le **HTML** est choisi comme langage universel, multi plateforme afin de transmettre les scripts structurant les pages WEB. le protocole HTTP (Hyper Text Transfert Protocole) est le protocole qui permet d'acheminer les pages au navigateur qui interprète le code et affiche la page à l'utilisateur.

HTML (Hyper Text Markup Language) est un langage de script qui définit la structure des documents, leurs attributs, les liens entre les différentes pages du WWW, l'importation des objets dans la page (images, fond de page, son, vidéo) ainsi que les éléments d'interface tel boutons, boîtes de dialogues... Pour ceci le langage utilise des balises qui permettent de définir l'ensemble de la structure des données.

Le web est basé sur la notion de client/serveur. L'utilisateur accède au serveur par le biais du protocole HTTP en entrant l'adresse du serveur voulu (url pour *Uniform resource locator* de type `http://www.titre de la page.domaine`). Les données sont transmises au navigateur qui interprète le script source et charge les données.

Le script HTML définit tout les éléments de la page. la structure même du script, les éléments liés à l'indexation des pages dans les moteurs de recherche, la mise en forme du texte, les éléments décoratifs, la mise en page de chaque document(marges, tableaux, listes etc.).

Pour s'adonner aux joies de la création de pages Web et pourquoi pas d'un site entier plusieurs solutions (sérieuses) s'offrent à nous :

- Utiliser un éditeur html (clic là puis clic là puis... oh la belle bleue !!!)
- Utiliser un éditeur de script (déjà plus sérieux)
- Utiliser un simple éditeur de texte (bloc note de windaube comme moi par exemple)
- Prendre une page web qui nous plait et faire une copie/collé du source avant de débroussailler tout ça (ça se voit !!!)

Dans tous les cas il faudra au final avoir une (des) pages incluant le code html, qu'il soit généré par un RAD ou par votre clavier 102 touches préféré et donc (voilà le but de mon propos) être capable de déchiffrer la structure de la page, ce qui veut dire :

ON NE DOIS PAS FAIRE L'IMPASSE SUR QUELQUES NOTIONS DE CODE !!!

au risque de se retrouver un peu paumé au moindre problème d'affichage ou autre.

Sans rire je concède qu'on puisse faire appel à un logiciel permettant de générer des pages web et de réaliser de véritables projets web mais il me paraît aussi absurde d'utiliser ce genre de produit sans savoir ce qui est généré par icelui que de se servir d'une calculette sans savoir ce qu'est une soustraction. voilà c'est dit ! ;-)

Si vous utilisez la bonne vieille méthode de l'éditeur de texte pour écrire vos pages, vous les sauvegarderez avec l'extension .html ou .htm au choix. **Optez une fois pour toute** pour l'une ou l'autre et rappelez vous que les liens ne fonctionneront pas si vous ne mettez pas la bonne extension dans la balise de liens hypertexte.

Voici donc quelques éléments sur la structuration d'une page WEB

EN TETE DE PAGE

<head> EN TETE </head>

Caractéristiques :

- Ne s'affiche pas dans la page
- Contient le titre de la page entre deux balises < titre>
- Contient les éventuelles déclarations globales : css, php, javascript.
Balise < style>, < script>...
- Contient les balises meta qui détermine la nature de la page, son contenu ...

EXEMPLE :

```
<html> <!--balise définissant le document comme un script html -->

<head>          <!-- En tête de la page  -- >

<title>INDEX</title><!-- titre de la page, il s'affichera dans la barre
de déplacement du navigateur-->

<!--déclaration des meta tag                                -- >
<meta HTTP-EQUIV="Content-Type" CONTENT="text/html; charset=iso-8859-1">
<meta HTTP-EQUIV="Content-language" CONTENT="fr">
<meta NAME="expires" CONTENT="never">
<meta NAME="author" CONTENT="EMMANUEL BESNARD">
<meta NAME="Rating" CONTENT="general">
<meta NAME="robots" CONTENT="all">
<meta NAME="Keywords" CONTENT="">

<style type="text/css">
</head><!--fin de l'en tête de la page                                -- >
```

<html>	Balise toujours suivi d'une balise fermante. Si balise unique finir avec / <hr />
Commentaires	Ne seront pas affiché dans la page. Utile pour annoter son script
Attribut= « valeur »	Attributs et valeurs d'attributs

CORPS DE DOCUMENT

<body>CORPS</body>

C'est évidemment ici que la structure de la page ainsi que son contenu (texte, images, bannières fond d'écran seront insérés, soit directement (texte) ou pour les objets extérieurs (images, sons etc. ...) le lien vers leur localisation.

Le principe est simple :

<élément attribut="valeur">----- contenu ----- </élément>

LE DOCUMENT SE FINIT EVIDEMMENT PAR LA BALISE FERMANTE : </html>

```
<html>
<body>
<center><h1><font color= #2222BB >
MA PAGE WEB
</font color></h1></center>

<p align= left font size="5">
Hello à tous, voici ma page web personnelle ou vous trouverez mon CV, les recettes de<br />
cuisine de tante Danièle et les photos de mes dernières vacances à Patatras les flots<br />
</font size>
</p>
<p align= "center">
<font color= "#5588bb"><font size="6">
Voici la photo de ma maison de vacances<br /><br />
</font size></font color>
<img src= "maison.jpg" align= "center" border="5" alt="photo de ma maison">
</p>

<p align="right">
Cette page est créée par Marcel DUPONT.
</p>
</body>
</html>
```

IL RESTE, UNE FOIS QUE LA STRUCTURATION DE LA PAGE EST COMPRISE, A APPRENDRE LES BALISES LES PLUS COURANTES AFIN DE COMMENCER A ECRIRE SON CODE

LES PRINCIPALES BALISES

Il faut noter que même si les balises html sont reconnues par les navigateurs (à quelques exceptions près), beaucoup d'entre elles sont aujourd'hui considérées comme déclassées et supplantées par les mises en forme organisées en cascading style sheet. (css)

PRINCIPE DES BALISES HTML (TAG)

On utilise les noms de balises pour encapsuler le texte ou les objets à mettre en forme.

Les navigateurs ne sont pas sensibles à la casse aussi `<BR>` est équivalent à `<br>`.

Cependant, les nouveaux standards se développant xhtml et xml sont eux sensibles à la casse. Pour assurer une meilleure portabilité de vos codes source je conseille d'écrire les balises en minuscule.

Les balises sont en principe ouvrantes et fermantes :

`< balise > ..... < / balise >`

NB les balises simples sont terminées par un `" / "` . ex : `< br / >`

Non obligatoire en HTML 4.1 (1999), ce paramètre deviendra obligatoire dans les nouvelles versions (XHTML). Il est donc conseillé de s'y conformer dès à présent.

Dans la plupart des cas les balises possèdent des attributs qui vont agir sur l'élément mis en forme

La syntaxe est la suivante :

`< balise attribut="valeur" > ..... < / balise >`

Voici par type de structuration les principales balises et leur attributs

Structure de page :

- `< html> < /html>`
- `< head> < /head>`
 - `<title>` titre dans la barre de titre du navigateur `</titre>`
 - `<script </script>` déclaration de script (ex :javascript)
 - `<style </style>` déclaration de style css
 - `<meta> ... </meta>` vos méta-tags
- `< body> < /body>`
 - `Background ="location de l'image"` : définit une image en arrière plan
 - `bgcolor ="#xxxxxx"` : définit la couleur de fond du document
 - `text ="#xxxxxx"` définit la couleur de base du texte
 - `link, alink, vlink ="#xxxxxx"` définit la couleur des liens hypertextes

Mise en forme de la page

Le document HTML simple possède malgré tout peu de possibilités de mise en page. Les différents éléments d'une page web se succèdent les uns au dessous des autres successivement (position relative). Les navigateurs ne connaissent pas le retour de chariot aussi le passage à la ligne du dessous se fait :

- Automatiquement après une balise header <Hn > titre de 1 à 6
- Automatiquement après une balise paragraphe <p > ... </p >
- Automatiquement après une balise de liste simple ou ordonnée ... ou ...
- Automatiquement après une balise <hr> qui trace une ligne dans la page - attributs width="n" size="n" align ="left" "center" right"
- Après une balise
 (break)

L'attribut align="" avec comme valeur left,center,right permet de positionner les éléments de la page sommairement

ex : <p align="center"> centre le paragraphe dans la page </p>

centre le paragraphe dans la page

La balise <hr > permet de tracer dans la page une ligne de séparation



Lorsque l'on souhaite effectuer une mise en page plus sophistiquée, le html simple permet un quadrillage de la page sous forme de tableau qui permettra de placer des éléments à un endroit voulu. (voir le paragraphe consacré aux tableaux - html deuxième partie).

L'utilisation des feuilles de style rend aujourd'hui obsolète la plupart de ces balises de mise en page. Le CSS permet en effet le positionnement d'éléments au pixel près dans la page et avec un positionnement absolu ce qui permet la juxtaposition, la superposition de ces éléments. Voir la présentation des feuilles de style en fin de document.

Mise en forme de texte :

Définit les polices, leurs tailles, et leurs mise en forme

- ... : attributs color="#xxxxxx"
 - attribut face ="type de police" helvetica, arial, verdana ...
 - attribut color="#xxxxxx"
 - attribut size ="taille de police" de 1 (plus petit) à 7 (plus grand)
- <basefont > ... </basefont> : définit le type de police par défaut, mêmes valeurs que font
- <big > ... </big> : police plus grande que standard
- <small > ... </small> : police plus petite que standard
- ... : mets le texte en gras
- <u > ... </u> : souligne le texte
- <i> ... </i> : mets le texte en italique
- ... : souligne le texte
- <strike> ... </strike> : texte barré
- ^{...} : texte exposant
- _{...} : texte indicé

LES LISTES

HTML permet de générer facilement des listes simples ou numérotées

- listes simple avec leurs attributs `< ul >` liste non ordonnée `</ul >`
- listes ordonnées avec leurs attributs `< ol >` liste ordonnée `</ol >`

Chaque nouvel item débute par `<li>` et se termine conformément au xhtml par `</li>`

Il existe plusieurs type de puce qui seront indiquées avec l'attribut `type="valeur"`

- disc : Les puces sont en forme de disques
- circle : Les puces sont en forme de cercles
- square : Les puces sont en forme de carrés

Dans les listes ordonnées, le type définira l'item souhaité

- `type="a"` : Minuscules
- `type="A"` : Majuscules
- `type="i"` : Chiffres romains (minuscules)
- `type="I"` : Chiffres romains (majuscules)
- `type="1"` : Nombres (par défaut)

LES COULEURS DU WEB

Représentation des couleurs par un identificateur "nom_couleur"

- nom_couleur = "blue"; "green"; "yellow"; "navy"; "purple"
- exemple (purple) --> violet; (navy) --> bleu marine; (orange) --> orange

Codage des couleurs :

- Trois longueurs d'onde :

ROUGE **VERT** **BLEU**

- Deux valeurs pour chaque longueur d'onde en hexadécimal
 - 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F
 - les couleurs se représentent avec un # devant, ex : #FFFFFF
 - 00 = 0% de la couleur
 - FF = 100% de la couleur
 - mélange des trois valeurs

ROUGE	VERT	BLEU	CODE	COULEUR
100%	100%	100%	#FFFFFF	BLANC
0%	0%	0%	#000000	NOIR
100%	0%	0%	#FF0000	ROUGE
0%	100%	0%	#00FF00	VERT
0%	0%	100%	#0000FF	BLEU

Un exemple :

<body bgcolor = «#ff0000 » et votre fond de page web recevra

100 % de rouge

0% de vert

0% de bleu

Donc rouge vif

<body bgcolor = « red » aura le même effet

LES LIENS ET ANCRES

Comme vous le savez sans doute, le WEB repose sur la notion de lien hypertexte et il est évident qu'aucun site ne peut se concevoir sans ces renvois vers les différentes pages de votre site ou vers des pages extérieures.

Le liens hypertexte est extrêmement facile à mettre en place et la difficulté principale viendra certainement plus d'une arborescence claire de votre site et de sa navigation interne que de la syntaxe utilisée.

Il y a plusieurs types de liens hypertexte :

- Liens vers une autre page de votre site syntaxe `<a href="location de la page" >` l'élément cliquable `</a >`
- Liens vers un site extérieur syntaxe `<a href="url du site" ou adresse e mail >` l'élément cliquable `</a >`
- Lien à l'intérieur d'une même page (ancres) qui permet de se rendre à un élément de la page
 - On marque l'élément cible avec l'attribut name ou id : `<p name="cible" > ... </p>`
 - On crée le lien vers *cible* : `<a href="#cible" >` l'élément cliquable `</a >`

On a parfois intérêt à ouvrir le lien dans une nouvelle fenêtre (souvent les liens vers des sites extérieurs). On utilisera alors dans le liens l'attribut `target =" blank"` pour une nouvelle fenêtre. Les éléments cliquables sont en général du texte ou très souvent des images. Dans ce cas, l'attribut `border ="0"` vous affranchira d'une solide bordure encadrant votre objet.

Les couleurs des liens peuvent être définis dans la balise body tel que nous l'avons vu plus haut

- `link ="#xxxxxx"` lien
- `alink ="#xxxxxx"` lien actif
- `vlink ="#xxxxxx"` lien visité

Peu de choses à dire de plus sur ces liens si ce n'est que de leur utilisation à bon escient dépend la clarté de votre site, qu'il vaut mieux les vérifier minutieusement avant de mettre votre site en ligne.

Il est intéressant de savoir qu'une suite de plus de 5 liens (ce qui revient à descendre à 5 niveaux d'arborescence) a de grande chance de voir l'internaute aussi perdu que dans un labyrinthe inextricable, ce qui se solde bien souvent par un clic rageur sur la petite croix blanche sur fond rouge de votre navigateur. Il est extrêmement prudent de prévoir un lien vers votre page d'accueil dès que l'arborescence du site devient complexe.

LES TABLEAUX HTML

Les tableaux en html ont une double utilité. (avait ?)

Tout d'abord de présenter des informations structurées en lignes et colonnes, tout le monde sait que la clarté en est avérée.

Dans une deuxième mesure, le tableau permet de s'affranchir de la mise en page assez pauvre du code html simple en permettant de positionner des éléments quand on ne pourrait pas le faire avec des valeurs d'alignement aussi réduites que left, center ou right. (oui il y a aussi les « margin »).

J'entends par "html simple" l'utilisation des balises html sans recours aux feuilles de style (css) qui sont désormais quasiment indispensables à toute page web un peu complexe

Un tableau html est encadré par deux balises ouvrante et fermante <table> </table>

Une ligne de titre peut être insérée en premier lieu : <caption> </caption>

Chaque ligne du tableau est insérée avec la balise <tr> </tr> (*tr* = *table row*)

Les colonnes sont construites à l'intérieur de chaque ligne par la balise <td> </td> (*td* = *table data* ce qui constitue une cellule)

LES ATTRIBUTS de TABLE, TR et TD		
BALISE	ATTRIBUT	EFFET
<i>TABLE</i>		
	align = "left", "center", "right"	aligne le tableau dans la page
	width = "x %" ou width="nbe pixels"	largeur du tableau
	border="n"	taille de la bordure en pixel
	bordercolor="#xxxxxx"	couleur de la bordure
	bgcolor="#xxxxxx"	couleur de fond
	cellpadding	espace entre bordure des cellule et contenu
	cellspacing	largeur de la grille intérieure
	background = "nom_image.ext"	image de fond
<i>TR</i>		
	align = "left" "center" "right"	alignement horizontal sur la rangée

	valign = "top", "middle", "bottom", "baseline"	aligne le contenu verticalement
	bgcolor="#xxxxxx"	couleur de fond
	bordercolor="#xxxxxx"	couleur de bordure
	background = "nom_image.ext"	image de fond
<i>TD</i>		
	width = "n"	largeur d'une cellule
	height = "n"	hauteur d'une cellule
	bgcolor="#xxxxxx"	couleur de fond
	bordercolor="#xxxxxx"	couleur de bordure
	background = "nom_image.ext"	image de fond
	colspan = "n"	fusion de "n" cellules horizontales"
	rowspan = "n"	fusion de "n" cellules verticales"

LES IMAGES

Que seraient les sites WEB aujourd'hui sans l'affichage d'images, de gif animés ? c'est certainement à un univers multimédia que pensaient les concepteurs du web qui en 1989 mirent en place les outils qui nous permettent aujourd'hui de surfer dans des environnement de plus en plus visuels, voire sonores !

Les navigateurs acceptent facilement des images à différent formats (gif, jpeg, png). Le principe du web basé sur le modèle client/serveur implique le téléchargement des fichiers qui composent les pages visitées sur la machine de l'internaute. Si les pages de sources sont forcément de tailles très réduites, dès que l'on affiche des images sur notre site il faut penser au temps de chargement de fichiers forcément plus volumineux et donc de temps d'affichage pouvant être plus ou moins long en fonction de la taille de nos fichier mais aussi de la vitesse de connexion de l'internaute que nous ne maîtrisons pas. Certes aujourd'hui l'adsl et le câble permettent des connexions très rapides mais il y a encore beaucoup d'internautes qui utilisent des modems RTC de faible débit.

Pensez-y en écrivant votre code.

Une image se place dans la page web avec la balise ``

- img représente la balise image
- src (pour source) représente le chemin vers votre fichier image

NB: Il est fortement recommandé de placer ses images dans un répertoire spécifique. Il en va de même d'ailleurs de toutes les parties de votre site afin de pouvoir s'y retrouver facilement en cas de modification de vos sources, de zippage de vos répertoires d'images ou autre.(aparté). **Rappel :** on descend dans un répertoire en indiquant son nom, on remonte avec ../

Les attributs de `<img >` sont assez nombreux et concernent aussi bien leur alignement par rapport au texte, leur dimension de même que les textes de remplacement.

Spécifier la taille de l'image : *width* pour largeur et *height* pour hauteur. permet d'afficher une image avec une taille personnalisée. cela permet aussi un affichage plus rapide en évitant au navigateur de calculer la taille de l'image

Spécifier un texte de remplacement : Dans le cas où l'image n'est pas accessible ou pour les navigateurs "texte" l'attribut *alt="texte"* permet d'afficher un texte de remplacement.

Organiser le texte autour de l'image

- Espace entre l'image et le texte : avec les attributs *vspace = "n"* et *hspace = "n"* vous forcez un espace de n pixels respectivement verticalement et horizontalement autour de l'image.
- Alignement par rapport au texte : par défaut, le texte est aligné sur le bas de l'image. Il y a 9 attributs pour spécifier l'alignement. les principaux sont : *left*, *right*, *middle*, *absbottom* (en bas), *absmiddle*(au milieu); *bottom* (en haut)

Entourer une image : Vous pouvez afficher une bordure à l'image avec l'attribut *border = "n"* n pixels. Elle sera automatique si l'image est encapsulée dans une balise de lien hypertexte. La valeur « 0 » de l'attribut *border* vous en affranchira. (c'est pas beau une grosse bordure bleue sur vos images artistiquement ciselées !

LES FORMULAIRES HTML

C'est avec les formulaires que l'on entrevoit enfin une possibilité d'interactivité entre l'internaute et les créateurs ou gestionnaires d'un site Web. Avec ces procédés nous pouvons en effet demander à l'internaute de saisir du texte, de cocher des cases, de choisir un item dans une liste de choix.

Les données ainsi saisies sont en général traitées par un script s'exécutant coté serveur et non plus sur la machine de l'utilisateur.

On saisie un bloc de formulaire entre les balises <form>... </form>

Quelques attributs de cette balise sont essentiels

- ACTION ="url" définit l'adresse sur laquelle les données seront postées
- METHOD ="POST ou GET" définit la méthode utilisée pour poster les données
 - post** Méthode permettant d'envoyer un grand nombre de données, plus sûre que "get" les données sont stockées dans une variable d'environnement
 - get** Méthode par défaut, plutôt utilisée pour données limitées, peu sûre les données sont visibles dans l'url
- NAME sera important en DHTML pour nommer formulaire parmi plusieurs dans la page

A l'intérieur des balises de formulaire on trouvera deux types de balises :

I. Saisies et choix de l'internaute

- < input type="text" > texte de saisie
- <input type="radio" > bouton radio à cocher
- <input type="checkbox" > case à cocher
- <input type="password" > champ de saisie cachée

La balise input accepte plusieurs attributs :

- name ="..." *nom du champs*
 - align ="..." *alignement "right" "center" "left"*
 - maxlength ="..." *longueur maximale de saisie en caractères*
 - size ="..." *longueur d'affichage du champ dans la page*
 - type ="..." *énumérés plus haut 'text' 'password' 'radio' 'checkbox' 'button' 'submit' ...*
 - value ="..." *valeur par défaut du champ*
- < select > *liste des options* < /select > *liste déroulante*
- <option value="" > </option > Chaque balise option ajoute un item dans la liste
 - la balise option admet comme attribut : *class, id, selected (valeur par défaut), value (valeur associée à l'option)*
- <textarea > texte de saisie mais sur plusieurs lignes

II. Action globale

Bouton submit

< type="submit" value="Envoyer" > active l'action définie dans la balise formulaire, attribut action

Bouton Reset

< type="reset" value="Recommencer " > remet à blanc ou à défaut les valeurs saisies dans le formulaire

Tableau récapitulatif des balises et attributs de formulaires

BALISE	ATTRIBUTS	DESCRIPTION
FORM (balise fermante)		
	action	adresse url ou poster le formulaire
	enctype	encodage des données
	name	Nom de l'objet formulaire format des données à soumettre
	method GET ou POST	Méthode utilisée pour poster le formulaire
INPUT		
	type	button, checkbox, file, hidden, image, password, radio, reset, submit, text
	name	nom du champ de saisie
	src	source de l'image pour type="image"
	class	nom de la class
	maxlength	longueur maximale en caractère du champ de saisie
	size	largeur du champs en nombre de caractères
	value	valeur par défaut
SELECT (balise fermante)		
	name	Nom de l'objet
	option (balise fermante)	Chacun des items. attributs selected(défaut) et value (valeur associée à l'option)
	align = "left", "right", "middle", "absbottom", "absmiddle", "baseline", "bottom", "texttop", "top"	Alignement dans la liste
	multiple	Permet un choix multiple dans la liste
TEXTAREA (balise fermante)		
	name	Nom de l'objet
	cols	Nombre de colonnes du texte
	rows	Nombre de lignes
	wrap	gestion des retours à la ligne (virtual, physical, off)

Voici l'aspect d'un Formulaire de base en HTML

FORMULAIRE SATISFACTION

Entrez votre identifiant (max 12caractères):

Vous avez entre :

Etes-vous satisfait de ce tutorial ?
☐ Très satisfait
☐ Satisfait
☐ Moyennement satisfait
☐ Peu satisfait

☐ Voulez vous recevoir des informations futures sur ce site ?

Ce petit formulaire qui termine la partie consacré aux bases du HTML ne fait rien d'autre qu'afficher une boîte d'alerte lorsqu'on clique sur le bouton submit. Pour récupérer les données saisies ou choisies par l'internaute il nous faudra utiliser un script serveur (php ou asp par exemple) afin de stocker les données dans une base de données.

LES FEUILLES DE STYLE

Les feuilles de style en cascade (**cascading style sheet ou css**) ne datent pas d'hier puisqu'elle ont été élaborées et rendues publiques en 1996. Ne faisant pas partie du standard html, elle sont en fait une recommandation du W3C. ce fait imposera, comme souvent en matière de langage web, qu'il faudra s'assurer que les différents navigateurs réagissent correctement à vos scripts, des différences pouvant avoir lieu entre les ténors du web IEx et Netscape et aussi les navigateurs plus confidentiels. Ceci étant dit les CSS sont un atout majeur pour la réalisation de site web ayant une interface élaborée. En effet si le html structure les pages on peut dire que c'est plutôt à coup de hache que d'une manière ciselée. HTML ne permet pas de mise en page très précise, certaines de ses balises deviennent obsolètes avec les années, le texte des pages est noyé dans les propriétés des balises FONT, TABLE, etc..

BREF : tout ça manque de clarté dans le code et de précision dans la mise en page

Les feuilles de style, c'est l'outil qui vient mettre de l'ordre dans tout ça

Elles :

- Facilitent l'homogénéité de vos sites
- Améliore la lisibilité du code
- Permettent un positionnement très précis des éléments de la page (au pixel près)
- permettent d'intervenir sur des modifications globale au site en quelques lignes
- Facilite l'utilisation du DHTML et du DOM

Elle permettent aussi d'adapter un site à différents medias (impression, écran, rétroprojecteurs)

Comment ça marche ? : Le css regroupe un ensemble complet de propriétés liées au élément de la page web

- couleur, tailles style et familles de polices
- couleurs d'arrière plan,
- propriété de positionnement dans la page y compris superposition
- sélecteur de balises ou pseudo classe qui nommeront un ensemble de propriétés que vous définirez

L'ensemble de ces paramètre peut être inséré de plusieurs manières

- Dans l'en-tête de la page web entre les balise `<style type="text/css"> vos déclarations de styles </style >`
- Dans la page même `<p style="font-size:12px;text-align:center"> ..... </p>`
- Dans un fichier séparé en général au format .css ou vos définition de style sont regroupées

Il va sans dire que cette dernière manière dont nous allons parler plus en détail est la plus intéressante

Le fichier séparé est appelé par un lien et donc les mêmes définitions de style peuvent être

appliquées et modifiée dans tous le site en ne les déclarant qu'une fois
La définition d'une feuille de style séparée est simple. il suffit de créer un fichier
nom-de_fichier.css

D'y déclarer ses styles, css introduit la notion d'objet avec l'attribut "ID" ou "CLASS"
d'appeler la feuille de style dans l'en tête de la page avec la commande :

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

il faudra ensuite appeler le style dans la balise qui doit être éditée :

<p id="#non-du-style"> le contenu du paragraphe **>**

EXEMPLE RAPIDE SOUS FORME DE COPIE D'ECRAN :

chaque page de mon site doit contenir un paragraphe :

positionné à 60 pixels de la marge gauche

avec une taille de police de 22 px

Une police de la famille verdana

Une couleur de police en rouge

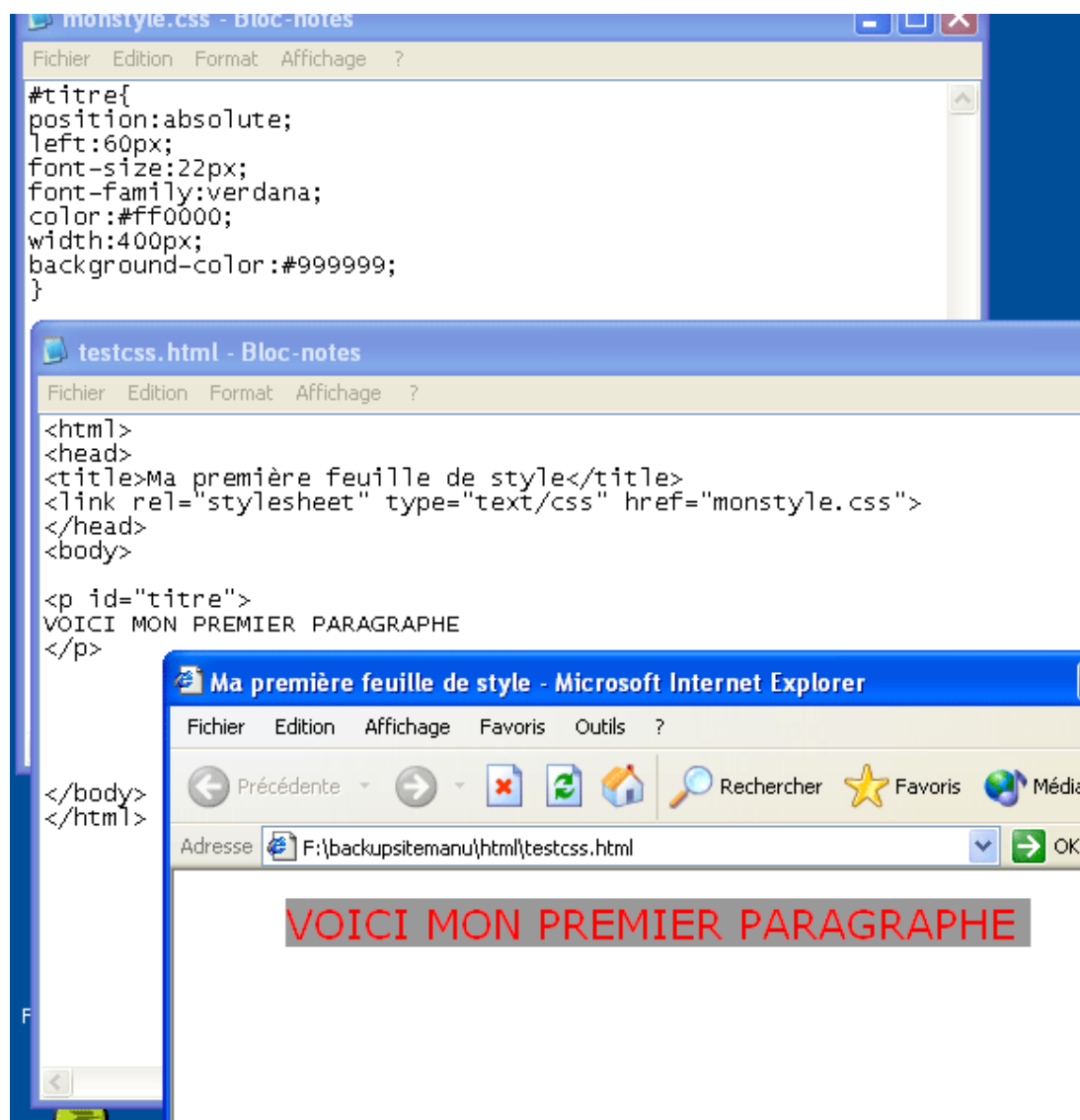
je souhaite que ce paragraphe ait une largeur de 200px

qu'il soit associé à un fond gris.

1/ Je définis un style dans un fichier monstyle.css

2/ j'appelle le fichier css dans l'en tête de ma page

3/ J'appelle l'id du style défini dans la balise <p



A noter que toutes les pages de mon site devant reprendre cette mise en page pourront être appelée de la même manière :

- appel de la feuille de style dans le head de la page
- balise p avec l'id du style
- bien évidemment le contenu du paragraphe peut être différent pour chaque page mais il aura la même mise en page

QUELQUES CONSTANTES A CSS

On appelle ces feuilles de style "en cascade" car elles comprennent une notion hiérarchique : les styles définis dans la page sont prioritaires sur ceux définis dans l'en-tête, qui sont eux-mêmes prioritaires à ceux définis dans un fichier séparé. Les propriétés et leurs valeurs sont définies à l'intérieur d'accolades, ne pas oublier le point-virgule final. Mozilla ou Netscape ne vous le pardonneront pas !

Vous l'avez peut-être deviné, si en HTML les "calques" du document se succèdent les uns au-dessous des autres, les feuilles de styles positionnent les éléments en fonction des propriétés d'alignement, type de position (relative ou absolue), left ou right, top.

Les valeurs des propriétés css sont modifiable dynamiquement par le biais de javascript (intéressant non ?)

Attention ! Ne rien mettre d'autre dans vos fichier css que des déclarations de style
Ecrivant ces pages de mémoire je ne vois rien d'autre pour le moment mais si des choses importantes me reviennent en mémoire (pas toujours très vive je dois dire), promis je reviendrai sur cette liste.
Poursuivons avec les principales propriétés css

LES PRINCIPALES PROPRIETES CSS

Présentation des principales propriétés

les unités

Les Unités en css		
Types	Unités	description
données absolues	cm	Centimètres
	in	Inch pouce=2.54cm
	mm	millimètres
	pc	pica 12pt ou 1/6 de pouce
	pt	point 1/72 de pouce
données relatives	em	taille de l'attribut font-size
	ex	la moitié d'un em environ
	px	pixel
	%	pourcentage

unités de couleurs	
Unités	Description
#rrvvbb	méthode classique pour afficher les couleurs avec des valeurs hexadécimales
rgb(x,x,x)	red green blue avec des valeurs décimales de 0 à 255
rgb(x%,x%,x%)	si ça vous amuse avec des pourcentage de la couleur maxi (255)
nom_couleur	les quelques dizaines de couleur format texte reconnue. pas facile à mémorisé

Les propriétés liées aux fonds de balises

fonds de balises	
propriétés	Description
background-color: <i>valeur</i>	couleur du fond ou transparent

background-image:url(chemin)	insère l'image spécifiée en fond de balise
background-repeat	spécifie le mode de répétition de l'image : no-repeat l'image n'est affichée qu'une fois : repeat-x l'image est répétée dans le sens horizontal x fois : repeat-y l'image est répétée dans le sens vertical x fois : repeat l'image est répétée dans les deux sens
border border-color border-width border-style	spécifie une bordure autour du calque border-style : solid;dashed;dotted;double. 3d : groove,ridge,inset, outset affiche sur le périmètre : top;bottom;left;right la combinaison permet des effets intéressants

Les propriétés liées au texte

propriétés du texte	
propriétés	Description
color	couleur du texte
text-align	alignement du texte dans la balise valeurs ; left;right;justify,center
text-decoration	décoration: none;underline,overline;line-through. plusieurs attributs possibles

Les propriétés liées aux polices

propriétés des polices	
propriétés	
font	générique : font-style; font-color; font-weight ect..
font-family	comme pour html éviter les police exotiques qui ne serait rarement sur le système de l'internaute
font-size	taille de la police voir unités de mesures
font-style	italic;normal;oblique
font-weight	graisse de la police. normal=400; bold=700 valeurs admises jusqu'à 900

Les propriétés liées aux marges

propriétés des marges internes	
propriétés	description
padding	left;top;right;bottom : générique détermine un espace entre la bordure interne et le contenu
margin	left;top;right;bottom : définit la taille d'une marge

positionnement du contenu des balises

Propriétés de positionnement	
propriétés	description
position	absolute, relative. en absolu permet le chevauchement de plusieurs balises
z-index	Détermine le niveau d'empilement du calque. plus la valeur est grande plus le calque sera au dessus des autres cet attribut est intéressant si l'on souhaite faire se chevaucher des calques. attention positionnement absolu de rigueur
left, right	Détermine l'éloignement de la balise par rapport à la balise parent
top	Détermine l'éloignement vertical de la balise par rapport à la balise parent
vertical-align	alignement vertical de la balise : bottom,middle;super;sub
overflow	indique le comportement du contenu en rapport à la taille de la balise auto :contenu clippé, le navigateur affiche une barre de défilement hidden : contenu typé mais pas de barre de scrolling (!bof) visible : le contenu dépasse le cadre de la balise

Fin de ce tutorial. Je le compléterai dans le futur (proche) mais il y a là l'essentiel de ce que l'on doit savoir en html pour faire des pages de bonne tenue.