



Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Direction Générale des Etudes Technologiques



Institut Supérieur des Etudes Technologiques de Djerba

Programmation web 1

Support de cours



Elaboré par : Mr. BOUREGBA Aymen

Version : 2010

Public cible : Classes de la 1^{ère} année

Licence : Technologies de l'Informatique

Département : Technologies de l'informatique

Année universitaire : 2010-2011

AVANT PROPOS

De nos jours, l'Internet et le web sont devenus des expressions courantes et fréquemment utilisées. Ce support est consacré d'une part à l'étude des services et les fichiers supportés par Internet ainsi que les notions de base nécessaires pour la création de pages web, et d'autre part à la présentation des outils nécessaires permettant d'obtenir des IHM répondant aux besoins tel que le langage HTML avec ces différents type de composants : tableaux, formulaires ; les feuilles de style, ainsi que le langage supportant le standard d'échange XML : le XHTML. Il est destiné aux étudiants des établissements (ISETs, 1^{ère} année licence technologies de l'informatique).

La pratique des différentes notions abordées dans ce support, sera assurée par le module « Atelier web 1 » par le biais d'un ensemble de TPs et un mini-projet.

Le contenu, ainsi que l'enseignement pédagogique de ce cours restent toujours aptes à évoluer en fonction des critiques et remarques constructives des lecteurs. Aussi, avec la participation des étudiants, une évaluation semestrielle de l'enseignement de ce module est effectuée par le biais de fiches à rendre commentées librement.

J'adresse d'avance mes remerciements aux lecteurs qui voudront bien faire part de leurs critiques et de leurs remarques constructives.

FICHE MATIERE

Pré requis

- Connaissances générales sur l'Internet et le Web

Objectifs généraux

Ce cours vise à développer chez l'apprenant les compétences adéquates pour planifier et concevoir un site web en utilisant les logiciels appropriés.

Objectifs spécifiques

L'étudiant doit être capable de :

- Maîtriser les services offerts par Internet.
- Savoir écrire une page web en utilisant le langage HTML/CSS

Public cible

Etudiants de la première année licence (L1) : Technologies de l'Informatique.

Volume horaire

0.75h pour chaque groupe (de manière hebdomadaire)
Soit en total : 11.25h.

Sommaire

Chapitre I : Généralités et notions de base.....	7
Introduction.....	7
I Historique.....	7
I.1 Les années 60 : Contexte de la guerre froide	7
I.2 Les années 70/80 : Arpanet et le protocole TCP/IP.....	8
I.3 Les années 80/90/2000 : d'Internet au WWW, une montée exponentielle.....	8
II Les services d'Internet.....	9
II.1 Messagerie électronique	9
II.2 Forum de discussion.....	9
II.3 Transfert de fichiers	10
II.4 Telnet et SSH	10
II.5 Cyber-bavardage (chat).....	10
II.6 Journal personnel (Blog).....	10
II.7 Moteur de recherche et annuaire Web.....	11
III Principaux formats de fichiers.....	11
III.1 Les documents textes	12
III.2 Les images.....	12
III.3 Les fichiers audio	12
III.4 Les fichiers vidéo	13
IV Création d'un site Web	13
IV.1 Définition.....	13
IV.2 Objectifs et cible	14
IV.3 Les technologies utilisées	14
IV.4 Présentation du site	15
V Le référencement et la gestion	16
V.1 Le référencement.....	16
V.2 La gestion.....	16
Conclusion	17
Chapitre II : Les bases du HTML	19
Introduction.....	19
I Qu'est-ce qu'une balise HTML ?	19
I.1 Les balises simples	19
I.2 Les balises doubles	20
I.3 Structure d'un document HTML.....	20
I.4 Le DOCTYPE.....	20
I.5 Les commentaires	21
II Le corps du document – body.....	21
II.1 Mise en forme.....	21
II.2 Les images.....	23
II.3 Les liens hypertextes	23
II.4 Les listes.....	25
Conclusion	29
Chapitre III Les notions avancées du HTML	31
Introduction.....	31
I Les tableaux	31

I.1 À quoi servent-ils ?.....	31
I.2 Du tableau simple au tableau complexe	31
I.3 L'utilité des tableaux aujourd'hui.....	35
II Les formulaires	35
II.1 À quoi servent-ils ?	35
II.2 La balise form.....	35
II.3 Les champs texte mono lignes	36
II.4 Les boutons radio	38
II.5 Les cases à cocher	38
II.6 Les boutons	39
II.7 Les listes déroulantes	40
III Le XHTML	41
III.1 Définition	41
III.2 Validation W3C.....	41
III.3 Syntaxe.....	42
Conclusion	44
Chapitre IV : Les feuilles de style.....	46
Introduction.....	46
I Utilité et avantages	46
II Définition	47
II.1 CSS Globale	48
II.2 CSS Intra-lignes	48
II.3 CSS importée.....	49
III Les classes et les ID	49
III.1 Notion de classes.....	49
III.2 Notion des ID	50
IV Les pseudo-éléments et les pseudo-classes.....	51
IV.1 Les pseudo-classes.....	51
IV.2 Les pseudo-éléments.....	53
V Les balises structurales.....	54
VI Le flux.....	54
VII Le positionnement.....	55
VII.1 Généralités.....	55
VII.2 Attributs.....	57
Conclusion	59
Bibliographie	60
Webographie	60

CHAPITRE

I

Généralités et notions de base

Pré-requis

- ☞ Aucun.

Objectifs généraux

- ☞ Introduire la notion d'Internet ainsi que ses services.
- ☞ Présenter les étapes et les techniques de création d'un site web.

Plan :

- Introduction.
- I. Historique.
- II. Les services d'Internet.
- III. Principaux formats de fichiers.
- IV. Création d'un site web
- V. Le référencement et la gestion
- Conclusion.

Chapitre I : Généralités et notions de base

Introduction

Surfer, ou naviguer, sur Internet sont devenus des expressions courantes et fréquemment utilisées. Elles montrent bien l'image de cet océan d'informations qui s'étend aujourd'hui à la grandeur de la planète et continue de croître et de se diversifier chaque jour [1].

L'Internet marque un virage technologique et culturel important en diffusant aussi largement l'information électronique et en donnant accès à des grandes quantités d'information pour un grand nombre de personnes à travers le monde.

Il s'agit d'un réseau international d'ordinateurs qui communiquent entre eux dans une toile dense de relations et de connexions. Les différents ordinateurs branchés au réseau Internet peuvent communiquer ensemble de façon transparente pour l'utilisateur indépendamment des types d'ordinateurs utilisés (Mac, PC, Unix ou autres), mais en utilisant cependant les logiciels appropriés et surtout les protocoles standard.

Le mot Internet (abréviation du mot anglais internetwork) désigne un réseau informatique constitué par l'interconnexion de plusieurs réseaux. Pour cette raison, on appelle aussi Internet « le réseau des réseaux ».

I Historique

Internet est l'infrastructure de communication à l'échelle planétaire issue de l'interconnexion de réseaux informatiques publics et privés, il a été créé dans les années 70. À l'origine, l'Internet relie les sites informatiques militaires des États-Unis.

I.1 Les années 60 : Contexte de la guerre froide

- En 1957, l'ARPA (Advanced Research Project Agency) a été créée aux USA pour piloter un certain nombre de projets dans le but est d'assurer la supériorité scientifique et technique sur les Russes. Cette organisation regroupait certains des plus brillants cerveaux américains.

- En 1967, Lawrence G. Roberts, récemment arrivé à la tête du projet de réseau informatique à l'ARPA, publie ces « Plans pour le réseau ARPANET ».

- Présentation d'une conception de réseau à commutation de paquets par Donald Davies et Robert Scantlebury du NPL (National Physical Laboratory Royaume Uni).

Également un article de Paul Barran de la RAND Corporation (militaires) au sujet de l'utilisation d'un réseau à commutation de paquet pour transmission sécurisée de la voix.

I.2 Les années 70/80 : Arpanet et le protocole TCP/IP

- En 1969, le réseau Arpanet est opérationnel reliant initialement 4 instituts universitaires. Cette liaison a permis aux quatre institutions de transférer des données et d'effectuer à distance des calculs longs sur plusieurs ordinateurs.

- Tout au long des années 70, les laboratoires de recherche sont progressivement reliés au réseau Arpanet.

- En 1970, le protocole NCP (Network Control Protocol) a été utilisé sur Arpanet afin de pouvoir mettre en liaison des machines hétérogènes (Unix, IBM,...).

- En 1983, le NCP est définitivement abandonné au profit du TCP/IP (Transfer Control Protocol / Internet Protocol) que nous utilisons encore aujourd'hui et qui constitue le protocole de base d'Internet. TCP est chargé de segmenter le message en paquets et de réarranger ces derniers à la réception, tandis qu'IP est chargé d'assurer l'acheminement des paquets d'un ordinateur vers un autre jusqu'à la destination.

I.3 Les années 80/90/2000 : d'Internet au WWW, une montée exponentielle

- En 1977, Internet devient une réalité car le TCP/IP est effectivement utilisé pour relier divers réseaux à Arpanet. Le nombre d'ordinateurs connectés dépasse alors la centaine et depuis, ce nombre ne cesse d'augmenter d'année en année.

- L'origine du Web remonte à mars 1989 lorsque Tim Berners-Lee, un informaticien au CERN (Conseil Européen pour la Recherche Nucléaire), propose de créer sur le site Internet du CERN un ensemble de documents reliés les uns aux autres par des liens hypertextes, afin de faciliter la recherche d'informations pour les physiciens.

- Le début des années 1990 marque la naissance d'Internet tel que nous le connaissons aujourd'hui : le Web, un ensemble de pages en HTML (HyperText Markup Language) mélangeant du texte, des liens et des images, accessibles à l'aide d'une URL (Uniform Resource Locator) en utilisant le protocole HTTP (HyperText Transfert Protocol).

- En 1991, Tim Berners-Lee du CERN à Genève met au point l'interface d'Internet appelée World Wide Web, qui permet d'ouvrir le réseau au grand public en simplifiant les procédures de consultation des sites.

- En 1991, 300 000 ordinateurs sont connectés, ce nombre atteint 1 000 000 en 1992.
- En 1992, le premier lien (on parle de lien hypertexte) vers le site Web du CERN est créé sur le serveur de Fermilab aux États-Unis : la toile mondiale commence à se tisser. Les années 1990 ont vu le Net continuer à grossir à une vitesse exponentielle sous l'impulsion du Web.
- En 1993, Apparition du premier navigateur web (browser) par Netscape, supportant le texte et les images. Cette même année, la NSF (National Science Foundation) crée une compagnie pour enregistrer les noms de domaine.
- En 1993, on comptait 600 sites, ce chiffre dépasse 15 000 en 1995. Le WWW devient le service le plus important sur Internet.
- En 2008, on compte 1,5 milliard d'internautes dans le monde, 1.3 milliard d'utilisateurs d'email, 210 milliards d'emails envoyés par jour, 186,7 millions de site web, 133 millions de blogs.
- Le chiffre d'affaires du commerce en ligne (e-commerce), qui a dépassé la barre des 20 milliards d'euros en 2008, devrait atteindre 30 milliards en 2010.

II Les services d'Internet

Internet offre plusieurs services à ses usagers. Pour y avoir accès, l'utilisateur (l'internaute) doit donc disposer des logiciels clients adéquats, navigateur web, client de messagerie électronique, boîte pour chat, etc.

II.1 Messagerie électronique

La messagerie électronique (e-mail ou encore mail), c'est utiliser Internet comme on utilise les lettres dans la poste. Il est possible de déposer un message dans la boîte aux lettres de son correspondant, qu'il soit ou non devant une machine. Ce dernier sera capable, à sa prochaine connexion, de consulter sa boîte aux lettres pour lire ou envoyer des messages à ses correspondants.

II.2 Forum de discussion

La principale limitation du courrier électronique est qu'il faut choisir un destinataire pour pouvoir envoyer un mail. La solution est dans les forums de discussion.

Il s'agit d'un espace de discussion sur Internet fonctionnant, de manière asynchrone, comme une boîte aux lettres publique, dans laquelle chacun est libre de consulter les messages et d'y

répondre. Un forum ou newsgroup (groupe de discussion) s'établit généralement autour d'un sujet donné. Dans certains cas, une personne appelée modérateur se charge d'animer les débats et d'éliminer les messages non conformes à la charte du forum.

II.3 Transfert de fichiers

Dans Internet, il est possible de transférer des fichiers d'un ordinateur à un autre grâce au protocole FTP (File Transfer Protocol). En pratique, ce protocole permet à un usager d'accéder aux disques durs montés sur un ordinateur distant.

Le service d'échange de fichiers permet de déposer des fichiers sur une machine distante, mais aussi, et c'est le plus fréquent de télécharger des fichiers sur sa machine. La distribution de logiciels gratuits, la diffusion d'images, de sons, de notes de cours pour les étudiants ou d'articles scientifiques sont parmi les utilisations les plus courantes de ce service.

II.4 Telnet et SSH

Ces services permettent à un internaute de se connecter et donc d'utiliser à distance une machine comme s'il se trouvait face à elle. Ce contrôle à distance permet d'exécuter des commandes, de démarrer ou arrêter un programme, de consulter l'état de la machine ...etc.

Telnet est plus ancien, et commence à être abandonné au profit de SSH pour des raisons de sécurité : avec Telnet toutes les informations transmises transitent en clair sur le réseau, y compris les mots de passe, ce qui permet à un éventuel pirate à l'écoute du réseau d'intercepter des données sensibles. À l'inverse, pour SSH toutes les données transmises sont cryptées et donc illisibles par toute autre personne que le destinataire.

II.5 Cyber-bavardage (chat)

Le cyber-bavardage (chat en anglais) désigne les activités permettant à des utilisateurs d'Internet d'échanger, de manière interactive et en temps réel, avec d'autres utilisateurs. Cette forme de conversation s'effectue soit textuellement (chat), soit avec images vidéos (webcam, vidéoconférence) ou avec du son (audioconférence).

II.6 Journal personnel (Blog)

Le Blog (ou WeBlog) est un journal personnel en ligne qui permet à un internaute de publier régulièrement des informations ou de commenter l'actualité sur un sujet.

Un journal de bord personnel sur internet permettant de publier facilement des actualités (les Articles ou Notes ou Billets ou Postes dans la langue des blogueurs) sur un sujet, de les

illustrer de façon multimédia (dessins, photos, vidéos, sons...) et de partager ses idées en recueillant des commentaires sur ses articles.

II.7 Moteur de recherche et annuaire Web

Un moteur de recherche est un logiciel permettant de retrouver des ressources (pages web, forums, images, vidéos, fichiers... etc.) associées à des mots quelconques (mots clés). Certains sites web offrent un moteur de recherche comme principale fonctionnalité.

Il s'agit d'un outil de recherche sur le Web constitué de Robots, (appelés encore bots, spiders, crawlers ou agents) qui parcourent régulièrement les sites de façon automatique (sans intervention humaine, ce qui les distingue des annuaires Web) pour découvrir de nouvelles adresses (URL). Ils suivent les liens hypertextes (qui relient les pages les unes aux autres) rencontrés sur chaque page atteinte.

Selon une étude réalisée en août 2007 :

- Google : environ 60% des 61 milliards de recherches sur internet,
- Yahoo : 8,5 milliards de recherches, soit 14% du total,
- Baidu (le Google chinois) : 3,3 milliards de requêtes, soit 5,4% du total,
- Bing (remplaçant de Live search de Microsoft) : 2,1 milliards de recherches, 3,4%.

Un *annuaire Web* (appelé également *répertoire Web*, *annuaire Internet* ou *répertoire Internet*) est un site Web proposant une liste classée de sites Web. Le classement se fait dans une arborescence de catégories, censée à couvrir tout ou une partie des centres d'intérêt des visiteurs.

III Principaux formats de fichiers

L'utilisation, sur Internet, des composantes multimédia et interactives ou d'extensions logicielles nécessite plusieurs formats et standards de fichiers. Bien que plusieurs de ces composantes s'intègrent de façon transparente aux pages Web, les utilisateurs d'Internet rencontrent souvent des formats de fichiers qui peuvent nécessiter l'utilisation de logiciels particuliers.

Les fichiers d'ordinateurs sont de deux types : ASCII ou binaires. Les premiers sont constitués de caractères qui respectent les codes ASCII (American Standard Code for Information Interchange), donc lisibles à l'écran facilement. Les fichiers binaires contiennent des informations

codées directement en binaire (des 0 et des 1). Pour afficher ou exécuter un fichier binaire, il faut utiliser un logiciel compatible avec celui qui l'a créé.

III.1 Les documents textes

- HTML (HyperText Markup Language) [Type ASCII]. Utilisé pour la création des pages Web.
- PDF (Portable Document Format) [Type binaire]. Format du logiciel Acrobat d'Adobe.
- Très utilisé pour la diffusion de documents sur Internet.
- XML (eXtensible Markup Language) [Type ASCII]. Il sert essentiellement à stocker/transférer des données structurées en champs arborescents.
- TXT [Type ASCII]. Par convention, on donne cette extension aux fichiers de texte simple non formaté.

III.2 Les images

- GIF (Graphics Interchange Format) [Type binaire]. Format assez compacts mais limité à un maximum de 256 couleurs. Utilisé pour produire de petites images peu texturées, des icônes, des boutons dans les pages Web, etc. Permet la transparence et les images animées.
- JPEG, JPG (Joint Photography Experts Group) [Type binaire]. Très fréquent sur Internet, produit des fichiers très compacts grâce à son algorithme de compression efficace mais qui provoque des pertes d'information. Utilisé pour produire des photos et images texturées dans les pages Web.
- PNG (Portable Network Graphics) [Type binaire]. Appelé à remplacer les formats GIF et JPEG.
- TIFF, TIF (Tagged Image File Format) [Type binaire]. Utilisé pour l'archivage d'images de qualité. Utilise un algorithme de compression sans perte. Fichiers de grandes dimensions.

III.3 Les fichiers audio

- AIFF, AIF (Audio Interchange File Format) [Type binaire]. Relativement répandu dans Internet.

- AU (Unix Audio sound file format) [Type binaire]. Développé pour Unix. Très répandu dans Internet.
- MID (Music Instrument Digital Interface) [Type binaire]. Norme pour transmettre des informations musicales entre instruments électroniques et ordinateurs.
- MP3 (Moving Pictures Expert Group) [Type binaire]. Format public pour la numérisation audio en haute-fidélité (bonne qualité).
- RA (RealAudio) [Type binaire]. Format propriétaire très ancien, il permet de diffuser de la musique sur internet en utilisant la technique du streaming.
- WMA, WAV (Waveform Audio File Format) [Type binaire]. Format utilisé par Windows pour stocker des séquences sonores non compressées (WAV) ou multimédia assez bien compressées (WMA).

III.4 Les fichiers vidéo

- AVI (Audio Video Interleave movie) [Type binaire]. Format vidéo développé par Microsoft.
- MOV, QT (QuickTime Movie) [Type binaire]. Développé par Apple. Bonne qualité d'image.
- MPEG, MPG (Moving Pictures Expert Group) [Type binaire]. Un standard de compression vidéo et audio très répandu dans Internet.
- DivX [Type binaire], capable de compresser des fichiers vidéo volumineux en fichiers plus légers.

IV Création d'un site Web

IV.1 Définition

Un site Internet ou Intranet d'informations reste avant tout une vitrine, un support. C'est pour cela qu'il est considéré comme un important moyen de communication et d'identification. La mise en place d'un site doit être étudiée de façon approfondie et ne peut être abordée comme la rédaction d'un document classique. Il s'agit de rendre cohérent une masse d'information et de la mettre à la disposition d'un panel hétérogène d'utilisateurs plus. C'est pourquoi un intérêt tout particulier doit lui être porté, aussi bien au niveau du contenu que de la présentation [3].

Un site Internet se décompose " physiquement " en plusieurs éléments :

- Un nom de domaine (une URL) accessible notamment par les moteurs de recherche mais aussi par des sites partenaires et à travers différentes publications papier. C'est pourquoi, l'adresse du type `http://www.notresite.com` doit être claire, courte et explicite.
- Un ensemble de pages statiques et/ou dynamiques reliées entre elles par des liens. Ces pages seront développées dans une technologie dépendante de l'objectif et du contenu du site.
- Les pages seront ensuite hébergées chez un prestataire ou en interne en fonction notamment de la technologie utilisée, du volume du site, du trafic escompté, du budget alloué.

IV.2 Objectifs et cible

Il est crucial de s'interroger sur la finalité du site car il doit répondre à un besoin et non pas à une envie.

- Définir l'objectif du site (complément d'un outil spécifique, sources d'informations, portail, commerce en ligne, base de données...).
- Quel est le public visé ? Est-ce un groupe d'amis, des personnes partageant les mêmes idées ou de futurs clients ? Il faut donc ensuite cadrer le style : les informations doivent-elles être pointues ou vulgarisées ?
- Que sera l'environnement cible des visiteurs: type de configuration (PC/Mac), navigateur, outils disponibles (Pack office, décompresseur type Winzip, messagerie), puissance des machines, performance et durée des connexions (l'accès à Internet se fait il par modems ou via une ligne à haut débit ?).

IV.3 Les technologies utilisées

- Une bonne connaissance du langage L'HTML s'avère indispensable pour créer des pages Web. A savoir qu'il existe des éditeurs (WYSIWYG : What you see is what you get), comme FrontPage, DreamWeaver, HTMLPad, Nvu..., qui nous assiste à la création rapide des pages Web.
 - A savoir qu'il existe des systèmes de gestion de contenu « CMS » : est une famille de logiciels destinés à la conception et à la mise à jour dynamique de site web ou d'application multimédia.

Compte rendu : menez une recherche sur les différents CMS du marché ainsi que leurs fonctionnalités.

- JavaScript / DHTML (animation simple de texte, interactivité réduite via des formulaires).
- Flash (animation graphique avancée) à réserver aux sites en Intranet ou dédiés au graphisme.
- Pages actives : il existe plusieurs technologies pour la création de pages actives (c'est à dire capables de générer un contenu dynamique en fonction d'informations données par le visiteur ou l'administrateur du site) : ASP, JSP, PHP et les CGI avec PERL associées à plusieurs SGBD (MYSQL, SQL Server, ACCESS, Oracle...).

IV.4 Présentation du site

La charte graphique et l'ergonomie d'un site sont deux points à ne pas négliger.

- Obtenir la charte graphique existante et l'adapter pour une diffusion en ligne.
- Créer et structurer une charte suivant les besoins énumérés.
- Respecter les contraintes des services et les critères de communication de groupe (disposition de logo, informations légales).
- Savoir limiter le poids (en chargement) de la charte en fonction des objectifs et du public visé.
- Toute navigation doit être intuitive quel que soit le média. A ce titre, nous respecterons les règles d'usage d'accès aux informations à savoir : maximum 3 clics et 7 grandes rubriques d'informations par page. Le contenu et l'organisation des informations du site devront donc s'y plier.
- De quelle façon peut-on rappeler les principaux éléments et sections du site ? Par une bannière horizontale, verticale, un arbre de navigation ?
- Il est appréciable de disposer d'un plan du site afin de s'y retrouver.

V Le référencement et la gestion

V.1 Le référencement

Le référencement est une étape importante dans la création d'un site web, puisqu'il est inutile de mettre en ligne un site fabuleux s'il n'est pas visible par le public. En effet, le référencement a pour mission de faire connaître le site à des moteurs de recherche et à d'autres sites.

Il existe deux types de référencement:

- Le référencement automatique assuré par des "spiders" programme qui scrute les pages d'un site en s'appuyant sur les balises META situées dans la balise d'en-tête de page <HEAD> et la description des liens et des images.
- Le référencement manuel, comme sur Yahoo, c'est le fait d'ajouter l'URL du site à la liste d'URLs de ce moteur de recherche.

Il existe plusieurs façons de se référencer :

- De manière gratuite.
- De manière payante.

V.2 La gestion

Afin d'assurer la continuité du site, il est nécessaire de suivre son évolution, d'enrichir son contenu et de corriger les éventuelles erreurs, c'est la tâche du webmaster.

Afin d'anticiper sur les modifications à apporter et de vérifier l'audience du site, nous aurons recours à des outils statistiques. En effet, il est utile voir indispensable de quantifier et qualifier le trafic du site. Ces outils sont soit fournis par l'hébergeur, ou bien disponibles (même gratuitement) sur Internet, ils nous renseignent notamment sur :

- La provenance des visiteurs (pays, site référant, moteur de recherche).
- Les pages consultées, pendant combien de temps et à quelle heure.
- Le nombre de visites, visiteurs et pages vues.

Conclusion

L'objectif de cette première leçon est de donner une idée globale sur Internet et les phénomènes qui en dérivent. Les définitions et la description des différentes étapes qui ont marqué l'évolution du réseau mondial ont été présentées au début, ensuite on a abordé les différents services offerts par Internet et leurs utilisations, ainsi que les divers formats de fichiers manipulés sur ce réseau.

Tout au long de cette leçon on a énuméré les éléments qui tournent autour d'un site web, ainsi que les étapes à suivre avant de se lancer dans l'implémentation et le codage. Aussi on a vu les technologies qu'on peut utiliser afin de concevoir et de développer notre site, ainsi que les éléments de gestion et de référencement.

Nous allons passer à apprendre les outils permettant de créer et de programmer des contenus pour le Web.

CHAPITRE II

Les bases du HTML

Pré-requis

- Notion du Web.
- Les techniques de création d'un site Web.

Objectifs généraux

- ☞ Savoir concevoir un site web.
- ☞ Connaître la syntaxe des principales balises du langage HTML.

Plan

- Introduction.
- I. Le corps du document –body.
- II. Les métas.
- Conclusion.

Chapitre II : Les bases du HTML

Introduction

Le **HTML** (HyperText Markup Language) est le format de données conçus pour présenter des **pages web** qui pouvant être lues par des **navigateurs** (Exp : Netscape, Internet Explorer, Mozilla Firefox, Opera ...).

C'est un langage de présentation de données, et non pas un langage de programmation. Il est figé, c'est-à-dire qu'une fois le document chargé dans le navigateur, il ne répond à aucune action de l'utilisateur sur le contenu de la page. Ce langage est pourvu d'un système de balisage (basé sur les balises) qui va permettre de structurer les documents [4].

I Qu'est-ce qu'une balise HTML ?

Une balise HTML est un élément qui s'ajoute au texte de départ. La balise sera interprétée par le navigateur, «dire au navigateur de quelle manière va afficher le texte », elle est toujours délimitée par les signes « < » et « > ».

Une balise peut en plus comporter de zéro à plusieurs attributs. Les attributs sont des informations complémentaires qui la caractérisent. Ils se présentent sous la forme « nomattribut="valeur" ».

Exemple de balise et son attribut

```
<html lang="fr">
```

Chaque balise ouverte doit être fermée, cependant il existe des exceptions. On distingue deux types de balises :

- Les balises simples.
- Les balises doubles.

I.1 Les balises simples

Ce sont des balises qui sont dites "vides", c'est-à-dire qu'elles ne vont contenir aucune autre balise HTML. Ces balises n'ont pas besoin d'être fermées.

Exemple de balise simple

```

```

I.2 Les balises doubles

Les balises doubles sont dites ouvrantes/fermantes, c'est-à-dire qu'elles nécessitent deux balises, une ouvrante et une fermante dans lesquelles on va pouvoir mettre d'autres balises ou du texte.

La balise fermante est identique à la balise ouvrante, à la différence qu'elle contient un "/".

Exemple de balise double

```
<p>Ici du texte ou toute autre balise. </p>
```

Imbrication des balises

Lorsque'on cumule l'écriture de plusieurs balises ouvrantes (forcément inévitable), il est impératif de respecter la hiérarchie des balises, c'est-à-dire que la première ouverte sera la dernière fermée.

Le chevauchement des balises est une erreur grave.

Exemple de balises imbriquées

```
<b><u>Ici les balises sont correctement imbriquées.</u></b>
```

Exemple de balises chevauchées

```
<b><u>Ici les balises ne sont</b>pas correctement imbriquées, c'est une erreur.</u>
```

I.3 Structure d'un document HTML

Document HTML de base

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<title>Titre de la page</title>
<meta http-equiv="content-type" content="text/html; charset=utf-8">
</head>
<body>
<!-- Ici votre site -->
</body>
</html>
```

La première ligne du document s'appelle le **DOCTYPE**. C'est une balise indispensable pour conserver la compatibilité du rendu de votre page sur les différents navigateurs modernes.

I.4 Le DOCTYPE

Il sert d'une part à déclarer quel "langage", "grammaire", ou norme on applique à notre page.

Cette déclaration nous permet de soumettre nos pages au **validateur du W3C** (World Wide Web

Consortium) qui vérifie que notre code HTML ne comporte aucune erreur de syntaxe (balises mal fermées, utilisation d'attributs inexistantes...).

I.5 Les commentaires

Les commentaires permettent de laisser des informations ou des descriptions du code et sont utilisables à n'importe quel endroit du document.

Commentaires HTML

```
<!-- Voici un premier commentaire sur une ligne. -->
<!-- Voici un deuxième commentaire.
Celui-ci est multilignes. -->
<!------- Cette méthode est déconseillée ! ----->
```

II Le corps du document – body

II.1 Mise en forme

II.1.1 Les niveaux de titres

Il existe différents niveaux de titres, ils vont de 1 à 6 et ont chacun leur importance. La balise représentant ces niveaux de titre est **<hX>** où X représente le niveau.

Exemple de titre

```
<h1>Titre de ma page</h1>
```

II.1.2 Les paragraphes de texte

Comme tout document texte, un document HTML doit être segmenté en paragraphes. Cette segmentation est possible en précisant où commence chaque paragraphe et où il se termine. C'est le rôle de la balise **<p>**.

Les paragraphes de texte en HTML

```
<p>Voici un premier paragraphe. </p>
<p>Voici un deuxième paragraphe. </p>
```

II.1.3 La mise en forme de texte

Style	Balise	Aperçu
gras	Ce texte s'affichera en gras. 	Ce texte s'affichera en gras.
italique	<i>Ce texte s'affichera en italique. </i>	<i>Ce texte s'affichera en italique.</i>
souligné	<u>Ce texte sera souligné. </u>	<u>Ce texte sera souligné.</u>
barré	<s>Ce texte sera barré. </s>	Ce texte sera barré.

exposant	Ce texte sera en ^{exposant.}	Ce texte sera ^{exposant} .
indice	Ce texte sera en _{indice.}	Ce texte sera en _{indice} .

II.1.4 Modifier la couleur du texte

Pour modifier la couleur du texte on utilise l'attribut « **color** » de la balise « **** » :

Texte en rouge

```
<font color="#ff0000">Ce texte sera en rouge. </font>
```

Les couleurs peuvent être écrites de deux manières :

- En hexadécimal de type RVB et précédées d'un dièse (#) ;
Exemples : #ff0000 => rouge, #00ff00 => vert, #0000ff => bleu.
- Textuelles en anglais US ; Exemples : red, yellow, pink.

II.1.5 Modifier la police du texte

Pour modifier la police du texte on utilise l'attribut « **face** » de la balise « **** » :

Texte en verdana

```
<font face="verdana">Ce texte sera en verdana. </font>
```

On a tendance à écrire des familles de polices plutôt qu'une seule police. En effet, il se peut que sur un système, différent du notre, la police choisie ne soit pas installée. On sépare alors les différentes polices par une virgule.

Texte en verdana ou sans-serif

```
<font face="verdana,sans-serif">Ce texte sera en verdana ou en sans-serif si verdana n'est pas installée.</font>
```

II.1.6 Modifier la taille du texte

Pour modifier la taille du texte on utilise l'attribut « **size** » de la balise « **** » :

Texte en taille 5

```
<font size="5">Ce texte sera en taille 5.</font>
```

Par défaut, la valeur de l'attribut *size* vaut "3".

Il y a deux manières pour modifier la valeur de cet attribut :

- En absolu : Les valeurs possibles sont les entiers de "1" à "7".
- En relatif : Les valeurs possibles les entiers de "-2" à "+4".

Et il est bien évidemment possible de renseigner ces trois attributs (**color**, **face**, **size**) à la fois dans la même balise « ».

II.2 Les images

Pour afficher une image en HTML, on utilise la balise « ». Elle admet des différents attributs, dont les plus importants sont « **src** » et « **alt** ».

L'attribut « **src** » permet de renseigner le chemin de l'image, qui peut être **absolu** ou **relatif**. L'attribut « **alt** » affiche un texte alternatif à la place de l'image dans le cas où, pour une raison ou une autre, celle-ci ne peut pas être chargée. Il aide de plus les moteurs de recherche à référencer l'image.

Afficher une image

```

```

Les différents formats d'image supportés par cette balise sont :

- GIF
- JPG
- PNG
- BMP (à éviter).

II.3 Les liens hypertextes

Un lien hypertexte est un élément HTML permettant d'envoyer le visiteur vers une nouvelle page. On peut insérer dans cet élément toute sorte de textes, images et autres balises dites "**inline**" (comme les balises , , etc). Un lien sera représenté dans le code par la balise « <a> ». L'adresse de destination doit se trouver dans l'attribut « **href** ».

L'attribut « **title** », permet de renseigner un petit descriptif sur le contenu de la page cible, c'est un atout non négligeable en matière de référencement et d'accessibilité.

Exemple d'un lien texte

```
<a href="adresse/de/destination.html">Texte du lien</a>
```

Exemple d'un lien image

```
<a href="adresse/de/destination.html">  
</a>
```

On distingue deux types de liens, les liens externes et les liens internes.

II.3.1 Les liens externes

Un lien externe est un lien hypertexte qui renvoie vers une autre page. Cette autre page peut être une page du même.

II.3.2 Les liens internes

Un lien interne est un lien qui renvoie vers la même page, mais pas forcément au même endroit de la page.

Attention : A ne pas confondre "liens internes" avec "lien interne au site" !

Un lien interne au site est en général un lien externe à la page.

Les ancres permettent de naviguer à l'intérieur d'une même page.

Pour faire une ancre, il faut utiliser le signe dièse (#) suivi d'un mot-clef. Ce mot-clef sera repris dans l'attribut « **id** » d'une balise à laquelle on veut faire une ancre :

Exemple d'une ancre interne

```
<a href="#motClef">Texte de mon ancre</a>
<h2 id="motClef">Texte vers lequel l'ancre est renvoyée</h2>
```

Ou bien

```
<a name=" motClef "> Texte vers lequel l'ancre est renvoyée </a>
```

Si on veut faire une ancre vers une page externe, il suffit de rajouter l'adresse de destination dans la balise de lien :

Exemple d'une ancre externe

```
<a href="adresse/de/destination.html#motClef">Texte de mon ancre</a>
```

• Les liens e-mail

Exemple d'un lien vers une adresse e-mail

```
<a href="mailto:xxx@monsite.com">Ecrivez-moi</a>
```

⇒ *Ce type de liens ouvre le logiciel de messagerie par défaut des visiteurs.*

Exemple d'un lien vers une adresse e-mail avec un sujet par défaut

```
<a href="mailto:xxx@monsite.com?subject=Sujet du mail">Ecrivez_moi</a>
```

Exemple d'un lien vers une adresse e-mail avec un corps par défaut

```
<a href="mailto:xxx@monsite.com?body=Corps du mail">Ecrivez-moi</a>
```

Exemple d'un lien vers une adresse e-mail avec un sujet et un corps par défaut

```
<a href="mailto:xxx@monsite.com?subject=Sujet du mail&body=Corps du mail">Ecrivez-moi</a>
```

Exemple d'un lien vers plusieurs adresses e-mail

```
<a href="mailto:destinataire1@monsite.com;destinataire2@monsite.com"> Ecrivez-nous</a>
```


• Les Protocoles

Un protocole est un ensemble de conventions nécessaires qui permet à deux entités distantes de communiquer entre eux. Il diffère selon le type de données échangées. Il existe de nombreux protocoles disponibles par lien hypertexte, en voici une liste non exhaustive :

Protocole	Signification de l'acronyme	Description	Syntaxe utilisée
HTTP	HyperText Transfer Protocol	Permet de transférer des données sur l'Internet.	..
HTTPS	HyperText Transfer Protocol Secured	Permet de transférer des données sur l'Internet en permettant de vérifier l'identité du site grâce à un certificat d'authentification.	..
SMTP	Simple Mail Transfer Protocol	Permet de transférer des e-mails.	..</ a>
FTP	File Transfer Protocol	Permet de transférer des fichiers.	...
IRC	Internet Relay Chat	Permet de se connecter à un serveur de Chat.	...

II.3.3 Informations complémentaires

• Pour éviter d'avoir le cadre bleu autour d'une image lorsqu'elle est dans un lien, il suffit de renseigner l'attribut « **border** » à "0" :

Image sans bordure bleue

```
<a href="adresse/de/destination.html"></a>
```

• Si on veut avoir une info bulle qui s'affiche sur le lien lors du passage de la souris, on peut renseigner l'attribut « **title** » :

Info bulle sur le lien

```
<a href="adresse/de/destination.html" title="Description de la page cible">Texte du lien</a>
```

II.4 Les listes

Elles servent, comme leur nom l'indique, à créer une liste d'items, plus précisément à les énumérer, comme dans le cas d'un sommaire par exemple. On distingue cinq types différents de listes, chacun prévu pour un usage bien précis :

- Les listes non ordonnées, ou à puces.
- Les listes ordonnées.
- Les listes de définitions.
- Les listes de répertoires.
- Les listes de menus.

Dans ce présent chapitre nous allons étudier que les deux premiers types.

II.4.1 Les listes non ordonnées

Elles sont démarquées par la balise « **** » et chacun des items est entouré de la balise « **** » :

Exemple de liste non ordonnée

<pre><p>Ma liste de courses :</p> Champignons Avocat Dentifrice Céréales </pre>	<p>Ma liste de courses :</p> <ul style="list-style-type: none"> • Champignons • Avocat • Dentifrice • Céréales
---	--

➤ Les différents types de puces

On renseigne le type des puces avec l'attribut **type** de la balise ****.

Les types sont :

- **disc** (par défaut au niveau 1 d'imbrication)
- **circle** (par défaut au niveau 2 d'imbrication)
- **square** (par défaut au niveau 3 d'imbrication)

Dans l'exemple précédent, nous avons donc vu une liste non ordonnée avec une puce de type disc. Voici à quoi ressemblent les deux autres :

Exemple de liste non ordonnée

<pre><p>Ma liste de courses :</p> <ul type="circle"> Champignons Avocat Dentifrice Céréales </pre>	<p>Ma liste de courses :</p> <ul style="list-style-type: none"> ◦ Champignons ◦ Avocat ◦ Dentifrice ◦ Céréales
---	--

Exemple de liste

<pre><p>Ma liste de courses :</p> <ul type="square"> Champignons Avocat Dentifrice Céréales </pre>	<p>Ma liste de courses :</p> <ul style="list-style-type: none"> ■ Champignons ■ Avocat ■ Dentifrice ■ Céréales
---	--

```
</ul>
```

➤ **Les puces en image**

Pour avoir une puce sous forme d’image, il suffit d'utiliser cette image pour remplacer la puce originale. Cependant cette méthode n'est possible qu'à l'aide de CSS et de la propriété *list-style-image* (Voir chapitre IV).

Exemple de liste non ordonnée

<pre><p>Ma liste de courses :</p> Champignons Avocat Dentifrice Céréales </pre>	<pre>Ma liste de courses : ▶ Champignons ▶ Avocat ▶ Dentifrice ▶ Céréales</pre>
---	---

CSS (voir chapitre IV)

```
li{list-style-image : url(puce.gif);}
```

II.4.2 Les listes ordonnées

Elles sont introduites par la balise **** et chacun des items est encadré par la balise **** :

Exemple de liste ordonnée

<pre><p>Top 3 du tiercé :</p> Jolly Tonnerre Tornade </pre>	<pre>Top 3 du tiercé : 1. Jolly 2. Tonnerre 3. Tornade</pre>
--	--

Il existe cinq manières différentes de numérotter les listes ordonnées. On renseigne le type de numérotation avec l'attribut **type** de la balise ****.

Les types sont :

- **1** : Numérotation par les chiffres arabes (par défaut).
- **I** : Numérotation par les chiffres romains.
- **i** : Numérotation par les chiffres romains minuscules.
- **A** : Numérotation par les lettres de l'alphabet.
- **a** : Numérotation par les lettres de l'alphabet minuscules.

Dans l'exemple précédent, nous avons donc vu une liste ordonnée avec une numérotation par les chiffres arabes. Voici à quoi ressemblent les autres :

Exemple de liste ordonnée

<pre><p>Top 3 du tiercé :</p> <ol type="I"> Jolly Tonnerre Tornade </pre>	<p>Top 3 du tiercé :</p> <p>I. Jolly</p> <p>II. Tonnerre</p> <p>III. Tornade</p>
---	--

Exemple de liste ordonnée

<pre><p>Top 3 du tiercé :</p> <ol type="i"> Jolly Tonnerre Tornade </pre>	<p>Top 3 du tiercé :</p> <p>i. Jolly</p> <p>ii. Tonnerre</p> <p>iii. Tornade</p>
---	--

Il est tout à fait possible, pour une raison ou une autre, de modifier le numéro de départ par lequel va s'incrémenter la numérotation.

Pour cela, il suffit de préciser une valeur à l'attribut **start** de la balise ****. Cette valeur doit dans tous les cas être numérique :

Exemple de liste ordonnée

<pre><p>Top 3 inversé du tiercé :</p> <ol type="1" start="8"> Tortue Escargot Limace </pre>	
---	--

II.4.3 Les listes imbriquées

L'imbrication des listes permet une présentation claire et détaillée d'un menu ou d'un sommaire par exemple. On peut imbriquer les listes non ordonnées et ordonnées sans distinction et même les mélanger :

Exemple de listes imbriquées :

<pre><ol type="I"> La genèse La vie <ol type="A"> L'enfance L'adolescence Le passage à l'âge adulte Le jeune adulte La crise de la quarantaine La vieillesse </pre>	<p>I. La genèse</p> <p>II. La vie</p> <p>A. L'enfance</p> <p>B. L'adolescence</p> <p>C. Le passage à l'âge adulte</p> <p> ■ Le jeune adulte</p> <p> ■ La crise de la quarantaine</p> <p>D. La vieillesse</p> <p>III. La mort</p>
---	--

```
</ol>  
</li>  
<li>La mort</li>  
</ol>
```

II.4.4 L'utilité des listes aujourd'hui

La principale utilisation des listes de nos jours, se fait par les listes non ordonnées pour la création de divers menus. L'imbrication de ces listes non ordonnées couplée avec un soupçon de *CSS* permet de faire des menus à multiples niveaux d'affichage, comme les menus déroulants. D'autre part, on peut retrouver les listes ordonnées principalement pour la rédaction de sommaires.

Conclusion

Ce chapitre a fait l'objet des notions de base du langage HTML, ses caractéristiques et ses types.

De même, on a présenté la structure générale d'un document HTML ainsi que les balises de base pour le formatage du contenu d'un document HTML.

CHAPITRE III

Les notions avancées du HTML

Pré-requis

- ☞ Notions de base du HTML

Objectifs généraux

- ☞ Introduire les notions HTML avancées tels que les tableaux, les formulaires.
- ☞ S'introduire au langage XHTML.

Plan :

- Introduction.
- I. Les tableaux.
- II. Les formulaires.
- III. Le langage XHTML.
- Conclusion.

Chapitre III Les notions avancées du HTML

Introduction

Nous avons vu dans les chapitres précédents, l'essentiel du langage HTML, pour créer les premières pages web. Dans ce chapitre, nous allons montrer l'avantage d'utiliser des notions HTML plus avancées tels que les tableaux, les formulaires ainsi que l'extension du langage HTML : XHTML, pour supporter le standard XML.

I Les tableaux

I.1 À quoi servent-ils ?

Un tableau est une suite de lignes et de colonnes qui forment un ensemble de cellules. Les tableaux servent avant tout à présenter des données tabulaires dont voici un exemple :

Elève	Math	Web	SGBD	Sport
Ali	14	10	11	11
Mohamed	15	14.5	10.5	12
Meriem	7.5	18	12	14
Kaled	8.5	12	9	15

Malgré cela, l'emploi le plus fréquent des tableaux reste la mise en page des documents. En effet, l'implémentation des tableaux étant assez intuitive, s'ajoute à cela les éditeurs **WYSIWYG** des logiciels de création de sites qui génèrent ce genre de code automatiquement.

Cependant, la mise en page par tableau devient dépréciée.

I.2 Du tableau simple au tableau complexe

• Les bases d'un tableau

Un tableau est délimité par la balise « **<table>** ». Le contenu d'un tableau HTML est généré horizontalement. C'est-à-dire qu'il n'est pas créé colonne par colonne mais ligne par ligne. Pour créer un tableau, nous commençons par créer des lignes grâce aux balises « **<tr>** ». Puis nous insérons dans ces lignes les cellules du tableau grâce aux balises « **<td>** ». Le contenu (textes, images, autres balises) se trouve uniquement dans les balises « **<td>** » et ne doit se trouver **en aucun cas** en dehors.

Donc pour avoir un tableau d'une cellule, soit une ligne et une colonne, on obtient ce code :

Tableau d'une cellule

<pre><table border="1"> <tr> <td>Contenu de la cellule</td> </tr> </table></pre>	Contenu de la cellule
--	-----------------------

Et pour un tableau de deux lignes et deux colonnes :

Tableau de deux lignes et deux colonnes

<pre><table border="1"> <tr> <td>Cellule 1</td> <td>Cellule 2</td> </tr> <tr> <td>Cellule 3</td> <td>Cellule 4</td> </tr> </table></pre>	<table border="1"> <tr> <td>Cellule 1</td> <td>Cellule 2</td> </tr> <tr> <td>Cellule 3</td> <td>Cellule 4</td> </tr> </table>	Cellule 1	Cellule 2	Cellule 3	Cellule 4
Cellule 1	Cellule 2				
Cellule 3	Cellule 4				

Par défaut, la bordure n'est pas visible. Dans les codes précédents on a rajouté l'attribut « **border** » à la balise « `<table>` » pour la faire apparaître. On peut évidemment agrandir la bordure en augmentant la valeur.

Il y a encore deux éléments (définis par défaut mais modifiables) :

L'espace entre les cellules ou l'épaisseur des lignes du quadrillage	<TABLE cellpadding=?>
L'enrobage des cellules ou l'espace entre le bord et le contenu	<TABLE cellspacing=?>

Exemple :

<pre><TABLE border=2 cellpadding= 10> <TR> <TD>1</TD> <TD>2</TD> </TR> <TR> <TD>3</TD> <TD>4</TD> </TR> </TABLE></pre>	<table border="2"> <tr> <td>1</td> <td>2</td> </tr> <tr> <td>3</td> <td>4</td> </tr> </table>	1	2	3	4
1	2				
3	4				

<pre><TABLE border=2 cellspacing=10> <TR> <TD>1</TD> <TD>2</TD> </TR> <TR> <TD>3</TD> <TD>4</TD></pre>	<table border="2"> <tr> <td>1</td> <td>2</td> </tr> <tr> <td>3</td> <td>4</td> </tr> </table>	1	2	3	4
1	2				
3	4				


```
<TD>4</TD>
</TR>
</TABLE>
```

• **Le titre d'un tableau**

Pour décrire le contenu d'un tableau, par exemple, pour expliquer dans notre premier exemple de tableau qu'il s'agit des notes de la classe de TI20, il faut placer la balise «<caption>» directement après l'ouverture du tableau. Cette balise prend l'attribut «*align*» qui accepte comme valeurs «top» (par défaut), «bottom», «left» et «right», en fonction de l'endroit où l'on veut afficher cette légende.

• **La ligne d'en-tête d'un tableau**

Dans l'exemple des notes de la classe de TI20, on remarque que la première ligne peut faire office d'en-tête. En effet, elle renseigne sur le contenu de chacune des colonnes et n'a donc pas le même genre de contenu que tout le reste du tableau.

Pour préciser au navigateur qu'il s'agit donc de cellules spéciales, on va les modifier en remplaçant pour cette ligne seulement les balises «<td>» par des balises «<th>» :

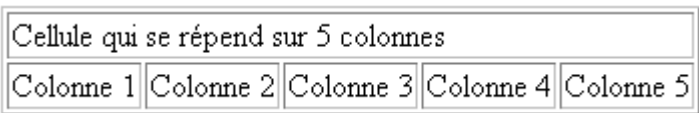
• **La fusion de lignes et de colonnes d'un tableau**

On distingue deux types de fusion : la fusion horizontale où sont mises en jeu deux (ou plus) colonnes, la fusion verticale où sont mises en jeu deux (ou plus) lignes. Dans les deux cas, elle s'opère sur les cellules, donc sur la balise «<td>». Elle se traduit par l'utilisation de l'attribut «**colspan**» dans le cas d'une fusion horizontale et par l'attribut «**rowspan**» dans le cas d'une fusion verticale.

Ces attributs prennent pour valeurs le nombre de cellules à fusionner.

Exemple d'utilisation de colspan

```
<table border="1">
<tr>
<td colspan="5">Cellule qui se réprend sur 5 colonnes</td>
</tr>
<tr>
<td>Colonne 1</td>
<td>Colonne 2</td>
<td>Colonne 3</td>
<td>Colonne 4</td>
<td>Colonne 5</td>
</tr>
</table>
```



⇒ La somme des « colspan » de chaque ligne doit être identique à toutes les autres lignes.

En ce qui concerne l'attribut « rowspan », on aurait un code tel que :

Exemple d'utilisation de rowspan

```

<table border="1">
<tr>
<td rowspan="2">Cellule sur deux lignes</td>
<td>Ligne 1 - Colonne 2</td>
<td>Ligne 1 - Colonne 3</td>
<td>Ligne 1 - Colonne 4</td>
</tr>
<tr>
<td>Ligne 2 - Colonne 2</td>
<td>Ligne 2 - Colonne 3</td>
<td>Ligne 2 - Colonne 4</td>
</tr>
</table>
    
```

Cellule sur deux lignes	Ligne 1 - Colonne 2	Ligne 1 - Colonne 3	Ligne 1 - Colonne 4
	Ligne 2 - Colonne 2	Ligne 2 - Colonne 3	Ligne 2 - Colonne 4

On pourrait aller plus loin en combinant les deux techniques sur un seul et même tableau :

Exercice : réalisez le tableau suivant

Cellule 1		Cellule 2	
	Cellule 4	Cellule 5	Cellule 6
Cellule 3	Cellule 7	Cellule 8	
	Cellule 9		

Correction

```

<table border="1">
<tr>
<td colspan="3">Cellule 1</td>
<td>Cellule 2</td>
</tr>
<tr>
<td rowspan="3">Cellule 3</td>
<td>Cellule 4</td>
<td>Cellule 5</td>
<td>Cellule 6</td>
</tr>
<tr>
<td>Cellule 7</td>
<td colspan="2" rowspan="2">Cellule 8</td>
</tr>
<tr>
<td>Cellule 9</td>
</tr>
</table>
    
```

On remarque que le code est devenu extrêmement compliqué et que la moindre erreur de fusion de notre part entraînerait une interprétation du tableau complètement fautive par les navigateurs.

I.3 L'utilité des tableaux aujourd'hui

Grâce aux techniques de fusion vues précédemment, il y a quelques années, la tendance était de profiter de ce système pour mettre en page assez rapidement l'ensemble de son site web. Cependant, si cette méthode peut séduire, elle est fortement déconseillée.

En effet, les tableaux sont censés (et doivent) être utilisés uniquement pour l'affichage de données tabulaires.

Dans le cas d'une mise en page, on conseille d'adopter la solution du couple "DIV + CSS". (**Voir chapitre 4**).

On peut facilement voir dans notre dernier exemple où l'on combine les deux techniques de fusion, que notre code sera difficilement maintenable dans le cas d'une retouche ou de l'ajout d'une nouvelle cellule.

Voici quelques raisons pour lesquelles il ne faut pas utiliser les tableaux pour la mise en page :

- Ils alourdissent les pages HTML : beaucoup de balises pour finalement y mettre peu de contenu.
- Ils sont très difficiles à maintenir dans le cas d'une retouche.
- Ils nuisent à l'accessibilité : problème d'interprétation des navigateurs spécialisés destinés aux malvoyants par exemple.
- Ils sont mal gérés par les navigateurs à l'impression dans le cas de tableaux complexes.

II Les formulaires

II.1 À quoi servent-ils ?

Les formulaires servent à envoyer des données au serveur. Ces informations sont remplies en général par un visiteur. Par exemple : inscription sur un site, un formulaire de contact, etc.

Les formulaires sont dotés de divers contrôles comme des champs de saisie, des boutons, des listes de choix, etc., qui permettent au visiteur d'interagir avec la page qu'il est en train de consulter.

II.2 La balise form

Tous les champs de formulaires doivent se trouver dans cette balise « **<form>** ».

C'est cette balise qui va permettre de renseigner la page de destination du formulaire, à l'aide de l'attribut « **action** », qui peut prendre pour valeurs une URL en absolu ou en relatif, voire même une adresse e-mail.

Elle est aussi utilisée pour préciser le mode d'envoi des données grâce à l'attribut « **method** ». Cet attribut prend deux valeurs différentes qui sont "**get**" et "**post**". Les valeurs envoyées en "GET" passent par l'URL alors qu'en "POST" elles sont envoyées de manière transparente.

Exemple de balise form

```
<form action="envoi-formulaire.php" method="post">
<!-- Contenu du formulaire -->
</form>
```

II.3 Les champs texte mono lignes

On distingue deux types de champs texte monolignes : les champs de type "**texte**" et les champs de type "**mot de passe**".

Le champ de type "texte" est l'élément de base d'un formulaire, il permet d'introduire du texte court sur une ligne, comme un login, un e-mail, ou autres, alors que le champ de type "mot de passe" permet de renseigner un mot de passe, dont l'affichage sera camouflé aux yeux des autres personnes.

Le champ de texte monoligne est caractérisé par la balise « **<input>** » et l'attribut « **type** » dont la valeur sera « "text" » ou « "password" ».

On renseignera aussi l'attribut « **name** » qui sera le nom du champ, celui qui va permettre de retrouver la valeur lors de l'envoi du formulaire.

D'autres attributs sont disponibles, comme « **value** » qui permet d'attribuer une valeur par défaut au champ, « **size** » qui permet de préciser la taille du champ en nombre de caractères, « **maxlength** » qui sert à limiter le nombre de caractères possibles, « **readonly** » qui prend pour seule valeur « "readonly" » et verrouille le champ, « **disabled** » qui prend pour seule valeur « "disabled" » et permet de désactiver le champ.

⇒ *Lorsqu'un champ est défini en "disabled", sa valeur n'est pas envoyée au serveur.*

Voici quelques exemples de champs texte :

Exemples de champs texte

```
<input type="text" size="5" name="code_postal" maxlength="5">
<input type="text" name="nom_champ" readonly="readonly" value="Champ
verrouillé">
<input type="text" name="login" value="Entrez ici votre login" size="20">
<input type="password" name="mdp">
```

II.3.1 Les champs texte multilignes

Contrairement au champ de texte monoligne, le **textarea** est une balise double et son contenu est écrit entre les balises ouvrantes et fermantes et non dans un attribut value.

Deux attributs sont nécessaires au « <textarea> » : « **rows** » qui contient le nombre de lignes et « **cols** » qui contient le nombre de colonnes.

Tout comme le champ de texte monoligne, on peut, rajouter les attributs « name », « readonly » et « disabled ».

L'attribut « *maxlength* » n'existe pas pour cette balise, et pour en limiter le nombre de caractères il est

possible de passer par du **JavaScript**.

Exemples de champs texte multilignes

```
<textarea name="commentaire" rows="5" cols="80"></textarea>
<textarea name="sujet" rows="6" cols="100">Texte par défaut</textarea>
```

II.3.2 Les champs cachés

Il nous arrivera souvent de vouloir envoyer des données au serveur de manière transparente pour le visiteur. On utilisera alors un champ caché, qui ne sera pas affiché dans le navigateur du visiteur mais dont les données seront envoyées. Attention, cette balise apparaît tout de même dans le code source de la page.

Pour cela, on utilise la balise « `<input>` » avec « `"hidden"` » comme valeur de l'attribut « **type** ».

Il est aussi nécessaire de renseigner l'attribut « `name` » pour retrouver ces données cachées du côté du serveur ainsi que l'attribut « `value` » qui va contenir ces données.

Exemple de champ caché

```
<input type="hidden" name="clef" value="a182f7d8e844d956a65b18e84f">
```

II.3.3 Les champs de fichiers

Pour permettre à l'internaute, d'envoyer des fichiers par l'intermédiaire du formulaire, que ce soit en pièce jointe d'un e-mail ou le chargement d'une image sur le serveur, on utilise la balise « `<input>` » dont l'attribut « `type` » est renseigné à « **"file"** ».

Ceci fait, un champ avec un bouton « "Parcourir" » qui sera disponible.

Mais pour que le champ soit opérationnel, il faut impérativement renseigner l'attribut « **enctype** » de la balise `<form>` à **"multipart/form-data"**.

Il est aussi plus qu'utile de préciser l'attribut « `name` » du champ « "file" ». On peut aussi renseigner l'attribut « `size` » qui affecte la taille du champ dans lequel sera écrit le nom du fichier.

Exemple de champ file

```
<input type="file" name="image">
```

Il est également possible de limiter le poids du fichier en rajoutant un champ caché dans le formulaire contenant le champ "file". Ce champ caché doit avoir pour `name` la valeur **"MAX_FILE_SIZE"** et pour `value` le poids maximum en octet.

Limitation du poids du champ file

```
<input type="hidden" name="MAX_FILE_SIZE" value="20000">
```

⇒ Cette limitation n'affranchit pas de faire la vérification côté serveur ! En effet, le code HTML peut facilement être modifié par les visiteurs, donc cette valeur aussi.

II.4 Les boutons radio

Les boutons radio sont utilisés pour laisser aux visiteurs un choix et un seul parmi une liste de propositions. La balise est alors « `<input>` » dont l'attribut « `type` » est renseigné à « `"radio"` ».

Ces boutons radio vont par groupe, c'est-à-dire qu'ils doivent avoir le même nom pour un groupe de propositions. Le nom est renseigné avec l'attribut « `name` ».

La valeur de l'attribut « `value` » va être transmise au serveur en fonction du bouton choisi. On peut aussi forcer un bouton radio d'un groupe à être coché au chargement de la page en lui mettant l'attribut « `checked` » dont la seule valeur est « `"checked"` ».

Exemples de groupes de boutons radio :

```
<p>
<input type="radio" name="civilité" value="mlle"> Mademoiselle
<input type="radio" name="civilité" value="mme"> Madame
<input type="radio" name="civilité" value="mr"> Monsieur
</p>
<p>
<input type="radio" name="genre" value="homme" checked="checked"> Homme
<input type="radio" name="genre" value="femme"> Femme
<input type="radio" name="genre" value="et"> Alien
<input type="radio" name="genre" value="indéfini"> Indéfini
</p>
```

⇒ Si on donne un nom différent aux boutons d'un groupe, le navigateur **ne sait pas qu'ils appartiennent au même groupe**, donc plusieurs choix seront possibles et on perd toute l'utilité des boutons radio

II.5 Les cases à cocher

Les cases à cocher sont sensiblement identiques aux boutons radio, mais permettent un choix multiple pour un groupe de propositions. Les attributs sont identiques, à la différence que l'attribut « `type` » prend pour valeur « `"checkbox"` ».

Un problème peut arriver, lorsque vous désignez un groupe de cases à cocher par le même nom, seule la dernière sera envoyée au serveur. Pour remédier à ceci, il suffit de rajouter des crochets ouvrant et fermant au nom du groupe. Il indiquera au serveur qu'il s'agit d'un tableau de valeurs à récupérer et non une seule valeur (comme dans le cas des boutons radio). En revanche, dans le cas d'utilisation d'une case à cocher unique (sans groupe), les crochets sont inutiles.

Exemple de cases à cocher

```
<input type="checkbox" name="qualite[]" value="intelligent"> Intelligent
<input type="checkbox" name="qualite[]" value="beau"> Beau
<input type="checkbox" name="qualite[]" value="serviable"> Serviabile
<input type="checkbox" name="qualite[]" value="fort"> Fort
<input type="checkbox" name="qualite[]" value="généreux"> Généreux
```

II.6 Les boutons

Il existe deux balises différentes pour présenter les boutons : la balise « *<input>* » et la balise « *<button>* ». La différence entre les deux est que la balise « *<button>* » est double, donc qu'elle a une balise ouvrante et une balise fermante alors que la balise « *<input>* » est simple.

Il y a 4 types différents de boutons :

- Le bouton simple
- Le bouton d'envoi
- Le bouton image
- Le bouton effacer

• Le bouton simple

Il est défini par l'attribut « *type* » des balises « *<input>* » et « *<button>* » ayant pour valeur « "button" ».

Ce bouton n'est utile que lors d'appel de scripts JavaScript au moment du clic. Il n'a aucune autre fonction particulière. Exemples de boutons simples :

Exemples de boutons simples

```
<input type="button" name="addition" value="Additionner" onclick="addition();" >
<button type="button" name="multiplication"
onclick="multiplication();" ><b>Multiplier</b></button>
```

• Le bouton d'envoi

Le bouton d'envoi sert, comme son nom l'indique, à envoyer les données du formulaire au serveur. C'est celui qui va permettre de valider le formulaire.

Il est caractérisé par la valeur "submit" de l'attribut *type* des balises *<input>* et *<button>*.

Exemples de boutons d'envoi

```
<input type="submit" name="envoyer" value="Valider" >
<button type="submit" name="soumettre">Soumettre<br>les réponses</button>
```

• Le bouton image

Le bouton image se comporte exactement comme un bouton d'envoi, c'est-à-dire qu'il sert à la validation du formulaire. Il sert à personnaliser un bouton.

Il est défini par l'attribut « *type* » de la balise « *<input>* » ayant pour valeur « "image" » et l'on renseigne l'adresse de l'image avec l'attribut « *src* ». C'est le seul bouton qui ne peut pas être géré par la balise « *<button>* ».

Exemple de bouton image

```
<input type="image" name="envoyer" src="adresse/de/l/image" >
```

• **Le bouton effacer**

Ce bouton sert à réinitialiser toutes les valeurs du formulaire. Il est caractérisé par la valeur « "reset" » de l'attribut « *type* » des balises « <input> » et « <button> ».

Exemples de boutons effacer

```
<input type="reset" name="effacer" value="Effacer toutes les valeurs">
<button type="reset" name="effacer">Réinitialiser</button>
```

II.7 Les listes déroulantes

Les listes permettent de laisser un choix aux visiteurs, au même titre que les boutons radio ou les cases à cocher, mais sont plus pratiques lorsqu'il s'agit de proposer un grand nombre de choix.

On distingue deux types de listes, les listes normales et les listes déroulantes.

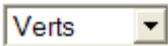
Les deux sont introduites par la balise « <select> » qui comprend les différents choix sous forme de balise « <option> » dont on renseigne la valeur qui sera envoyée au serveur grâce à son attribut « *value* ».

On peut même suggérer une proposition par défaut en précisant l'attribut « *selected* » de la balise « <option> » de notre choix. Par défaut, une liste est déroulante.

On peut (doit) renseigner le nom de la liste avec l'attribut « *name* ».

Exemple de liste

```
<select name="couleur_yeux">
<option value="bleus">Bleus</option>
<option value="bruns" selected="selected">Bruns</option>
<option value="verts">Verts</option>
<option value="noisettes">Noisettes</option>
<option value="gris">Gris</option>
<option value="vairons">Vairons</option>
</select>
```



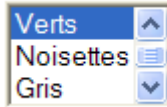
Cette liste est donc une liste déroulante avec un seul choix possible parmi les six proposés.

On peut rendre cette liste non déroulante en rajoutant l'attribut « *size* » qui prend pour valeur le nom de choix à afficher, par exemple "3".

Ce qui donne :

Exemple de liste

```
<select name="couleur_yeux" size="3">
<option value="bleus">Bleus</option>
<option value="bruns">Bruns</option>
<option value="verts" selected="selected">Verts</option>
<option value="noisettes">Noisettes</option>
<option value="gris">Gris</option>
<option value="vairons">Vairons</option>
```





```
</select>
```

Par défaut, un seul choix est possible. Pour pouvoir sélectionner plusieurs choix, comme dans le cas de cases à cocher, il suffit de rajouter l'attribut « *multiple* » qui prend pour seule valeur « "multiple" ».

Exemple de liste à choix multiple

```
<select name="matieres_preferees" size="5" multiple="multiple">
<option value="maths">Maths</option>
<option value="français">Français</option>
<option value="anglais">Anglais</option>
<option value="histoire">Histoire</option>
<option value="sport">Sport</option>
<option value="dessin">Dessin</option>
<option value="musique">Musique</option>
<option value="physique">Physique</option>
<option value="sciences naturelles">Sciences naturelles</option>
</select>
```



⇒ *Le choix multiple ne peut pas se faire sur une liste déroulante. L'attribut size doit être renseigné pour que le multiple fonctionne.*

III Le XHTML

III.1 Définition

eXtensible HyperText Markup Language, dit XHTML, est un langage de balisage servant à écrire des pages pour le World Wide Web. Il est conçu à l'origine comme le successeur d'HTML, XHTML se fonde sur la syntaxe définie par XML, plus récente, mais plus exigeante que celle définie par SGML (langage normalisé de balisage généralisé) sur laquelle repose HTML : il s'agit en effet de présenter un contenu affichable non seulement par les ordinateurs classiques, mais également sans trop de dégradation par des PDA (Personal Digital Assistant) [1].

III.2 Validation W3C

Le **W3C, Worl Wide Web Consortium**, est un organisme travaillant sur la mise en place de standards Web.

Ce qui nous intéresse plus particulièrement, ici, est leur système de validation de code source.

Pour valider le code source, le système du W3C, va vérifier la syntaxe du code conformément aux règles établies.

Ce n'est pas dénué de sens de valider sa page:

- Permet d'avoir un code portable, compatible avec la plupart des navigateurs.
- Permet d'avoir un code standardisé et connu de tout le monde.

- Permet une meilleure accessibilité.

• **Ouverture avec un DOCTYPE**

Une des particularités des langages de la famille XML est de pouvoir définir ses propres balises par l’intermédiaire d’un DTD (*Document Type Definition*). La référence à un DTD est indispensable car il contient la liste et la définition des balises utilisées dans le document.

Tout document XHTML devra donc commencer avec une balise qui indique à l’agent utilisateur (le logiciel de navigation) comment interpréter les autres balises du document. Voici un exemple de déclaration de DOCTYPE.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
```

• **Espace de nommage (namespace)**

La déclaration du type de document XHTML doit être suivie par la déclaration de l’espace de nommage suivant, inséré dans l’élément **html** :

```
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr" lang="fr">
```

Le DTD se place au début de tout document XHTML. La déclaration du DOCTYPE peut parfois être précédée du prologue XML inséré en tout début du document :

```
<?xml version="1.0" encoding="iso-8859-1"?>
```

Un des rôles du prologue XML est de spécifier le type d’encodage des caractères qui est utilisé à l’intérieur du document. C’est l’encodage qui est spécifié lors de l’enregistrement du document.

Si le prologue est oublié, on peut tout aussi bien spécifier le type d’encodage en utilisant, à cet effet, un élément « **meta** » :

```
<meta http-equiv="content-type" content="text/html; charset=iso-8859-1" />
```

III.3 Syntaxe

- *Le document DOIT comporter au minimum les éléments de structure html, head et body.*

De plus, l’élément de titre **title** DOIT être présent dans l’élément **head**.

Le XML est très rigoureux quant à la structure du document. La structure de base d’un document XHTML est :

```
<!DOCTYPE ...>
<html ... >
  <head> ... <title> ... </title> ... </head>
  <body>
    ...
```

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

• **L'écriture des balises en bas de la case (minuscules)**

XML étant sensible aux majuscules/minuscules (case sensitive), toutes les balises du XHTML doivent être écrites en minuscules, tant pour les éléments que les attributs. Par contre, les valeurs peuvent ne pas respecter cette règle.

XHTML	HTML
<code><p>Bonjour</p></code>	<code><P>Bonjour</P></code>
<code><title>Le titre</title></code>	<code><TITLE>Le titre</TITLE></code>

• **La valeur des attributs doit toujours être écrites entre des guillemets (et ce même pour des valeurs numériques).**

XHTML	HTML
<code><table colspan="4"></code>	<code><table colspan=4></code>
<code>Test</code>	<code>Test</code>

• **Les balises d'images IMG doivent TOUJOURS comporter l'attribut alt pour le texte alternatif.**

XHTML	HTML
<code></code>	<code></code>

• **A toute balise d'ouverture doit correspondre une balise de fermeture.**

XHTML	HTML
<code><p>Bonjour</p></code>	<code><p>Bonjour</code>
<code></code>	<code></code>
<code>1</code>	<code>1</code>
<code>2</code>	<code>2</code>
<code></code>	<code></code>

• **Les balises uniques doivent également être refermées.**

XHTML	HTML
<code>
</code>	<code>
</code>
<code><hr /></code>	<code><hr></code>
<code></code>	<code></code>

L'utilisation d'une seconde balise fermante n'est pas en théorie erronée, par exemple `
</br>`, mais les navigateurs des anciennes générations peuvent ne pas l'interpréter correctement, c'est pourquoi `
` (avec un espace avant la barre oblique /) est conseillé. Il en est de même pour les balises **input**, **hr**, **meta** et **link** :

```
<meta http-equiv="content-type" content="text/html; charset=iso-8859-1" />
<link rel="stylesheet" href="styles.css" type="text/css" />
```

- *Les balises doivent être correctement imbriquées.*

XHTML	HTML
<code><i>gras et italique</i></code>	<code><i>gras et italique</i></code>

La règle générale est la suivante :

la dernière balise ouverte doit être la première refermée.

- *On ne doit pas utiliser de notations compactes ou raccourcies dans les attributs.*

XHTML	HTML
<code><hr noshade="noshade" /></code> <code><input checked="checked"></code> <code><option selected="selected"></code> <code><frame noresize="noresize"></code>	<code><hr noshade></code> <code><input checked></code> <code><option selected></code> <code><frame noresize></code>

- *On utilisera l'attribut ID plutôt que l'attribut NAME*

Pour identifier un élément comme A, APPLET, FRAME, IFRAME, IMG, et MAP on utilise l'attribut 'id'.

XHTML	HTML
<code>Ancre</code>	<code>Ancre</code>

Toutefois, on notera que pour garder une compatibilité avec les navigateurs des générations précédentes, on peut garder à la fois **name** et **id** avec les mêmes attributs comme ceci

```
<a id="haut" name="haut">Ancre</a>
```

Mais cette méthode n'est valide que dans un document utilisant un DTD XHTML 1.0 « Transitional ».

Conclusion

Ce chapitre a comme but de savoir les éléments qu'on peut trouver dans un document html, ainsi les propriétés utiles, cette liste des balises et propriétés reste toujours non exhaustive. Le langage HTML continu à évoluer depuis sa création jusqu'à nos jours. Cependant il garde toujours certains inconvénients notamment la non standardisation vis à vis les différents navigateurs qui a été résolu par le standard XHTML. Un autre problème pour les développeurs Web qui réside à la surcharge des pages HTML avec les balises de mise en forme, cet inconvénient est disparu avec l'apparition des feuilles de style. Dans le chapitre suivant nous allons étudier les différents concepts du CSS.

CHAPITRE IV

Les feuilles de styles

Pré-requis

- Langage HTML, XHTML

Objectifs généraux

- ☞ S'introduire à la mise en page des pages web en utilisant le langage CSS.

Plan

- Introduction
- I. Utilité et avantages
- II. Définition
- III. Les classes et les identificateurs
- IV. Les pseudo-éléments et les pseudo-classes
- V. Les balises structurales
- VI. Le flux
- VII. Le positionnement
- Conclusion

Chapitre IV : Les feuilles de style

Introduction

Le langage CSS (Cascading Style Sheets) est utilisé pour **définir l'aspect futur des sites web**, comme par exemple la couleur du fond de la page ou le type de police.

Plus concrètement, le **CSS** (ou feuille de style) est un document numérique qui va pouvoir spécifier toutes les caractéristiques de mise en forme du document lié à la balise à laquelle elle s'applique.

• Exemple concret

Dans un document d'une certaine importance, il arrive fréquemment que l'on attribue à certains éléments des caractéristiques de mise en forme identiques. Par exemple, les noms des chapitres seront mis en police Arial, en gras et en couleur bleu.

On peut imaginer qu'on peut donner à cette définition de mise en forme un nom soit "titre" et qu'à chaque nouveau chapitre, plutôt que d'écrire chaque fois le nom du titre et puis le mettre en Arial, gras, bleu, on peut affecter au nom du chapitre cette mise en forme, déjà défini, sous le nom de "titre". Cette définition de mise en forme particulière, on va l'appeler feuille de style.

<code><H1>Titre1</H1></code>	STYLE des titres
<code><H2>- A -</H2></code>	STYLE sous-titres
<code><H3>...a...</H3></code>	STYLE du texte
<code><H1>Titre2</H1></code>	STYLE des titres
<code><H2>- B-</H2></code>	STYLE sous-titres
<code><H3>...b...</H3></code>	STYLE du texte

On parle :

- de feuilles de style [style sheets] car le but du jeu est d'en définir plusieurs.
- de feuilles de style en cascade [Cascading Style Sheets ou CSS] car en cas de styles identiques, un ordre de priorité sera déterminé par le browser

I Utilité et avantages

Les apports des feuilles de style peuvent être résumés comme suit [2]. :

- Séparation du contenu et de la mise en forme.
- Cohésion de la présentation tout au long du site avec les feuilles de style externes.
- Modification de l'aspect d'une page ou d'un site sans en modifier le contenu et cela en quelques lignes plutôt que de devoir changer un grand nombre de balises.
- Ecriture concise et nette par rapport au Html qui devient vite fouillis.
- Réduction du temps de chargement des pages.

- Dépassement des certaines insuffisances du langage Html (contrôle des polices, contrôle de la distance entre les lignes, contrôle des marges (sans devoir utiliser de tableaux ou de balise <DD>...) et ainsi augmenter la créativité des écrivains du Web.
- Permettre le positionnement au pixel près du texte et/ou des images..

II Définition

La syntaxe d'un style est toujours la même, elle précise la balise à laquelle le style va s'appliquer, et les différents attributs du style. Les attributs sont enfermés entre deux accolades ou deux guillemets (précédés de Style=) selon les cas, et chacun des attributs est séparé par un point virgule. On donne la valeur de l'attribut par la syntaxe suivante "nom de l'attribut: valeur de l'attribut"

balise {propriété de style: valeur; propriété de style: valeur}

Exemple :

```
H3 {font-family: Arial; font-style: italic }
```

Donc ici, la balise H3 sera en Arial et en italique. Et dans le document en question, toutes les balises <H3> auront comme style Arial et italique.

NB :

- Pour la lisibilité, on doit écrire les styles sur plusieurs lignes :

```
H3 {font-family: Arial;
font-style: italic;
font-color: green}
```

- On peut attribuer plusieurs valeurs à une même propriété. Dans ce cas, on séparera les différentes valeurs par des virgules.

```
H3 {font-family: Arial, Helvetica, sans-serif}
```

- On peut attribuer un même style à plusieurs balises (séparées par des virgules).

```
H1, H2, H3 {font-family: Arial; font-style: italic}
```

Il existe trois manières d'utiliser et d'intégrer les feuilles de style dans un document HTML, **intra-lignes**, **globales**, ou **importées**. Chacune de ces manières a ses avantages et ses particularités.

II.1 CSS Globale

A l'intérieur des balises `<HEAD></HEAD>` :

Une feuille globale va se déclarer dans la tête du document, entre les balises « HEAD ». Elle va s'appliquer globalement à toutes les balises du document qu'elle a défini.

```
<HTML>
<HEAD>
<STYLE type="text/css">
<!--
La ou les feuille(s) de style
-->
</STYLE>
</HEAD>
<BODY>
```

- La balise « `<STYLE>` » avertit le navigateur que l'on va utiliser des feuilles de style.
- L'attribut `type="text/css"` informe que ce qui suit est du texte et qu'il s'agit de cascading style sheets (css).
- La balise Html de commentaires `<!-- ... -->`, empêche les browsers qui ne connaissent pas les feuilles de style, d'interpréter ces instructions. Les informations à l'intérieur des tags de commentaires seront ignorées par ces browsers.
- Pour les commentaires à propos des feuilles de style, on utilisera la convention désormais classique `/* commentaires */` de (C, C++, Javascript...).

II.2 CSS Intra-lignes

A coté des balises :

Une feuille intra-ligne s'insère directement à côté de la balise qu'elle définit, elle ne constitue donc pas véritablement une feuille, simplement elle permet localement de fixer des attributs à une partie d'un document. Sa déclaration est un peu particulière, en voici une (notez la forme `Style=" "`):

```
<HTML>
<BODY>
<H1 style="font-family: Arial; font-style: italic"> blabla </H1>
</BODY>
</HTML>
```

Signalons que:

- Le style Arial, italique n'affectera que cette seule balise H1.
- La syntaxe est légèrement différente de la précédente.

- L'écriture : `<STYLE type="text/css">H1 { "font-family: Arial; font-style: italic" }</STYLE>` fonctionne aussi.

II.3 CSS importée

Dans un document séparé :

Une feuille liée va se déclarer à part, dans un fichier avec une extension `.css` qui sera mis à côté des autres documents dans le répertoire du site web. Cette feuille de style sera valable pour toutes les pages du site qui l'appelleront dans les balises `HEAD`. C'est un outil très puissant pour uniformiser la mise en pages d'un grand nombre de documents.

Principe :

On crée d'abord, dans le répertoire du site, un fichier avec l'extension `.css` soit `styles.css` qui contiendra toutes les feuilles de style.

```
TD {font-family:Arial,Helvetica; font-size:10pt}
```

Ensuite, on crée une page soit `page.htm` avec dans la `<HEAD>` un lien vers ce fichier CSS :

```
<HTML>
<HEAD>
<LINK rel=stylesheet type="text/css" href="styles.css">
</HEAD>
```

Commentaires :

- La balise `<LINK>` avertit le browser qu'il faudra réaliser un lien.
- L'attribut `rel=stylesheet` (sans s et sans guillemets) précise qu'il y trouvera une feuille de style externe.
- L'attribut `type="text/css"` précise que l'information est du texte et du genre cascading style sheets (css).
- L'attribut classique de lien `href=" ... "` donne le chemin d'accès et le nom du fichier à lier.

III Les classes et les ID

III.1 Notion de classes

III.1.1 Définition

Un sélecteur contextuel, ou classe, associe une règle particulière à un élément en fonction de sa situation [2]. On appelle *sélecteur simple* les balises HTML classiques auxquelles on a attribué des caractéristiques de style. Mais ces sélecteurs simples présentent une contrainte importante, définie de

manière générale, il est difficile de changer momentanément certains de leurs attributs ou bien d'en rajouter. Pour résoudre ce problème, les concepteurs du CSS ont inventé les classes. Une classe est ce qu'on appelle un sélecteur contextuel. Il va pouvoir venir modifier contextuellement une caractéristique du document.

La définition d'un style était :

```
balise { propriété de style: valeur }
```

Elle devient :

```
balise.nom_de_classe { propriété de style: valeur }
```

Remarquez le point entre balise et nom_de_classe (indispensable).

Ou, comme la mention de la balise est facultative,

```
.nom_de_classe { propriété de style: valeur }
```

Pour appeler l'effet de style dans le document, on ajoute le nom de la classe à la balise.

```
<balise class="nom_de-classe"> .... </balise>
```

Exemple :

On souhaite mettre ce qui est important dans le texte en gras et en bleu. On crée la classe **.Toto** :

```
.Toto { font-weight: bold; font-color: #000080 }
```

Et dans le document Html, il suffit d'appeler la classe quand cela se révèle nécessaire :

```
<P class="Toto"> ... blabla ... </P>
<H1 class="Toto ">Titre 1</H1>
<TABLE><TR><TD class="Toto ">cellule</TD></TD>...
```

III.2 Notion des ID

Lorsqu'on désire utiliser les feuilles de style avec du Javascript on utilise les ID (appelés les identifiants). Les ID fonctionnent exactement comme les classes. La syntaxe est :

```
#nom_de_ID { propriété de style: valeur }
```

Et pour l'appeler :

```
<balise id="nom_de_ID"> .... </balise>
```

Notons : on ne pourra effectuer qu'un seul appel à **#nom_de_ID** par page. Ainsi, Pour :

```
#Toto { ... }
<P id=Toto> est correct, mais si on le rencontre dans la même page :
<H1 id=Toto> ce n'est plus correct !
```

IV Les pseudo-éléments et les pseudo-classes

Ils permettent une mise en forme à partir d'informations absentes des propriétés déjà vues.

Les pseudo-classes sont autorisées n'importe où dans le sélecteur, alors qu'un pseudo-élément ne peut être spécifié qu'après le sujet du sélecteur.

IV.1 Les pseudo-classes

- **La pseudo-classe : first-child**

La pseudo-classe : « first-child » correspond au premier élément enfant d'un autre élément. Dans l'exemple suivant, le sélecteur désigne chacun des éléments **P** qui soit le premier enfant d'un élément DIV. La règle supprime l'alinéa du premier paragraphe d'un élément DIV :

```
DIV > P:first-child { text-indent: 0 }
```

Exemple(s) :

Dans l'extrait suivant, ce même sélecteur viserait l'élément P dans l'élément DIV :

```
<P> Le dernier P avant la note.
<DIV class="note">
  <P> Le premier P à l'intérieur de la note.
</DIV>
```

Mais, dans cet extrait ci-dessous, il ne toucherait pas le second élément P :

```
<P> Le dernier P avant la note.
<DIV class="note">
  <H2>Note</H2>
  <P> Le premier P à l'intérieur de la note.
</DIV>
```

- **Les pseudo-classes d'ancre : :link et :visited**

En général, les agents utilisateurs représentent différemment les liens qui n'ont pas été visités de ceux qui l'ont déjà été. CSS en fournit un équivalent au travers des pseudo-classes ':link' et ':visited'.

- La pseudo-classe :link s'applique aux liens qui n'ont pas été visités ;

- La pseudo-classe `:visited` s'applique lorsque le lien a été visité par l'utilisateur.

Les deux déclarations CSS2 suivantes produisent le même effet :

```
A:link { color: red }
:link { color: red }Exemple(s) :
<A class="external" href="http://www.mon lien">lien extern</A>
A.external:visited { color: blue }
```

Celle-ci rendre bleue la couleur du lien visité.

• **Les pseudo-classes dynamiques : `:hover`, `:active` et `:focus`**

Les navigateurs interactifs changent parfois l'aspect du rendu en réponse aux actions de l'utilisateur. CSS2 propose trois pseudo-classes pour un usage courant :

- La pseudo-classe « `:hover` », qui est appliquée quand l'utilisateur désigne un élément (au moyen d'un appareil de pointage) sans l'activer. Par exemple, un navigateur usuel pourrait appliquer celle-ci quand le pointeur (d'une souris) survole la boîte produite par un élément.
- La pseudo-classe « `: active` », qui est appliquée quand l'utilisateur active un élément. Par exemple, entre le moment où l'utilisateur presse le bouton de la souris et le relâche.
- La pseudo-classe « `:focus` », qui s'applique quand un élément reçoit le curseur (celui-ci acceptant les événements du clavier ou d'autres formes d'entrées de texte).

Exemple :

```
A:link { color: red } /* lien non-visité */
A:visited { color: blue } /* lien visité */
A:hover { color: yellow } /* lien survolé */
A:active { color: lime } /* lien activé */
```

Notons que la règle **`A:hover`** doit être placé après **`A:link`** et **`A:visited`**, autrement les règles de cascade feront que la propriété **'color'** spécifiée par celle-ci sera cachée. De la même façon, comme la règle **`A:active`** est placée après **`A:hover`**, la couleur spécifiée par celle-ci (lime) sera appliquée quand l'utilisateur active et survole l'élément A.

Voici un exemple de combinaison de pseudo-classes dynamiques :

```
A:focus { background: yellow }
A:focus:hover { background: white }
```

Le dernier sélecteur correspond à un élément **A** qui a l'attention et qui est survolé.

IV.2 Les pseudo-éléments

• Le pseudo-élément `:first-line`

Le pseudo-élément « `:first-line` » produit un style particulier sur la première ligne formatée d'un paragraphe. Par exemple :

```
P:first-line { text-transform: uppercase }
```

La règle précédente signifie "mettre les lettres de la première ligne de chaque paragraphe en majuscule". Notons que la longueur de la première ligne dépend de plusieurs facteurs, dont la largeur de la page, la taille de la police, etc. Ainsi un paragraphe quelconque en HTML tel que :

```
<P>Voici un paragraphe HTML plutôt  
long qui va être découpé en plusieurs  
lignes. La première ligne sera identifiée  
à l'aide d'une séquence de balises fictives.  
Les autres lignes vont rester ordinaires  
dans la suite du paragraphe.</P>
```

Celui-ci va apparaître avec le découpage des lignes suivant :

```
VOICI UN PARAGRAPHE HTML PLUTÔT LONG  
qui va être découpé en plusieurs lignes.  
La première ligne sera identifiée à  
l'aide d'une séquence de balises fictives.  
Les autres lignes vont rester ordinaires  
dans la suite du paragraphe.
```

• Le pseudo-élément `:first-letter`

Le pseudo-élément « `:first-letter` » peut être employé pour faire des capitales initiales et des lettrines, ce sont des effets typographiques courants.

Cette feuille de style produit une lettrine qui s'étend sur deux lignes :

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0//EN">  
<HTML>  
<HEAD>  
<TITLE>Lettrine</TITLE>  
<STYLE type="text/css">  
  P      { font-size: 12pt; line-height: 12pt }  
  P:first-letter { font-size: 200%; font-style: italic;  
                  font-weight: bold; float: left }  
</STYLE>  
</HEAD>  
<BODY>
```

```
<P>Les premiers mots parmi ceux d'un article  
de journal.</P>  
</BODY>  
</HTML>
```

Ce qui pourrait apparaître de cette manière :

Les premiers mots parmi ceux d'un article
de journal.

V Les balises structurales

Il fallait penser à un système pour "déconnecter" certains morceaux de paragraphe ou plusieurs paragraphes de cette logique d'écriture avec des feuilles de style. Ce sont les balises structurales qui créaient ainsi des petits blocs particuliers dans le document sans devoir repasser par les éléments structurels du Html classique.

En CSS, il existe deux grandes familles d'éléments : les éléments de type "**bloc**", et les éléments de style "**en-ligne**".

Les éléments de type "bloc" (DIV, UL, FORM, BLOCKQUOTE, PRE, ...), servent à la mise en page générale de la page en créant de grands rectangles 'conteneurs'. Ils peuvent contenir soit d'autres éléments de type "bloc", soit des éléments de type "en-ligne".

Les éléments de type "en-ligne" (A, B, FONT, IMG, INPUT, U, SPAN, ...) sont juste des conteneurs plus petits que bloc, avec un comportement spécifique.

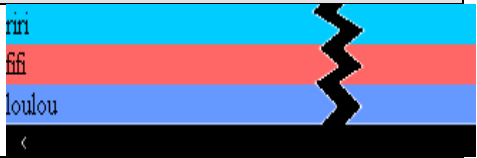

D'une manière générale, évitez d'utiliser les balises HTML pour la mise en forme du contenu, dans l'idée de séparer au maximum le fond de la forme (le tag U en HTML peut être avantageusement remplacé par `text-decoration:underline;` en CSS).

De plus, ces balises ont des marges externes (`margin`) et internes (`padding`) définies par défaut à 0 et non redéfinissables.

• Le flux

Le flux, c'est l'ordre d'affichage des éléments. Ainsi, le navigateur parcourt la page HTML, récupère les différents éléments, et les affiche dans un certain ordre. Les éléments sont affichés selon leurs ordres d'apparition dans le code source. Les éléments de type "bloc" vont s'afficher les uns aux dessous des autres, alors que des éléments de type "en-ligne" vont s'afficher les uns à la suite des autres.

```
Exemple :CSS commun aux deux types d'affichage
#element1 {
background-color:#00CCFF; /* bleu ciel */
}
#element2 {
background-color:#FF6666; /* rouge */
}
#element3 {
background-color:#6699FF; /* mauve */}
```

Affichage de type 'bloc' : HTML	Aperçu
<pre><div id="element1">riri</div> <div id="element2">fifi</div> <div id="element3">loulou</div></pre>	
Affichage de type 'en-ligne' : HTML	Aperçu
<pre>pim pam poum</pre>	

VI Le positionnement

VI.1 Généralités

La propriété CSS **position** permet de modifier le comportement d'affichage des blocs énuméré comme suit :

- **position:static** : indique que l'élément doit se comporter normalement.
- **position:inherit** (CSS2) : déclare que l'élément en question hérite de la propriété **position** de son parent. Si la position du parent est de type absolu, l'élément concerné le sera aussi.
- **position:fixed** : L'idée est de placer un élément sur la page, et qu'il y reste quel que soit le défilement.
- **position:relative** : ne retire pas l'élément du flux, mais le déplace simplement par rapport à sa position dans le flux.
- **position:absolute** : retire complètement l'élément concerné du flux, et le place aux coordonnées définies par les propriétés **bottom**, **left**, **right** ou **top**.

VI.1.1 Position absolue

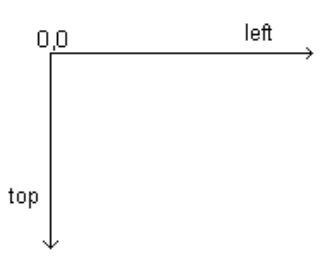
La position absolue *{position: absolute}* se détermine par rapport au coin supérieur gauche de la fenêtre du browser. Les coordonnées de ce point sont « top = 0 » et « left = 0 ».

Il y a trois options pour **top** et **left** :

- auto (par défaut).
- pixels (vous précisez la valeur suivie de px).
- pourcentage (la même chose suivie de %).

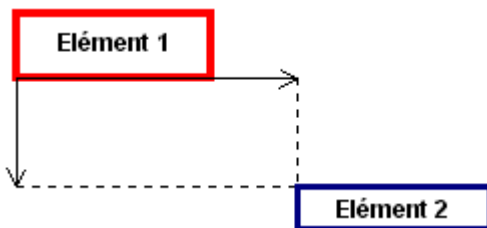
Les coordonnées d'un point de haut en bas pour **top** et de gauche à droite pour **left**.

Pour appliquer une position absolue à un texte on peut utiliser les balises « SPAN » ou « DIV ». La balise « DIV » est préférable pour les grands textes qu'elle fera précéder et suivre d'un saut de ligne; la balise « SPAN », conçue comme simple marqueur, sera utilisée pour encadrer un paragraphe.



VI.1.2 Position relative

La position relative *{position: relative}* se détermine par rapport à d'autres éléments de la page, par exemple un élément du code Html. Un positionnement relatif sera donc traité dans le flux du document (qui est traité de bas en haut), et fera référence pour son positionnement à l'élément qui lui est immédiatement supérieur. Cette propriété est valable pour tous les types de balises.



Exemple d'application :

Soit l'image *un.jpg* :

1

```
.toto {
position: absolute; top: 10px; left: 10px;
}
<BODY BGCOLOR="#000080" TEXT="#FF0000">
<IMG SRC="1.jpg" BORDER=0 WIDTH=50 HEIGHT=50>
<B><DIV CLASS="toto">Toto et titi</DIV></B>
```


Donne en position absolue :

Donne en position relative avec "Toto et Tata":



VI.2 Attributs

VI.2.1 Clip

Il s'applique uniquement aux éléments de position absolue. Il spécifié la zone de visibilité du document, c'est à dire l'endroit où le document contenu dans les balises pourra ne plus être visible s'il dépasse les paramètres de largeurs et de hauteurs fixés par clip. La syntaxe de clip varie selon les navigateurs, mais les recommandations du W3C sont du type:

clip:rect(haut,droite,bas,gauche)

Exemple : prenons l'image *un.jpg* avec CLIP

```
<HEAD>
<STYLE TYPE="text/css">
<!--
.toto {
position:absolute;
top:10px;
left:10px;
clip:rect(0, 25, 25, 0);
}
-->
</STYLE>
</HEAD>
<BODY BGCOLOR="#000080" TEXT="#FF0000">
<DIV CLASS="toto"><IMG SRC="1.jpg" BORDER=0 WIDTH=50 HEIGHT=50></DIV>
</BODY>
```

Donne :



Remarquons que les $\frac{3}{4}$ de l'image ont disparu.

VI.2.2 Z-index

Z-index, c'est l'attribut qui permet de définir l'empilement des feuilles de styles. Il indique l'axe vertical d'empilement, ou la priorité d'affichage entre les éléments superposés. Il s'applique à tous les éléments de position relative ou absolue. Plus l'index est grand, plus l'élément est situé dessus. Deux éléments de même index vont se superposer. C'est très pratique pour afficher du texte sur une image, ou vice-versa.

Exemple avec les deux images *un.jpg* et *deux.jpg* :

1

2

```
<STYLE TYPE="text/css">
<!--
.toto {
position:absolute; top:10px; left:10px;
}
.titi {
position:absolute; top:30; left:30; z-index:2;
}
-->
</STYLE>
</HEAD>
<BODY BGCOLOR="#000080" TEXT="#FF0000">
<DIV CLASS="toto"><IMG SRC="1.jpg" BORDER=0 WIDTH=50 HEIGHT=50></DIV>
<DIV CLASS="titi"><IMG SRC="2.jpg" BORDER=0 WIDTH=50 HEIGHT=50></DIV>
```

Donne :



Si l'on place le z-index:2 sur la classe Toto, on aura :



Conclusion

Ce chapitre était consacré à étudier les différents concepts des feuilles de style ainsi que leur importance dans la création de sites web élégants et bien organisés. Les CSS, comme toutes les technologies de développement Web, évoluent au même rythme que les navigateurs, ceci nous invite à s'aligner avec toutes ces évolutions tout en profitant des bonnes pratiques de développement Web.

Bibliographie

- [1] Christophe Porteneuve, Bien developers pour le Web 2.0, ©Groupe Eyrolles, 2007.
- [2] Pierre-Baptiste Nageon, Tutos «**Notion de base, CSS**» , 2 Mai 2006

Webographie

- [3] Christophe MAGDELAIN <http://www.netalya.com/fr/Article2.asp?CLE=121>
 - [4] Nicolas Chu <http://www.netalya.com/fr/HTML-1.asp>
-