Université Mohammed Premier
Faculté Pluridisciplinaire
Nador

Polycopié :

Technologies du web

Professeur : ***Mohamed ATOUNTI***

Département de Mathématiques et Informatique

Filière : Sciences Mathématiques et Informatique

Parcours : Administrateur des systèmes réseaux et technolologies web

Semestre : 3

Années universitaire : 2014-2016

Table des matières

Chapitre 1 :Introduction au web

Qu'est ce que Internet ?

Internet a été dérivé du concept d'interconnecter des réseaux dont la première utilisation remonte à octobre1972.

Internet : réseau mondial d'ordinateurs permettant aux utilisateurs de communiquer, de publier des informations, de transférer des données, de travailler à distance . . .

Historique :

1971: Naissance d'ARPAnet, Réseau sûr pour l'armée.

1973 : Apparition du TCP/IP.

1983 :Utilisationd'ARPAnet comme réseau de recherche. La base technique (TCP/IP) sera introduite au niveau international dans les années qui suivent.

1989 : Internet se profile comme réseau de recherche.

1993 : Explosion d'Internet suite à la popularité du WWW (apparition du premier navigateur)

1995/1996 : Percée de solutions Intranet/Internet dans le commerce (Intranet est un réseau TCP/IP fermé)

…. Commercialisation

Les fonctions les plus utilisées :

- WEB
- Courrier électronique : Envoyer des messages sous forme de fichiers texte. Peut être envoyé partout dans le monde: Accès Internet ; Un compte sur un serveur de messagerie.
- Recherche d'informations
- Messagerie instantanée
- Les groupes de discussion
- Radios, tv, multimédias,…

- Transfert de fichiers
- Les groupes de discussion
- Etc,…

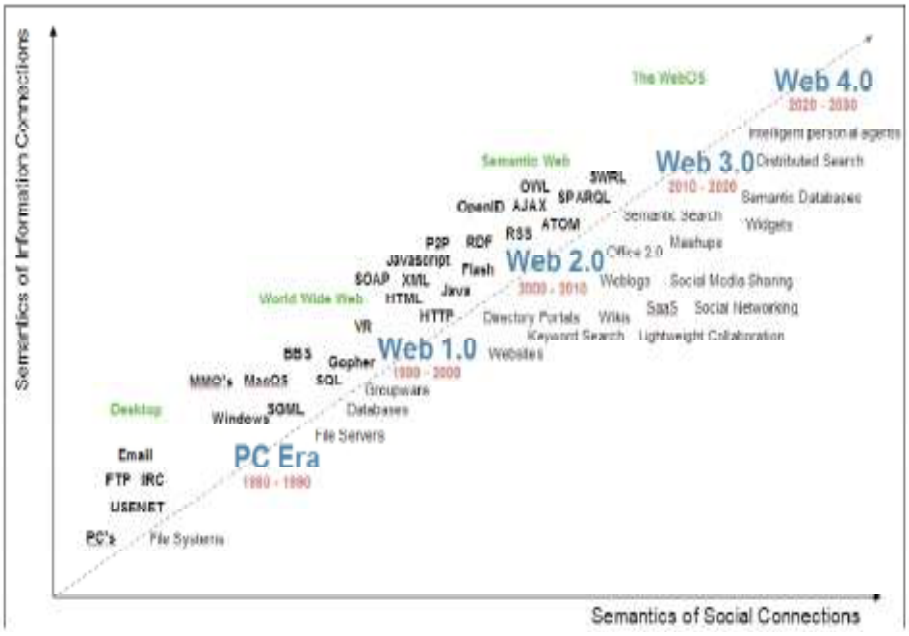
Qu'est-ce que le Web (ou World Wide Web, Toile, WWW, W3) ?

Système hypertexte public : système contenant des documents liés entre eux par des hyperliens permettant de passer automatiquement d'un document à l'autre.

Web est un des services offerts par le réseau Internet pour naviguer ; grâce à un navigateur ; et consulter des pages mises en lignes et reliées par des liens hypertextes (système hypertexte).

Evolution du web :

- Caractérisé par des pages Web statiques, rarement mises à jour.
- Considéré principalement comme un outil de diffusion et de visualisation de données.
- Evolution progressive vers un Web dynamique.



Web 2.0 : Une évolution et révolution :

Une nouvelle interactivité et un travail collaboratif : L'internaute = créateur de contenu et consommateur de contenu ; Diffusion de l'expérience de l'internaute et consulter celle des autres.

Nouveaux concepts liés à l'ergonomie: moins de clics, plus d'informations affichées à l'écran, moins de temps de chargement.

Nouvelles technologies : langage AJAX peut rendre les pages interactives et fluides.

Les réseaux sociaux : échange réalisé par des techniques synchrones comme les messageries instantanées, la téléphonie sur internet, ... ou des méthodes asynchrones comme les forums, les wikis, les messageries en ligne,…. (Facebook, MySpace, Twitter, Viadeo, LinkedIn, etc).

Web sémantique (le web intelligent):

Les informations ne seraient plus stockées mais "comprises" par les ordinateurs afin d'apporter à l'utilisateur ce qu'il cherche vraiment.

Objectif : Transformer la masse ingérable des pages Web en un gigantesque index hiérarchisé.

Web 3.0:

Web sémantique + Mobilité (applications disponibles sur tout type de support et notamment les mobiles).

Les serveurs web

- Logiciel permettant à des clients d'accéder à des pages web, à partir d'un navigateur installé sur leur ordinateur distant.
- Un serveur Web peut être:
  - ✓ Un ordinateur tenant le rôle de serveur informatique sur lequel fonctionne un logiciel serveur HTTP;
  - ✓ Le serveur HTTP lui-même;
  - ✓ Un ensemble de serveurs permettant le fonctionnement d'applications Web.

- Plusieurs serveurs Web:
  - ✓ Apache Windows/ UNIX : Le serveur le plus répandu sur Internet. S'appuie sur les protocoles HTTP ou HTTPS. Une application fonctionnant sur les systèmes d'exploitation de type Microsoft, Unix. Fonctionnalités : Configuration assez simple ; Accès sécurisé en fonction des adresses IP ; Chargement de modules pour ajouter de nouvelles fonctionnalités (php, mysql, ssl, ...) ; Etc,…
  - ✓ Microsoft IIS (Internet Information Services) : le serveur Web payant de Microsoft
  - ✓ Microsoft PWS (Personal Web Server).
  - ✓ Iplanet Web Server : serveur payant de Netscape ; Etc…

Accès aux serveurs Web : URL (Uniform Ressource Locator)

Une chaîne de caractères utilisée pour adresser les ressources dans le Web.

Exemple : http://www.exemple.com/chemin/page.html?q=req

http : protocole

www.exemple.com : hôte

/chemin/ : chemin absolu sur le service

page.html : nom de la page Web

q=req : chaine de requête, transmise à la page

Les pages Web

- Une ressource du World Wide Web
- Créée par des webmasters à l'aide des langages HTML/XHTML et CSS
- Possède une adresse Web.
- Peut contenir du texte, des images, des tableaux, des formulaires et autres éléments multimédias
- Visualisée par les internautes grâce à des navigateurs Web

Des navigateurs pour voir les pages Web

- Le logiciel le plus important sur l'ordinateur
- Grâce aux navigateurs, on peut:
  - ✓ lancer des recherches,
  - ✓ chatter,
  - ✓ échanger des e-mails,
  - ✓ faire des achats,
  - ✓ consulter votre compte en banque,
  - ✓ lire l'actualité,

- ✓ visionner des vidéos, etc.

- Rôle : analyser le code (X)HTML et CSS des pages web et d'en produire un résultat visuel, facile à lire pour un humain.

Navigateurs :

Il en existe un très grand nombre :

Navigateurs graphiques

Internet Explorer (Windows) ;

Mozilla Firefox (Windows, Mac et Linux) ;

Opera (Windows, Mac et Linux) ;

Safari (Mac, Windows)

Maxthon (Windows);

Google Chrome (Windows, Mac et Linux) ;

Konqueror (Linux), etc.

Navigateurs textuels

Links

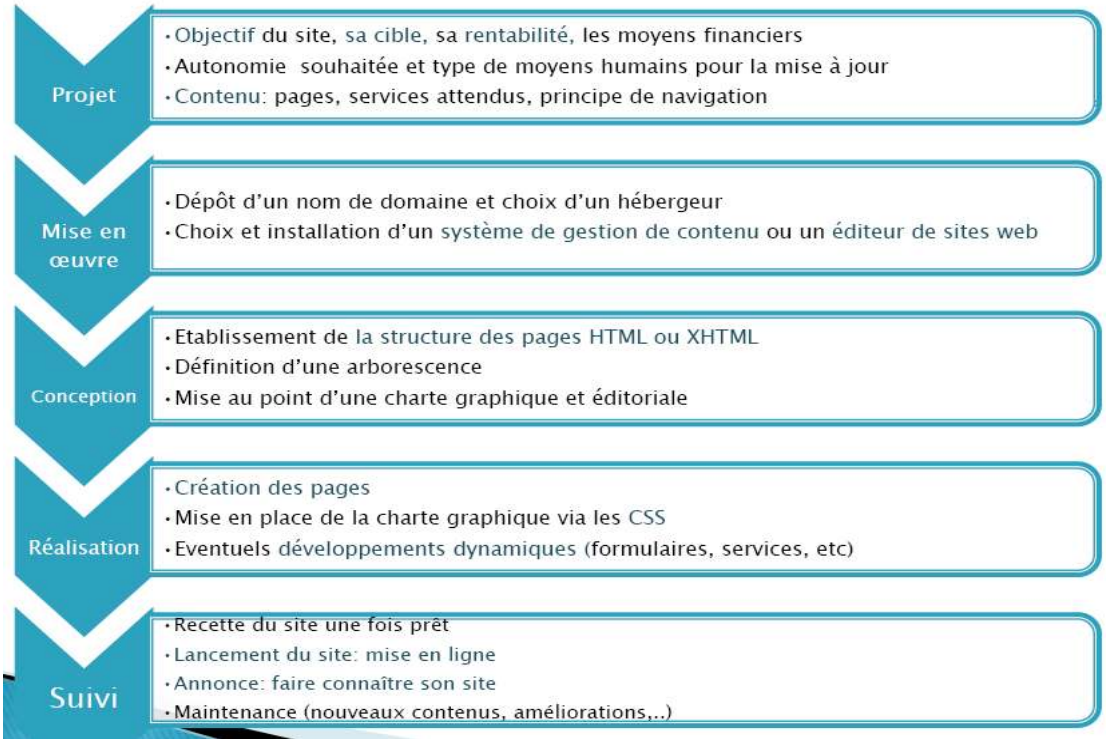
Linux, etc.

Comment créer un site web ?

- Etapes de création
- Quel type de pages créer?
- Editeurs de sites web
- Systèmes de gestion de contenu

Poser des bonnes questions pour créer un site web.

Les étapes de création d'un site web



Création de sites web

- Utiliser un éditeur. Il en existe deux types :

  Éditeurs WYSIWYG: permettent de créer un site à la manière d'un traitement de texte.

  Éditeurs de texte.
- Ou Utiliser un système de gestion de contenus (CMS: Content Management System)
- Connaitre et utiliser langages du Web (HTML, CSS)
- Tester continuellement son site sur au moins deux navigateurs à la fois pendant sa création, afin d'être sûr que tous vos visiteurs aient un résultat correct.

Quel type de pages web ?

- **Pages statiques**: chacune des pages est créée en HTML. Un ordinateur qui se connecte au serveur, demande une page. Celle ci lui est directement servie. Le serveur web (HTTP) se contente d'envoyer des fichiers stockés sur disque dur.
- **Pages dynamiques**: les pages peuvent être générées dynamiquement, en fonction des informations données par le navigateur (liens cliqués, formulaires, cookies...) et par le serveur (base de données SQL, fichiers de configuration...).
- Le serveur web (HTTP) exécute un ou plusieurs programmes qui vont renvoyer des données, que le serveur web retransmet au navigateur

- Le contenu est obtenu en combinant l'utilisation d'un langage de scripts ou de programmation et une base de données. Il s'agit souvent de PHP pour le langage et MySQL pour la base de données.

Les éditeurs de sites web :

Ce sont les programmes qui permettent de créer des sites web.

Les éditeurs WYSIWYG :

WYSIWYG : What You See Is What You Get, c'est-à-dire « ce que vous voyez est ce que vous obtenez ». Permet de rédiger le contenu du site sans avoir à taper la moindre ligne de code.

Logiciel	Système d'exploitation	Prix	Qualité	Adresse
Nvu	Win/Mac/linux	Gratuit/ OS	Assez bonne	http://www.nvu.com
Adobe Golive	Win et Mac	580 euros	Moyenne	http://www.adobe.fr/products/golive/
Macromedia Dreamweaver	Win et Mac	480 euros	Moyenne	http://www.adobe.fr/products/dreamweaver
FrontPage	Win et Mac	250 euros	Mauvaise	http://www.Microsoft.com/france/frontpage
Word	Win et Mac	290 euros	Très mauvaise	http://www.microsoft.com/france/word

Les éditeurs de texte

Gratuits pour la plupart

Logiciel	Système d'exploitation	Adresse
Bloc-Notes	Windows	Aucune (avec Windows)
Notepad++	Windows	http://notepad-plus.sourceforge.net
PSPad	Windows	http://www.pspad.com/fr
jEdit	Windows, Mac et Linux	http://www.jedit.org/
Emacs	Linux, windows et Maw	http://www.gnu.org/software/emacs/

Et d'autres.

Système de gestion de contenu (SGC ou CMS)

- Se charge de la partie commune aux pages du site et génère les pages à partir du texte ou des données qui lui sont fournis.
- Peut être statique et créer les pages avant qu'elles ne soient mises en lignes, ou dynamique et créer la page à la demande du visiteur.

- N'apporte rien quand au contenu lui-même, mais permet de gérer la structure du site utilisation de FAQ (Foire Aux Questions), de documents, de blogs, de forums de discussion, etc.), ajouter et classer les pages
- Permet d'adapter l'interface du site, de générer des pages imprimables plus dépouillées ou de faire participer des contributeurs au site.

Les blogs:

Wordpress : le plus répandu. Génère automatiquement les pages HTML à partir du texte édité qui est stocké dans une base de données. On peut créer un post sur son blog par envoi d'un e-mail. http://www.wordpress-fr.net/

DotClear : Prend en charge l'administration du blog, recherches, catégorisation, etc. http://fr.dotclear.org/

Les wikis : des sites dont le contenu est édité par les visiteurs

Media Wiki utilise PHP et MySQL. http://www.mediawiki.org/wiki/MediaWiki/fr

PmWiki et **DokuWikisont** en PHP mais n'utilisent pas de base de donnée. http://www.pmwiki.org/wiki/PmWikiFr/PmWikiFr

Les forums

PunBB http://punbb.informer.com/

phpBB est le plus utilisé sur les sites dédiés au forum. http://www.phpbb-fr.com/

MyBB et **SMF** sont équivalents en fonctionnalités à phpBB.

Les portails

Joomla. Projet collaboratif de CMS Internet et intranet en PHP. http://www.joomla.fr/

Drupal. http://drupalfr.org/

Xoops. http://www.frxoops.org/

Chapitre 2 :Protocole http

Introduction au protocole HTTP

Protocole : Ensemble normalisé de règles décrivant la manière de transmettre des informations, par exemple sur un réseau comme Internet entre un client et un serveur.

HTTP : HyperText Transfer Protocol, le plus utilisé des protocoles de communication sur le World Wide Web. Permet à un client Web d'indiquer quelle page il veut obtenir, et au serveur Web de lui répondre en lui donnant cette page.

Le protocole HTTP (HyperText Transfer Protocol) est le protocole le plus utilisé sur Internet depuis 1990.

La version 0.9 était uniquement destinée à transférer des données sur Internet, en particulier des pages Web écrites en HTML.

La version 1.0 du protocole permet de transférer des messages avec des en-têtes décrivant le contenu du message en utilisant un codage de type MIME.

Le but du protocole HTTP est de permettre un transfert de fichiers ; essentiellement au format HTML ; localisés grâce à une chaîne de caractères appelée **URL** entre un **navigateur** (le client) et un **serveurWeb**.

URL :

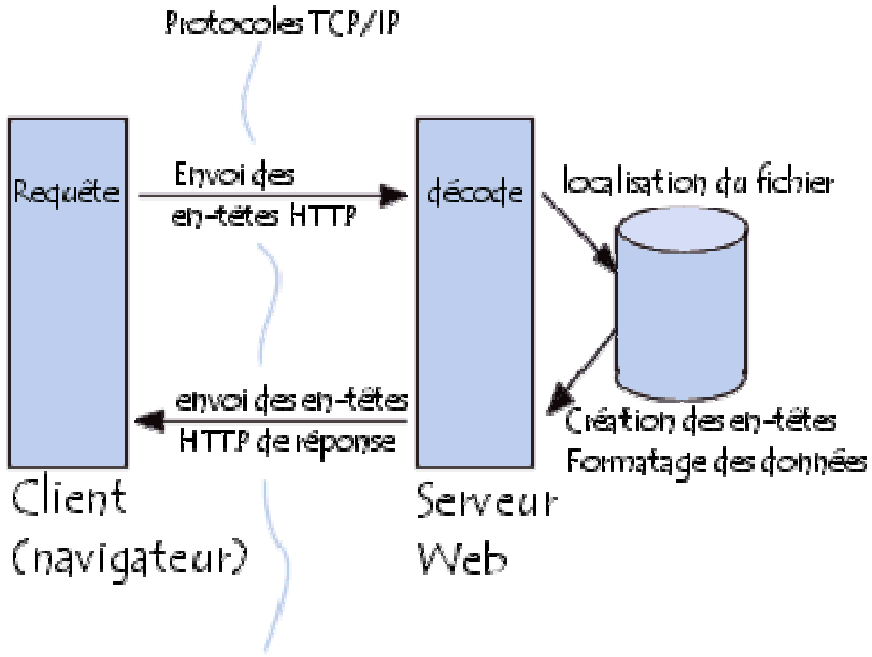
- **URL** : Uniform Resource Locator
- Identifie l'endroit où se trouve une ressource sur le Web.
- Dans le cas du Web, ressource = document ou fragment
- 

- Principaux protocoles utilisés dans les URL :

  http, https : deux protocoles du Web, sans et avec chiffrement et authentification ;
  ftp : protocole de transfert de fichier, parfois utilisé sur le Web pour le téléchargement de gros fichiers ;
  mailto : pseudo-protocole dénotant une adresse e-mail.
- Port par défaut : 80 pour HTTP, 443 pour HTTPS

Communication entre navigateur et serveur

La communication entre le navigateur et le serveur se fait en deux temps :



- Le navigateur effectue une **requête HTTP**
- Le serveur traite la requête puis envoie une **réponse HTTP**

Requête HTTP

Une **requête HTTP** est un ensemble de lignes envoyé au serveur par le navigateur. Elle comprend :

- **Une ligne de requête**: c'est une ligne précisant le type de document demandé, la méthode qui doit être appliquée, et la version du protocole utilisée.

  La ligne comprend trois éléments devant être séparés par un espace :

  - La méthode
  - L'URL
  - La version du protocole utilisé par le client (généralement *HTTP/1.0*)
- **Les champs d'en-tête de la requête**: il s'agit d'un ensemble de lignes facultatives permettant de donner des informations supplémentaires sur la requête et/ou le client (Navigateur, système d'exploitation, ...). Chacune de ces lignes est composée d'un nom qualifiant le type d'en-tête, suivi de deux points (:) et de la valeur de l'en-tête.
- **Le corps de la requête**: c'est un ensemble de lignes optionnelles devant être séparées des lignes précédentes par une ligne vide et permettant par exemple un envoi de données par une commande POST lors de l'envoi de données au serveur par un formulaire.

Une requête HTTP a donc la syntaxe suivante (<*crlf*> : retour chariot ou saut de ligne) :

METHODE URL VERSION<*crlf*>
EN-TETE : Valeur<*crlf*>
.
.
.
EN-TETE : Valeur<*crlf*>
Ligne vide<*crlf*>
CORPS DE LA REQUETE

Exemple :

GET http://www.yahoo.fr HTTP/1.0
Accept : text/html
If-Modified-Since : Saturday, 15-January-2000 14:37:11 GMT
User-Agent : Mozilla/4.0 (compatible; MSIE 5.0; Windows 95)

- ***Commandes ou méthodes :***

Commande	Description
GET	Requête de la ressource située à l'URL spécifiée
HEAD	Requête de l'en-tête de la ressource située à l'URL spécifiée
POST	Envoi de données au programme situé à l'URL spécifiée
PUT	Envoi de données à l'URL spécifiée
DELETE	Suppression de la ressource située à l'URL spécifiée

- ***En-têtes***

Nom de l'en-tête	Description
Accept	Type de contenu accepté par le browser (exemple *text/html*).
Accept-Charset	Jeu de caractères attendu par le browser
Accept-Encoding	Codage de données accepté par le browser
Accept-Language	Langage attendu par le browser (anglais par défaut)
Authorization	Identification du browser auprès du serveur
Content-Encoding	Type de codage du corps de la requête
Content-Language	Type de langage du corps de la requête
Content-Length	Longueur du corps de la requête
Content-Type	Type de contenu du corps de la requête (exemple *text/html*).
Date	Date de début de transfert des données
Forwarded	Utilisé par les machines intermédiaires entre le browser et le serveur
From	Permet de spécifier l'adresse e-mail du client
From	Permet de spécifier que le document doit être envoyé s'il a été modifié depuis une certaine date
Link	Relation entre deux URL
Orig-URL	URL d'origine de la requête

Referer	URL du lien à partir duquel la requête a été effectuée
User-Agent	Chaîne donnant des informations sur le client, comme le nom et la version du navigateur, du système d'exploitation

Réponse HTTP

Une **réponse HTTP** est un ensemble de lignes envoyées au navigateur par le serveur. Elle comprend :

- ***Une ligne de statut***: c'est une ligne précisant la version du protocole utilisé et l'état du traitement de la requête à l'aide d'un **code** et d'un **texte** explicatif.

  La ligne comprend trois éléments devant être séparés par un espace :

  - La version du protocole utilisé
  - Le code de statut
  - La signification du code

- ***Les champs d'en-tête de la réponse***: il s'agit d'un ensemble de lignes facultatives permettant de donner des informations supplémentaires sur la réponse et/ou le serveur.

  Chacune de ces lignes est composée d'un nom qualifiant le type d'en-tête, suivi de deux points (:) et de la valeur de l'en-tête.

- ***Le corps de la réponse***: il contient le document demandé.

Une réponse HTTP a donc la syntaxe suivante (<*crlf*> : retour chariot ou saut de ligne) :

VERSION-HTTP CODE EXPLICATION<*crlf*>
EN-TETE : Valeur<*crlf*>
.
.
.
EN-TETE : Valeur<*crlf*>
Ligne vide<*crlf*>
CORPS DE LA REPONSE

Exemple de réponse HTTP :

HTTP/1.0 200 OK
Date : Sat, 15 Jan 2000 14:37:12 GMT Server : Microsoft-IIS/2.0
Content-Type : text/HTML
Content-Length : 1245
Last-Modified : Fri, 14 Jan 2000 08:25:13 GMT

- ***En-têtes de réponse***

Nom de l'en-tête</souligne>	Description</souligne>
Content-Encoding	Type de codage du corps de la réponse
Content-Language	Type de langage du corps de la réponse
Content-Length	Longueur du corps de la réponse
Content-Type	Type de contenu du corps de la réponse (par exemple *text/html*).
Date	Date de début de transfert des données
Expires	Date limite de consommation des données
Forwarded	Utilisé par les machines intermédiaires entre le browser et le serveur
Location	Redirection vers une nouvelle URL associée au document
Server	Caractéristiques du serveur ayant envoyé la réponse

- ***Les codes de réponse***

  Ce sont les codes que vous voyez lorsque le navigateur n'arrive pas à vous fournir la page demandée.

  Le code de réponse est constitué de trois chiffres : le premier indique la classe de statut et les suivants la nature exacte de l'erreur.

Code	Message	Description
10x	**Message d'information**	**Ces codes ne sont pas utilisés dans la version 1.0 du protocole**
20x	**Réussite**	**Ces codes indiquent le bon déroulement de la transaction**
200	OK	La requête a été accomplie correctement
201	CREATED	Elle suit une commande POST, elle indique la réussite, le corps du reste du document est sensé indiquer l'URL à laquelle le document nouvellement créé devrait se trouver.
202	ACCEPTED	La requête a été acceptée, mais la procédure qui suit n'a pas été accomplie
203	PARTIAL INFORMATION	Lorsque ce code est reçu en réponse à une commande GET, cela indique que la réponse n'est pas complète.
204	NO RESPONSE	Le serveur a reçu la requête mais il n'y a pas d'information à renvoyer
205	RESET CONTENT	Le serveur indique au navigateur de supprimer le contenu des champs d'un formulaire
206	PARTIAL CONTENT	Il s'agit d'une réponse à une requête comportant l'en-tête *range*. Le serveur doit indiquer l'en-tête *content-Range*
30x	**Redirection**	**Ces codes indiquent que la ressource n'est plus à l'emplacement indiqué**
301	MOVED	Les données demandées ont été transférées à une nouvelle adresse

302	FOUND	Les données demandées sont à une nouvelle URL, mais ont cependant peut-être été déplacées depuis...
303	METHOD	Cela implique que le client doit essayer une nouvelle adresse, en essayant de préférence une autre méthode que GET
304	NOT MODIFIED	Si le client a effectué une commande GET conditionnelle (en demandant si le document a été modifié depuis la dernière fois) et que le document n'a pas été modifié il renvoie ce code.
40x	**Erreur due au client**	**Ces codes indiquent que la requête est incorrecte**
400	BAD REQUEST	La syntaxe de la requête est mal formulée ou est impossible à satisfaire
401	UNAUTHORIZED	Le paramètre du message donne les spécifications des formes d'autorisation acceptables. Le client doit reformuler sa requête avec les bonnes données d'autorisation
402	PAYMENT REQUIRED	Le client doit reformuler sa demande avec les bonnes données de paiement
403	FORBIDDEN	L'accès à la ressource est tout simplement interdit
404	NOT FOUND	Classique! Le serveur n'a rien trouvé à l'adresse spécifiée.
50x	**Erreur due au serveur**	**Ces codes indiquent qu'il y a eu une erreur interne du serveur**
500	INTERNAL ERROR	Le serveur a rencontré une condition inattendue qui l'a empêché de donner suite à la demande.
501	NOT IMPLEMENTED	Le serveur ne supporte pas le service demandé
502	BAD GATEWAY	Le serveur a reçu une réponse invalide de la part du serveur auquel il essayait d'accéder en agissant comme une passerelle ou un proxy
503	SERVICE UNAVAILABLE	Le serveur ne peut pas vous répondre à l'instant présent, car le trafic est trop dense (toutes les lignes de votre correspondant sont occupées veuillez rappeler ultérieurement)
504	GATEWAY TIMEOUT	La réponse du serveur a été trop longue vis-à-vis du temps pendant lequel la passerelle était préparée à l'attendre (le temps qui vous était imparti est maintenant écoulé...)

Chapitre 3 :Langage HTML

Généralités

- HTML (pour HyperText MarkupLanguage) est un langage de description de documents *hypermédia.*
- Un document hypermédia est un document qui est à la fois *multimédia et hypertexte.*
- Un document ***multimédia*** est un document constitué d'éléments de plusieurs natures : texte, images, son, vidéo, code exécutable, etc.
- Un document***hypertexte*** est un document dans lequel certains éléments sont rendus actifs (on parle alors de *liens*). En activant un lien (généralement en cliquant dessus), le lecteur peut accéder à un autre document. Ce dernier peut être un document hypertexte ou non.
- Le langage HTML est dérivé du langage SGML (Standard GeneralizedMarkupLanguage).
- En utilisant SGML, l'auteur *balise* et le contenu de son document sont des marques spéciales, dites *balises.*
- Certaines balises sont utilisées pour décrire la structure du document (titre, section, sous-section, etc.).
- D'autres pour décrire la présentation du document (caractères en gras, en italique, etc.).
- La troisième famille de balises sert à décrire lasémantique attribuée aux composants (ex. spécifierqu'une chaîne de caractères représente une adresse, le nom d'une personne, un numéro de téléphone, etc.).

Balises HTML

- Les balises servent à donner des instructions :
  mise en forme du texte.
  insertion d'objets multimédias comme les images
  insertion de liens hypertextes
  insertion d'éléments actifs : javascript, applets
  java, animations flash, ...
- Une balise se présente sous la forme d'une lettre ou d'un mot (son nom) entourés par les symboles **<** et **>**.
- ***Exemple***:

  ** ** commence un paragraphe;

  ** **termine le paragraphe.

  ****indique que le texte doit être écrit en gras;

  ****met fin à la commande.

- La syntaxe d'une balise est le suivant :

  <nom de la balise>

- Par exemple la balise qui marque le début d'un document HTML est la balise : <HTML>.
- Pour marquer la fin du champ d'action d'une balise on utilise une balise de fermeture dont la syntaxe est:

</nom de la balise>

- Ainsi pour marquer la fin d'un document HTML on utilise la balise </html>.
- Il existe des balises qui n'ont pas la fin.

Exemple :

 Simple saut de ligne.

<HR> Dessine une ligne.

Premier fichier HTML

```
<HTML>
<HEAD>
<TITLE>Ma première page</TITLE>
< /HEAD>
<BODY>
Salut! Tout le monde.
</BODY>
< /HTML>
```

- les balises <html> et </html> : elles entourent ledocument HTML.
- Le document est constitué de deux parties : l'en-tête et le corps.
- les balises <head> et </head> délimitent l'en-tête.
- les balises <body> et </body> délimitent le corps.

En-tête <HEAD>

L'entête est destiné à recevoir certains renseignements généraux sur le fichier : auteur, mots-clés, titre de la page, …

Les balises ***<title>*** et ***</title>*** servent à définir le titre utilisé par le navigateur pour l'affichage de la barre de titre de la fenêtre ou pour l'insertion dans les favoris.

Corps <BODY>

Le corps du fichier contient le texte qui constitue la partie visible du document et balisé par : <body> et </body>.

Voir le résultat

Enregistrer le fichier texte sous le nom *exemple1.html.* Double cliquer sur *exemple1.html* pour le voir apparaître dans un navigateur web.

Utilisation des couleurs

- Certaines balises peuvent être accompagnées d'attributs : la balise <body> permet de définir la couleur du fond de la page ainsi que celle du texte.
- Les couleurs sont définies par une suite de 3 nombres hexadécimaux représentant les valeurs du rouge, du vert et du bleu.
- Pour obtenir un texte écrit en blanc sur fond bleu, on complètera la balise <body> de la façon suivante :

  <body bgcolor="#0000FF" text="#FFFFFF">

- Il estaussi possible d'utilisercertainsnoms de couleurs :White, Black, Red, Green, Cyan, Magenta, Gray, Silver, Gold, Orange, Yellow et Blue.

  Ainsi, <body bgcolor="Gold" text= "blue"> donnera un texte bleu sur fond doré.

  La balise ***<body>*** accepte les principaux attributs suivants :

  Background **:** désigne la source d'une image de fond.

  Bgcolor**:** désigne la couleur de fond de la page.

  Link **:** désigne la couleur d'un lien non encore visité.

  Vlink**:** désigne la couleur d'un lien déjà visité.

  Alink**:** désigne la couleur d'un lien sélectionné.

  Text**:** désigne la couleur du texte.

Exemple:

<body background="butterfly.gif" bgcolor="#FFFFFF" text="#000000">

Cette balise spécifie que le fond de la page est rempli par l'image contenu dans le fichier *butterfly.gif*, le texte de la page est en noir et la couleur du fond est en blanc.

Organisation du texte

- Un texte est organisé en sections.
- Chaque section peut contenir de sous-sections et/ou des paragraphes.
- Un paragraphe est composé de plusieurs lignes.
- HTML fournit un ensemble de balises pour gérer une telle décomposition de textes.
- Les balises <Hx> et </Hx> où x = [1, 6] sont utilisées pour marquer les sections.
- La balise <H1> pour marquer les sections principales <H2> pour les sous-sections et <H3> pour les sous sous-sections et ainsi de suite.
- Chacune des balises <Hx> accepte un attribut d'alignement nommé***align***qui peut prendre les valeurs : left, right et center.

Exemple :

<H1 align="center"> Titre Principal </H1>

<H2 align="left "> Sous–titre </H2>

<H3 align="left "> Sous sous–titre </H3>

<H4 align="left "> 3eme sous–titre </H4>

<H5 align="left "> Avant dernier sous–titre </H5>

<H6 aligne="left " > Dernier sous-titre </H6>

Paragraphe :

- Le début d'un paragraphe est marqué par la balise <P> qui accepte aussi un attribut *align.*
- La fin d'une ligne (ou pour commencer une nouvelle ligne) est marquée par la balise
.
- La fin du paragraphe est marqué par la balise </P>

Exemple :

```
<p align="center">
Ceci est une <br>
Page web
<br>
C'est une autre ligne
</p>
```

Centré, Gras, Italique, etc.

```
<BODY>
<B>texte en gras</B><BR>
<I>texte en italique</I><BR>
<U>texte souligné</U><BR>
<CENTER>texte centre</CENTER><BR>
<B><FONT color="red" size=2>texte en rouge, gras de taille 2 </FONT></B>< /BODY>
```

- La balise <B> pour mettre le texte en gras.
- La balise <I> pour mettre le texte en italique.
- La balise <U> pour souligner un texte.
- La balise pour centrer le texte.
- La balise <FONT> pour changer la couleur et la taille du texte.

Remarques Importantes

- Attention à bien **refermer vos balises** (avec la balise de fin) sinon la suite du texte aura également les propriétés de la balise non refermée.
- Si vous voulez cumulez des balises, par exemple : pour mettre un texte en gras, en italique et centré, il faut bien faire **attention à l'ordre des balises** pour les refermer ! Voici ce qu'il ne faut pas faire :

```
<B><I><CENTER>mon texte</I></B</CENTER>
```

- **Il faut les refermer dans l'ordre inverse de l'ouverture**

```
<B><I><CENTER>mon texte</CENTER></I></B>
```

Les Tableaux

- Les balises principales sont <TABLE>...</TABLE>.
  Elles entourent le tableau.
- Entre ces balises seront définies les cellules, ligne par ligne.
- Les balises <TR>...</TR> entourent les cellules quiforment une ligne du tableau.
- Les balises <TD>...</TD> entourent une cellule.

Exemple :

```
<table border=1>
<tr>
        <td>Cellule Ligne1 Colonne 1</td>
        <td>Cellule Ligne1 Colonne 2</td>
        <td>Cellule Ligne1 Colonne 3</td>
</tr>
<tr>
        <td>Cellule Ligne2 Colonne 1</td>
        <td>Cellule Ligne2 Colonne 2</td>
        <td>Cellule Ligne2 Colonne 3</td>
</tr></table>
```

Attributs de la balise <TABLE>

La balise <TABLE> peut contenir un certain nombre d'attribut qui définissent son positionnement, sa taille et son aspect.

- *BORDER* :permet de fixer la taille en pixels des lignes qui délimitent les cellules du tableau. Si cette taille est 0, ces lignes ne sont pas affichées.
- *WIDTH* : permet de fixer la largeur du tableau en pixels ou en pourcentage de la largeur disponible.

- *ALIGN* : permet de définir l'alignement du tableau dans la largeur de l'espace disponible. Les valeurs possibles sont : LEFT, CENTER et RIGHT.
- *CELLSPACING* : permet de définir en pixels l'espace de séparation des cellules.
- *CELLPADDING* : permet de définir en pixels l'espace séparant le bord des cellules de leur contenu.

Attributs de la balise <TD>

La balise <TD> peut contenir un certain nombre d'attribut qui définissent son comportement d'une cellule.

Cellule Ligne1 Colonne 1	Cellule Ligne1 Colonne 2	Cellule Ligne1 Colonne 3
Cellule Ligne2 Colonne 1	Cellule Ligne2 Colonne 2	Cellule Ligne2 Colonne 3

- *WIDTH* : permet de fixer la largeur de la cellule en pixels ou en pourcentage de la largeur du tableau.
- *BGCOLOR* :permet de définir la couleur de fond de la cellule.

- *VALIGN* : permet de définir l'alignement vertical dans la cellule. Les valeurs possibles sont : TOP, MIDDLE et BOTTOM.
- *COLSPAN* : permet de définir le nombre de colonnes de tableau que devra occuper la cellule.
- *ROWSPAN* : permet de définir le nombre de lignes de tableau que devra occuper la cellule.

Insérer une image dans un document html

- On peut inclure une image dans le corps du document en utilisant la balise **IMG.**
  Exemple :

```
<IMG SRC="nom_fichier">
```

- Cette balise admet un certain nombre d'attributs :

  - ***ALT*** : permet de spécifier un petit texte associé à l'image; ce texte sert au remplacement tant que l'image n'est pas affichée, il apparaît aussi lorsqu'on laisse la souris au dessus de l'image.

    Exemple :

```
<IMG SRC="nom_fichier" ALT="Mon image">
```

  - ***BORDER :*** permet d'encadrer l'image ou de supprimer le cadre qui est mis automatiquement lorsque l'image joue le rôle de lien hypertexte.

  - ***WIDTH et HEIGHT*** : Permet de spécifier les dimensions de l'image en pixels. Cela permet des agrandissements, réductions ou déformations.
    Il est recommandé de spécifier les dimensions de l'image même si on ne les modifie pas, cela accélère le téléchargement.
    Le navigateur connaît immédiatement la place à réserver.

Cela lui permet de continuer à afficher le texte avant de s'occuper des images.

Exemple

```
<HTML>
<HEAD>
<TITLE>Exemple 4 </TITLE>
</HEAD>
<BODY bgcolor="#FFFFFF" text="#000000">
<Table border=1 align="left">
<TR>
<td colspan=2><B><I><FONT color="Cyan" size=4>Insertion d'une image
</FONT></I></B></td>
</TR>
<TR>
<td> L'image de droite est un papillon </td>
<td><IMG SRC="butterfly.gif" ALT="un beau papillon"></td>
</TR>
< /Table>
</BODY>
</HTML>
```

Séparateur :

<HR> : pour une ligne horizontale de séparation.

WIDTH : fait varier la largeur de la ligne en % soit en pixel (valeur par défaut 100%)

SIZE : fait varier l'épaisseur de la ligne en pixel (valeur par défaut 1)

ALIGN : fait un alignement de la ligne suivant les 3 possibilités :

- **CENTER :** par rapport au centre de la fenêtre,
- **LEFT :** par rapport à la gauche de la fenêtre,
- **RIGHT :** par rapport à la droite de la fenêtre.

NOSHADE : Lorsqu'il est présent dans le marqueur <HR> l'effet est une ligne pleine sans ombrage

Les balises <sub>, <sup>& :

- ***Le texte en exposant et en indice :***

  <**sub**> texte </**sub**> : texte en indice.

  <**sup**> texte </**sup**> : texte en exposant.

- ***Alignement du texte :***

  <**div**align=center> texte</**div**> : aligne le texte au centre.

  <**div**align=left>texte</**div**> : aligne le texte à gauche.

  <**div**align=right>texte</**div**> : aligne le texte à droite.

Les Listes :

- ***Listes numérotées ou Orderedlists :***

  <**ol**> texte </**ol**> : afficher le texte sous forme d'une liste ordonnée.

  <**li**> : voici un élément de la liste.

  Les options suivantes sont possibles :

  - Type=1 : pour une liste numérotée 1, 2, 3,…
  - Type =A : pour un repérage type A, B, C, …
  - Type =a : pour un repérage type a, b, c, …
  - Type =I : pour une liste numérotée I, II, III, …
  - Type =i : pour une liste numérotée i, ii, iii, …

- ***Listes non numérotées ou Unorderedlists***

<**ul**> texte </**ul**> : afficher le texte sous forme d'une liste non-ordonnée.

<**li**> : voici un élément de la liste.

Les options suivantes sont possibles :

- Type=disc : pour une liste non numérotée ●.
- Type =circle : pour un repérage type ○.
- Type =square : pour un repérage type □.

- ***Listes descriptives ou Definitionlists :***

<DL>texte</DL> : délimite une zone de liste de définition.

<DT> : introduit un nouveau terme de définition.

<DD> : introduit une description du terme de définition.

- ***Les listes peuvent s'imbriquer***

```
<HTML><HEAD>
<TITLE>les listes peuvent s'emboîter </TITLE></HEAD>
<BODY><OL><LI> BMW
        <UL><LI>série 3
            <LI> série 5 </UL>
    <LI> MERCEDES
        <DL><DT>classe  C< DD>180,  200,  220.
            <DT>  classe  E <DD>250, 300
        </DL>
    <LI> FIN </OL></BODY></HTML>
```

TEXTE PRÉ-FORMATÉ :

<PRE> texte </PRE> : pour inclure un texte pré-formaté.

Les espacements, tabulations et retour chariot gardent leur sens.

Exemple :

```
<HTML>
<HEAD><TITLE>textepré-formaté</TITLE></HEAD>
<BODY>
enseignements  :<BR><BR>
<PRE>
   MATIERES                    PROFESSEUR                          NB  d'heures
        ---------------------------------------------------------------
   Visual  Basic               B.  Mahric                          40
   Réseaux                     E.  Brassart                        32
   C++                         L.Delahoche                         32
 </PRE>
</BODY>
</HTML>
```

Les liens :

Html vous permet de vous transporter vers :

- Un autre endroit du document.
- Un autre fichier html situé sur votre ordinateur.
- Un autre ordinateur situé sur le web.

Syntaxe : <A HREF= ’’ document’’>texte </A>

HREF : hypertexte référence

Document : le document vers le quel on pointe.

Texte : qui sera affiché pour représenter le lien.

- *Lien vers la page locale fichier.html :*

<A HREF=’’fich2.html’’>aller au document 2</A>

<A HREF=’’fich1.html’’>retour au document 1</A>

- *Lien vers vers une adresse Email :*

  <A HREF=’’mailto :…’’>envoyer moi un message</A>

- *Lien vers vers une page web :*

  <A HREF=’’http://…’’>texte</A>

- *Lien pour le téléchargement d’un fichier :*

  <A HREF=’’ftp://…’’>texte</A>

- *Lien sur image :*

  <A HREF=’’fichier.html’’><IMG SRC=’’image.gif’’></A>

Les ancres :

Des liens peuvent aussi pointer vers un endroit précis du même document ou d'un autre fichier. C'est ce qu'on appelle les ancres, ancrages ou pointeurs [Anchor].

- Point d'ancrage : <A NAME="***">...</A> : Ceci est une cible.
- Lien vers une ancre dans la même page :

<A HREF="#*">...</A> : Lien vers la cible *** dans la même page.**

- Lien vers une ancre dans une autre page :

<A HREF="URL#*">...</A> : Lien vers la cible *** dans une autre page.**

Exemple :

```
<HTML><HEAD>
        <TITLE>voituresallemenades</TITLE></HEAD>
        <BODY><H1>voitures  allemandes</H1>
        <A  HREF="#BMW"><H2>BMW</H2></A>
        <A HREF="#MERCEDES"><H2>MERCEDES</H2></A>
        <A HREF="#PORSCHE"><H2>PORSCHE</H2></A><BR><BR>
        <A NAME="MERCEDES"></A> MERCEDES
        C<BR>
        MERCEDES  E<BR>
```

<A NAME="PORSCHE"></A> PORSCHE
956

PORSCHE 911
  PORSCHE 914
 PORSCHE 924

 PORSCHE 944

<A NAME="BMW"></A> BMW série 3
 BMW série 5

 BMW série 6
 BMW série 7
 BMW série 8

</BODY></HTML>

Les frames

Diviser une fenêtre en cadres ou plusieurs zones appelées **frames**, les balises sont peu nombreuses :

<FRAMESET> : début de zone avec des fenêtres.

</FRAMESET> : fin de zone avec des fenêtres.

- ***Frames en lignes :***

  <FRAMESET ROWS= ''n,n%,*,…''> : divise la zone en plusieurs fenêtres horizontales.

  n = hauteur en nombre pixels,

  n% = hauteur en pourcentage de l'écran,

  *= hauteur restante

- ***Frames en colonnes :***

  <FRAMESET COLS= ''n,n%,*,…''> : divise la zone en plusieurs fenêtres verticales.

  n = hauteur en nombre pixels

  n% = hauteur en pourcentage de l'écran.

  *= hauteur restante.

- ***Remarque :***

  <FRAMESET>…< /FRAMESET>remplace<body>…</body>.

  BORDER=0 : supprimer les bordures dans Netscape.

  FRAMEBORDER=NO : supprimer les bordures dans Explorer.

  FRAMESPACING=0 : enlever l'espace entre les cadres.

Exemple : on peut mélanger les deux types de frames:

```
<html><head>
<title> frames verticales et horizontales</title>
</head>
<frameset rows=''30%,70%">
        <frame>
        <frameset cols="35%, 65%">
            <frame>
            <frame>
        </frameset>
</frameset>
</html>
```

Contenu des frames :

<FRAME SRC= ''url ou adresse du document à afficher dans la fenêtre''>.

SCROLLING =''yes/no/auto '' : la fenêtre doit ou non posséder une barre de défilement.

NAME= ''NOM'' : le nom de la fenêtre de telle sorte que cette frame puisse être utilisée comme cible d'un lien hypertexte.

Lien avec des frames

<A HREF=''url ou adresse'' TARGET=''…''>lien</A>

TARGET = ''fenêtre1'' : afficher dans la fenêtre ''fenêtre1''.

TARGET = ''_self'' : afficher dans la même fenêtre.

TARGET = ''_blank'' : afficher dans une nouvelle fenêtre.

TARGET = ''_top'' : afficher sur toute la fenêtre du browser.

Formulaires

- ***La commande FORM***

Des formulaires peuvent être préparés afin de saisir des données et les traiter au niveau du serveur.

Pour rédiger un questionnaire, il faut:

- Établir une zone d'édition (appelée FORM) en utilisant les commandes <FORM></FORM>.
- Définir la méthode à employer pour transmettre au serveur l'information recueillie dans leschamps du formulaire.
- Identifier l'emplacement et le nom du programme qui devra traiter l'information recueillie.
- Fournir, s'il y a lieu, les arguments au programme de traitement des données.

Toute cette information se retrouve dans la commande suivante:

<FORM METHOD="POST" ACTION="/cgi-bin/questionnaire.cmd?xxx">

La méthode utilisée est ***POST***, le programme de traitement se nomme questionnaire.cmd et se retrouve dans le dossier cgi-bin, un seul argument est fourni au programme soit xxx.

Il est à noter que le programme de traitement des données (questionnaire.cmd en l'occurrence) doit être fourni par l'administrateur du serveur et créé en fonction de l'application à supporter.

- ***Les commandes INPUT***

Syntaxe : ***INPUT="TEXT"***

Parmi les choix disponibles en HTML, un des types d'entrée de données est le champ ***input type="text"***. Dans ce cas, il faut inscrire le type de champ,

le nom du champ et ses dimensions à l'écran.

Ainsi, dans la question ci-dessous, le code à utiliser pour entrer le nom de la personne est :

NOM: <input type="text" name="name" size=30>.

Le type ***text*** est un champ ou l'usager entre de l'information sur son clavier, dans une zone définie à l'écran par la commande ***size***.

Ainsi, une commande ***size=30*** est un champ d'une longueur de 30 espacements de largeur. Pour un champ plus long, entrez une valeur de 50 ou 70, selon l'espace requis pour couvrir toutes les possibilités.

La commande ***name="name"*** précise que l'on désire enregistrer le contenu du champ dans la rubrique ***"name"***.

- ***La commande INPUT="RADIO"***

  input type=''radio'' : permet d'afficher une série de boutons radio comme choix de réponses.

  <input type="radio" name="info" value="OUI">OUI

  <input type="radio" name="info" value="NON">NON

- ***La commande INPUT=CHECKBOX***

  input Type=checkbox : permet d'afficher une liste ou plusieurs choix sont possibles en même temps.

Texte **<input name="*nom_du_champ*" TYPE=checkbox VALUE="*texte*"> **

ou

**<input name="*nom_du_champ*" TYPE=checkbox VALUE="*texte*">*Texte* **

- ***La commande SELECT NAME et OPTION***

  selectname et ***Option*** : permet d'afficher une liste ou un seul choix est possible et qui s'affichent sous la forme d'un menu ***"pop-up"***.

<select name><option selected>*option1*<option>*option2*<option>*option3*</select>

- ***La commande TEXTAREA.***

<TEXTAREA name="nom_du_champ?" rows=n cols=m></TEXTAREA>

Dans ce type, on spécifie d'abord le type soit textarea, ensuite le nom de la rubrique soit name=" *nom_du_champ* " puis les paramètres d'affichage de la zone de dialogue en rangées (n) et en colonnes (m).

Remarque :

Les formulaires doivent être complétés avant fermeture par une commande permettant d'envoyer le contenu des champs remplis au serveur HTTP. On ajoute une deuxième commande qui permet à l'usager de reprendre le questionnaire s'il s'est trompé.

<input type="submit" value="Soumettre">

<input type="reset" value="effacer et recommencer">

Liste des principales balises Html

Mise en forme des caractères

<B>...</B>	Texte en gras
<BIG>...</BIG>	Agrandissement de la taille des caractères
<BLINK>...</BLINK>	Texte clignotant (Netscape seul)
<EM>...</EM>	Texte en italique
<FONT color="#XXXXXX">...</FONT>	Texte en couleur où XXXXXX est une valeur hexadécimale
<FONT size=X>...</FONT>	Taille des caractères où X est une valeur de 1 à 7
<I>...</I>	Texte en italique
<NOBR>...</NOBR>	Empêche les ruptures automatiques de ligne des navigateurs
<PRE>...</PRE>	Texte préformaté, soit avec affichage de tous les espaces et sautsde ligne
<SMALL>...</SMALL>	Réduction de la taille des caractères
<STRONG>...</STRONG>	Mise en gras du texte
_{...}	Texte en indice
^{...}	Texte en exposant
<U>...</U>	Texte souligné

Mise en forme du texte

<!--...-->	Commentaire ignoré par le navigateur
 	A la ligne
<BLOCKQUOTE>...</BLOCKQUOTE>	Citation (introduit un retrait du texte)
<CENTER>...</CENTER>	Centre tout élément compris dans le tag
<DIV align=center> ...</DIV>	Centre l'élément encadré par le tag
<DIV align=left> ...</DIV>	Aligne l'élément à gauche
<DIV align=right> ...</DIV>	Aligne l'élément à droite
<Hx>...</Hx>	Titre où x a une valeur de 1 à 7
<Hxalign=center>...</Hx>	Titre centré
<Hxalign=left>...</Hx>	Titre aligné à gauche
<Hxalign=right>...</Hx>	Titre aligné à droite
<P>...</P>	Nouveau paragraphe
<P align=center>...</P>	Paragraphe centré
<P align=left>...</P>	Paragraphe aligné à gauche
<P align=right>...</P>	Paragraphe aligné à droite

Listes

<UL>	Liste non numérotée (dite à puces)
<LI>	Elément de liste
</UL>	
<OL>	Liste numérotée
<LI>	Elément de liste
</OL>	
<DL>	Liste de glossaire
<DT>...</DT>	Terme de glossaire (sans retrait)

<DD>...</DD>	Explication du terme (avec retrait)
</DL>	Ligne de séparation
<HR>	Trait horizontal (centré par défaut)
<HR width="x%">	Largeur du trait en %
<HR width=x>	Largeur du trait en pixels
<HR size=x>	Hauteur du trait en pixels
<HR align=center>	Trait centré (défaut)
<HR align=left>	Trait aligné à gauche
<HR align=right>	Trait aligné à droite
<HR noshade>	Trait sans effet d'ombrage

Hyperliens

<A href="http://...">...</A>	Lien vers une page Web
<A href="mailto:...">...</A>	Lien vers une adresse Email
<A href="fichier.htm">...</A>	Lien vers la page locale fichier.htm située dans le même dossier
<A name="xyz">...</A>	Définition d'une ancre
<A href="xyz">...</A>	Lien vers une ancre
<A href="fichier#xyz">...</A>	

Images

<IMG src="xyz.gif">	Insertion d'une image au format Gif ou Jpg
<IMG src="xyz.pjg>	(voir liens pour l'adressage)
<IMG ... width=x height=y>	Mise à l'échelle de l'image en pixels
< IMG ... border=x>	Définition de la bordure d'une image avec lien
<IMG ... alt="votre texte">	Texte alternatif lorsque l'image n'est pas affichée
<IMG ... align=bottom>	Aligne l'image en bas
<IMG ... align=middle>	Aligne l'image au milieu
<IMG ... align=top>	Aligne l'image en haut
<IMG ... align=left>	Aligne l'image à gaughe
<IMG ... align=right>	Aligne l'image à droite
<IMG ... hspace=x>	Espacement horizontal entre l'image et le texte
<IMG ... vspace=y>	Espacement vertical entre l'image et le texte

Tableau

<TABLE>...</TABLE>	Définition d'un tableau
<TABLE width="x%">	Largeur du tableau en %
< TABLE width=x>	Largeur du tableau en pixels
<TABLE border=x>	Largeur de la bordure
<TABLE cellpadding=x>	Espace entre la bordure et le texte
<TABLE cellspacing=x>	Epaisseur du trait entre les cellules
<TR>...</TR>	Ligne dutab
<TD>...</TD>	Cellule du tableau
<TD bgcolor="#XXXXXX">	Couleur d'une cellule de tableau
<TD width="x%">	Largeur de colonne en %
<TD width=x>	Largeur de colonne en pixels
<TD align=center>	Texte dans la cellule centré
<TD align=left>	Texte dans la cellule aligné à gauche
<TD align=right>	Texte dans la cellule aligné à droite
<TD valign=bottom>	Alignement vers le bas du contenu d'une cellule
<TD valign=middle>	Centrage vertical du contenu d'une cellule
<TD valign=top>	Alignement vers le haut du contenu d'une cellule
<TD colspan=x>	Nombre de cellules à fusionner horizontalement
<TD rowspan=x>	Nombre de cellules à fusionner verticalement

Frames

<FRAMESET>...</FRAMESET>	Définit une structure de frames (remplace alors le tag BODY)
<FRAMESET rows="x%,y%,...">	Division horizontale de la fenêtre en %
<FRAMESET cols="x%,y%,...">	Division verticale de la fenêtre en %
<FRAME src="fichier.htm">	Fichier affiché dans une fenêtre de frames
<NOFRAMES>...</NOFRAMES>	Contenu pour les browsers non prévus pour les frames

Fichier Html

<HTML>...</HTML>	Début et fin de fichier Html
<HEAD>...</HEAD>	Zone d'en-tête d'un fichier Html
<TITLE>...</TITLE>	Titre affiché par le browser (élément de HEAD)
<BODY>...</BODY>	Début et fin du corps du fichier Html
<BODY bgcolor="#XXXXXX">	Couleur d'arrière-plan (en hexadécimal)
<BODY background="xyz.gif">	Image d'arrière-plan

Les couleurs plus utilisées :

La valeur de couleur est composée de trois nombres hexadécimaux accolés (codés de 00 à FF) représentant le mélange des trois couleurs primaires RGB (Red, Green, Blue). Le nombre obtenu est précédé d'un dièse (#). Le blanc à pour valeur # FFFFFF ; le noir à pour valeur #000000. Toutes les autres couleurs sont obtenues par des dosages précis dans chacune des composantes RVB.
Exemple de couleur :

avec c = #000000	pour noir
avec c = #FFFFFF	pour blanc
avec c = #C0C0C0	pour gris clair
avec c = #80FF80	pour vert clair
avec c = #FFFFE8	pour jaune clair
avec c = #80FFFF	pour bleu clair
avec c = #FF8080	pour rose
avec c = #FF80C0	pour mauve
avec c = #800000	pour marron
avec c= #000080	pour bleu
avec c = #008040	pour vert
avec c = #FF0000	pour rouge
avec c= #FF8000	pour jaune
avec c = #800080	pour violet

Chapitre 4 :Feuilles de style css

Présentation

Des feuilles de style introduit en 1997 par Microsoft avec son Internet Explorer 3.0. Elles n'ont connu qu'un intérêt limité du fait que celles-ci n'étaient pas reconnues par Netscape Navigator 3.0.

Depuis les choses ont bien changé. D'abord les browsers 4.0 de Microsoft et de Netscape reconnaissent tous deux les feuilles de style et surtout, la norme Html 4.0 en a repris le concept (CSS version 1) et le recommande d'ailleurs vivement aux "Web designers".

Limites de HTML

- HTML a des limitations importantes :
  - Mise en forme de la présentation :

    Mise en page difficile.

    Limitation des possibilités.
- Maintenance d'un site : Fastidieuse, Prend beaucoup de temps
- Ceci vient du mélange de données de différentes natures :
  - les balises servent à structurer le document (paragraphe, etc.),
  - les balises servent comme des objets (image, etc.),
  - les balises sont utilisées pour la mise en forme. (fonte, etc.);

Nécessaire : séparation de la présentation et des données.

Considérons une partie d'un document HTML

```
<h1 align="center">
        <font color="#0000FF" face="arial">
                <b>Ceci est un contenu bleu, en police arial, gras, centré</b>.
        </font>
</h1>
<h1 align="center">
        <font color="#0000FF" face="arial">
                <b>Ceci est deuxième contenu bleu, en police arial, gras, centré</b>.
        </font>
</h1>
<h1 align="center">
        <font color="#0000FF" face="arial">
                <b>Ceci est un troisième contenu bleu, en police arial, gras, centré</b>.
        </font>
</h1>
```

Beaucoup d'information redondante, ce qui provoque :

- Des difficultés pour les mises à jours (il faut changer partout)
- Et surtout des risques d'erreurs (e.g oublier de changer une...)

Qu'est ce que les CSS?

C'est une solution au problème. CSS : Cascading Style Sheet, Feuilles de style en cascade.

- **Objectif** : rendre cohérents et homogènes les sites Web et faciliter leur maintenance.
  - Séparation entre contenu et propriétés visuelles
  - Un hyper Document = un fichier HTML + une feuille de style.
  - Toute modification d'une feuille de style est répercutée sur l'ensemble des pages où elle s'applique.

• **Comment** ?

Un ensemble de propriétés de mise en forme. + Un mécanisme qui permet de spécifier à quelle partie du document il faut les appliquer.

Où placer vos règles de style

- Dans le document Html lui-même :
  - Avec l'attribut style (toutes balises) : local à la balise
    * Ce paragraphe est bleu et italique *
  - Avec la balise HTML **<style>** dans le "header " : local au document

```
<html>
<head>
        <title>Exemple de feuille de style</title>
        <style type="text/css">
                <!- -ici- - >
        </style>
</head>
```

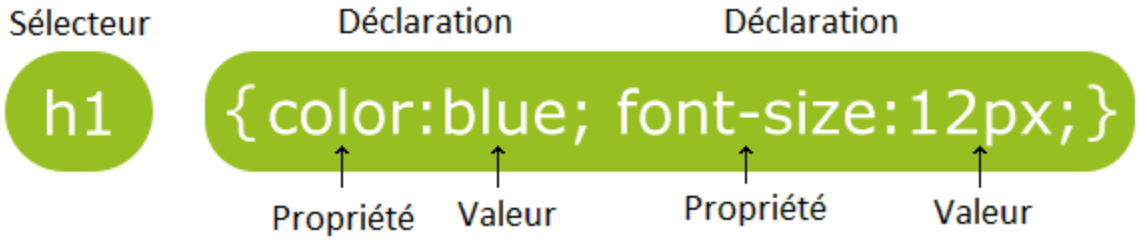
Il faudra renouveler l'opération pour chaque page !

- Dans un fichier séparé, nommé en **.css** :
  Il suffit d'indiquer le chemin dans l'entête du document Html avec une balise de la forme :
  <LINK REL="stylesheet" TYPE="text/css" HREF="style.css">.

Définition d'un style

La définition de base d'un style est simple :

H1
{
fontsize: large;
color: red;
…
}
	Nom de la balise considérée
	{
	Attribut1 : valeur1;
	Attribut2 : valeur2 ;
	...
	}



Les règles permettant de gérer l'aspect des éléments d'une page sont constituées de trois parties :
Le sélecteur est l'élément affecté par cette règle
La propriété est suivie de deux points **:**
La valeur est suivie d'un point-virgule **;**
Les **propriétés** et **valeurs** se placent entre accolades - entre **{** et**}** – elles forment un ensemble également appelé la **déclaration**.

- L'espace entre propriétés de style et valeur n'est pas obligatoire mais aide fortement à la lisibilité du code source.
- Pour la lisibilité toujours, vous pouvez écrire vos styles sur plusieurs lignes ;
  Exemple :
  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.
  Exemple :
  H3 {font-family: Arial, Helvetica, sans-serif}
- On peut attribuer un même style à plusieurs balises (séparées par des virgules).
  Exemple :
  H1, H2, H3 {font-family: Arial; font-style: italic}

Sélecteur Class

Une classe est un ensemble de règles auquel on a attribué un ***nom de classe*** qui permet de reproduire cet ensemble de règles sur n'importe quel élément d'une page.

Un nom significatif permettra d'y retrouver plus aisément.

Dans la feuille de style ; ***le nom de la classe doit obligatoirement être précédé d'un Point***:

Exemple :

```
.nom_de_classe {
                font-size: 10pt;
                padding: 3px;
                margin: 0px 3px 0 3px;
                }
```

Appel et Affectation de cette classe dans un fichier Html sans le ***Point*** :

Exemple :

* Ici Nous appliquons notre classe à cet élément paragraphe. Mais nous pouvons de la même façon l'affecter à d'autres éléments de la même page, et cela autant de fois que vous le souhaitez *	*<h1 class="nom_de_classe">* Ici Notre Titre h1 aura le même aspect que le paragraphe ci-contre *</h1>*

Il est possible d'appliquer le même style à toutes les ***instances d'une balise*** Html utilisées,

Exemple :

p { font-size: 10pt; padding: 3px; margin: 0px 3px 0 3px; }	* Tous les paragraphes ne comportant pas de classe spécifique (comme ci-dessus) auront l'aspect défini ci-contre *
*h1 { color: #0000FF; font-weight: **bold**; }*	* Tous les Titres h1 seront en bleu et en gras Exceptés ceux comportant une classe spécifique. Comme dans l'exemple précédent. *

Dans cet exemple, nous ***n'utilisons pas de point*** devant le sélecteur ***(p, h1)*** car nous utilisons directement le *Nom de la balise HTML* et ***non l'attributClass.***
Il est possible d'appliquer des styles différents à un même élément, dans ce cas le ***nom de la classe sera précédé du nom du sélecteur*** :

Exemple :

b.noir{ color:#000000; font-size:8pt; }	*b.rouge { color: #FF0000; font-size: 9pt; }*	*b.vert { color: #00FF00; font-size: 10pt; }*
Noir	*Rouge*	*Vert*

Pseudo-Classe

Les pseudo-classes sont en fait des classes prédéfinies, elles permettent d'affecter un style sur un élément en fonction de son état, une réaction à un événement.

Usage :

- Couleur des liens
- Apparence de la première ligne d'un paragraphe
- Apparence de la lettrine (première lettre d'un paragraphe)

Liés aux liens (balise <a>) :

a:active	lien actuellement sélectionné
a:link	lien non encore visité
a:visited	lien déjà visité
a:hover	lien survolé par la souris

Utilisation :

a:link {propriété 1; propriété 2; …}
a:visited {propriété 1; propriété 2; …}

Les autres pseudo-éléments :

:first-letter : Apparence de la première lettre d'un paragraphe (lettrine)
:first-line : Apparence de la première ligne d'un paragraphe

Utilisation

P:first-line {propriété 1; propriété 2}
P.classe1:first-letter {propriété 3; propriété 4}

Le Sélecteur ID : #id

Le sélecteur ***id*** sera utilisé à la mise en page qu'à la mise en forme de caractères. En effet, lorsqu'on applique un ***id*** à un élément, on lui attribue non seulement des styles mais une identité précise.
L'id ne doit donc désigner qu'un seul et unique élément d'un même document.
Dans la feuille de style, le nom de ***l'id*** (ici, footer_info) est précédé du signe dièse # :

```
#footer_info {
        background-color:#0000FF;
        text-align: center;
        width: 100%;
    }
```

Dans un fichier Html, on retrouvera ***id*** de cette façon (***sans le #):***

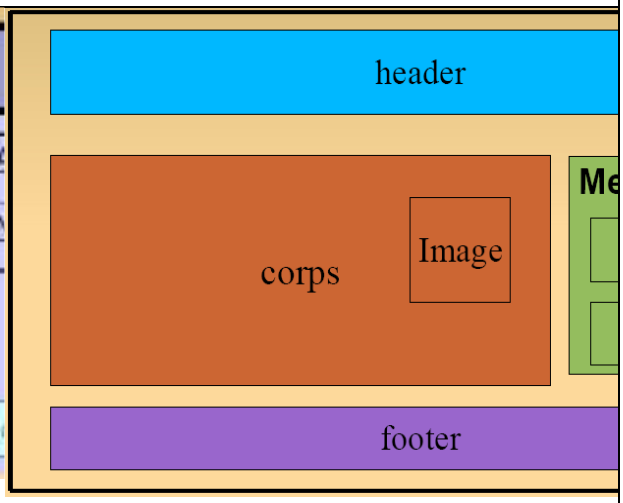
*** ****Le contenu de ce div se verra donc appliquer les styles définis ci-dessus. Par contre, il ne pourra pas y avoir 2 fois cet id dans la même page !**** ***

Pou commenter, il suffit de délimiter le texte par les caractères "/*" et "*/".

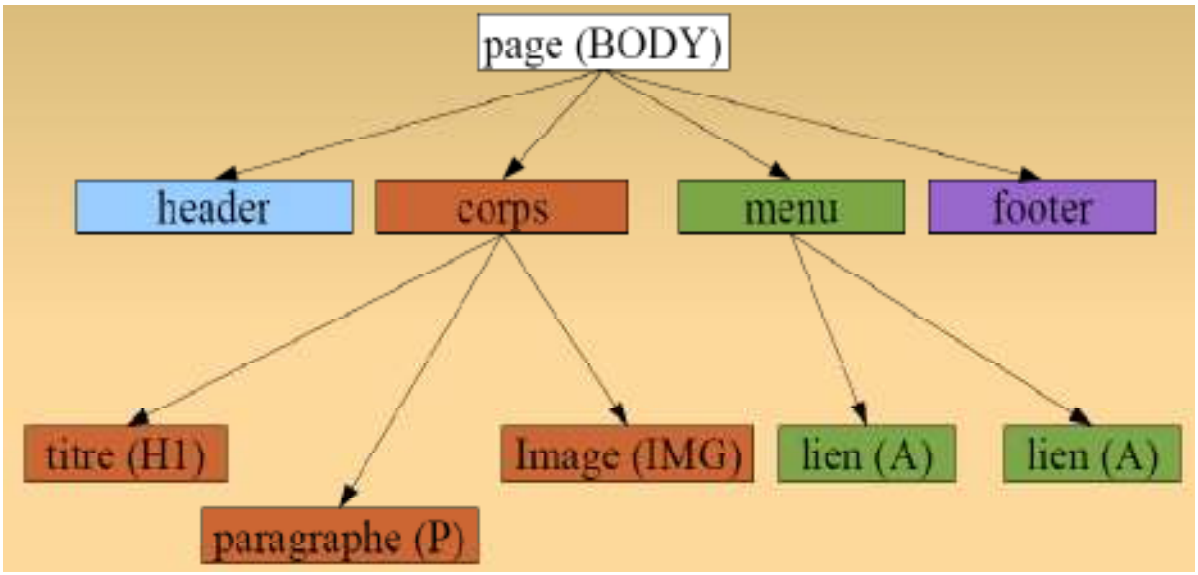
Mettre en page avec des CSS :

- Schématiser la feuille
- Repérez les différents éléments constituant de la page.
- Construire « l'arbre logique » de la page.
- Créer la page HTML : les boîtes seront matérialisées par des balises **div** munie de l'attribut class nécessaires.
- Construire la feuille de style CSS en descendant l'arbre.

Exemple : Ma page à Moi

Le schéma	La structure
	

Arbre logique



Donner le code en HTML.

La feuille de style CSS :

Je crée les styles dans l'ordre de mon arbre, en descendant.

1. la page (balise BODY)
2. le header, le corps, le menu, le footer
3. L'image et le titre du corps, les liens du menu, ...

Positionnement : notion de boîte

D'autres types de balises sont également connus comme des boîtes.

<table>
<tr>
<td>

- Flux vertical:
  - En HTML :

```
<p class="jaune">Une
  boîte jaune </p>
<p class="verte">Une
  boîte verte </p>
```

  - En CSS :

```
.jaune {
background-color: #ffff00;
        }
.verte {
background-color:#00ff00;
        }
```

</td>
<td>

- Flux horizontal:
  - En HTML :

```
<p>
<span class="jaune">Une
  boîte jaune</span>
<span class="verte">Une
  boîte verte</span>
</p>
```

  - En CSS :

```
.jaune {
background-color: #ffff00;}
.verte {
background-color: #00ff00;}
```

</td>
</tr>
<tr>
<td>Une boîte jaune
Une boîte verte</td>
<td>Une boîte jauneUne boîte verte</td>
</tr>
</table>

La balise ***<SPAN> ... </SPAN>*** permet d'appliquer des styles à des éléments de texte d'un paragraphe ou à un morceau de paragraphe.

La balise ***<DIV> ... </DIV>*** permet de regrouper plusieurs paragraphes ou de délimiter une zone comportant plusieurs paragraphes.

On peut utiliser les propriétés de positionnement pour organiser les positions des blocs.

Les types de positionnements : relatif, flottant et absolue.

- **Relatif** :

  Garde l'ordre par défaut mais permet un décalage horizontal ou vertical.

  On utilise la propriété, **position:relative;**

- **Flottant** :

  Fait sortir la boîte de l'ordre par défaut pour afficher le plus à gauche ou le plus à droite possible par rapport à son conteneur.

  On utilise la propriété, **float:left ou right;**

- **Absolue** :

  La position absolue se détermine par rapport au coin supérieur gauche de la page, les coordonnées étant top= 0 et left= 0.

  On utilise la propriété, **position:absolute;**

Exemples :

<table>
<tr>
<td>

- Positionnement relatif

- **En HTML** :

```
<p> avant Boîte
   <span class="jaune">
        boîte en position relative
       </span>
      après boîte.
</p>
```

- **En CSS** :

```
   .jaune {
position: relative;
bottom: 5px;
            background-color:#ffff00;
             }
```

</td>
<td>

- Positionnement flottant

- **En HTML** :

```
  <p class="jaune">Une boîte jaune flottante</p>
<p class="verte">Une boîte verte dotée d'un contenu
        plus long…</p>
```

- **En CSS** :

```
      .jaune {
background-color: #ffff00;
            float: right; width: 200px;
            margin: 0;
             }
      .verte {
             background-color: #00ff00;
             }
```

</td>
</tr>
<tr>
<td>avant Boîte boîte en position relative</td>
<td>Une boîte verte doté d'un contenu plus Une boîte ja
long… ce qui suit ne veux rien dire il a pour vocation du c
savcire le débordement du texte et comment le navigaeur
quand un positionnement flottant</td>
</tr>
</table>

Les styles de police

font-family définit un nom de police ou une famille de police <nom> ou <famille>

police précise (Arial, Times, Helvetica...) ou

famille (serif, sans-serif, cursive, fantasy, monospace)

H3 {font-family: Arial}

font-style définit le style de l'écriture.

normal ou italique ou oblique

H3 {font-style:italic}

font-weight définit l'épaisseur de la police

normal ou bold ou bolder ou lighter ou valeur numérique soit

(100 | 200 | 300 | 400 | 500 | 600 | 700 | 800 | 900)

P {font-weight:bold}

font-size définit la taille de la police.

xx-small ou x-small ou small ou médium ou large ou x-large ou xx-large

oularger ou smaller

ou taille précise en points (pt), inches (in), centimètres (cm), pixels (px)

ou pourcentage (%)

P {font-size: 12pt}

font-variant définit une variante par rapport à la normale

normalou small-caps

P {font-variant: small-caps}

font raccourci pour les différentes propriétés de police

P {font:bolditalic}

Les styles du texte

text-align définit l'alignement du texte.

left ou center ou right ou justify

H1 {text-align: center}

text-indent définit un retrait dans la première ligne d'un bloc de texte souvent utilisé avec <P>, n'oubliez pas dans ce cas </P>.

spécifié en inches (in) ou en centimètres (cm) ou en pixels (px)

P {text-indent: 1cm}

text-decoration définit une décoration du texte, soit barré, clignotant, etc.

blinkou underline ou line-through ouoverlineou none

A:visited {text-decoration: blink}

text-transform définit la casse du texte (majuscule, minuscule)

uppercase (met les caractères en majuscules) ou

lowercase (met les caractères en minuscules) ou

capitalize (met le premier caractère en majuscule)

P {text-transform:uppercase}

colordéfinit la couleur du texte.

H3 {color: #000080}

word-spacing définit l'espace entre les mots

en points (pt), inches (in), centimètres (cm), pixels (px)

ou pourcentage (%)

P {word-spacing: 5pt}

letter-spacing définit l'espace entre les lettres.

spécifié en points (pt), inches (in), centimètres (cm), pixels (px)

ou pourcentage (%)

P {letter-spacing: 2pt}

line-height définit l'interligne soit l'espace entre les lignes du texte
en points (pt), inches (in), centimètres (cm), pixels (px)
ou pourcentage (%)
P {line-height: 10pt}

width détermine la longueur d'un élément de texte ou d'une image.
en points (pt), inches (in), centimètres (cm), pixels (px)
ou pourcentage (%)
H1 {width: 200px}

height détermine la hauteur d'un élément de texte ou d'une image.
en points (pt), inches (in), centimètres (cm), pixels (px)
ou pourcentage (%)
H1 {heigh: 100px}

white-spacedétermineespace ou blanc
normal ou pre ou nowrap
PRE {white-space:pre}

Les arrière-plans

background-color définit la couleur de l'arrière-plan.

couleur (par exemple en hexadécimal) ou transparent

H1 {background-color: #000000}

background-image : définit l'image de l'arrière-plan.

URL de l'image

BODY {background-image: image.gif}

background-repeat définit la façon de répéter l'image d'arrière-plan.

repeat ou no-repeat ou repeat-x (x = nombre de répétitions horizontales) ou

repeat-y (y = nombre de répétitions verticales

P {background-image: image.gif; background-repeat: repeat-4}

background-attachment spécifie si l'image d'arrière-plan reste fixe avec les déplacements de l'écran.

scrollou fixed

BODY {background-image: image.gif; background-attachement: fixed}

background-position spécifie la position de l'image d'arrière-plan par rapport au coin supérieur gauche de la fenêtre

{1, 2}

{top ou center ou bottom ,left ou center ou right}

ou en points (pt), inches (in), centimètres (cm), pixels (px)

oupourcentage (%)

BODY {background-image: img.gif; background-position: right top}

background raccourci pour les différentes propriétés d'arrière-plan

P {background: image.gif fixed repeat}

Les marges

margin-top détermine la valeur de la marge supérieure en unité de longueur ou auto.

{ margin-top: 5px }

margin-right détermine la valeur de la marge droite en unité de longueur ou auto

{ margin-right: 5px }

margin-bottom détermine la valeur de la marge inférieure en unité de longueur ou auto

{ margin-bottom: 5px }

margin-left détermine la valeur de la marge gauche en unité de longueur ou auto

{ margin-left: 5px }

margin regroupe les différentes propriétés de la marge

Les bords et les "enrobages"

border-top-width donne l'épaisseur du bord supérieur

thin ou medium ou thick ou spécifié par l'auteur

H3 {border-top-width: thin}

border-right-width donne l'épaisseur du bord droit

thin ou medium ou thick ou spécifié par l'auteur

H3 {border-right-width: medium}

border-bottom-width donne l'épaisseur du bord inférieur

thin ou medium ou thick ou spécifié par l'auteur

H3 {border-bottom-width: thick}

border-left-width donne l'épaisseur du bord gauche

thin ou medium ou thick ou spécifié par l'auteur

H3 {border-left-width: 0.5cm}

border-width regroupe les différentes propriétés de border-width

border-color détermine la couleur de la bordure

H3 {border-color:yellow}

border-style détermine le style du trait de la bordure

none ou solid ou dotted ou dashed ou double

ou groove ou ridge ou inset ou outset

{border-style:soliddashed}

borderregroupe toutes les propriétés des bords

padding-top valeur de remplissage haut entre l'élément et le bord

en points (pt), inches (in), centimètres (cm), pixels (px)

oupourcentage (%)

H3 {padding-top: 3px}

padding-right valeur de remplissage droite entre l'élément et le bord

en points (pt), inches (in), centimètres (cm), pixels (px)

oupourcentage (%)

H3 {padding-right: 3px}

padding-bottom valeur de remplissage bas entre l'élément et le bord

en points (pt), inches (in), centimètres (cm), pixels (px)

oupourcentage (%)

H3 {padding-bottom: 3px}

padding-left valeur de remplissage gauche entre l'élément et le bord

en points (pt), inches (in), centimètres (cm), pixels (px)

ou pourcentage (%)

H3 {padding-left: 3px}

padding regroupe les différentes propriétés de remplissage

Les listes

list-style-type détermine le type de puces ou de numérotation

disc ou circle ou square

decimal ou lower-roman ou upper-roman ou lower-alpha ou upper-alpha

OL {list-style-type: square}

list-style-image permet de remplacer les puces par une image

url ou none

OL {list-style-image: image.gif}

list-style-position spécifie si les puces sont à l'intérieur ou à l'extérieur du texte

inside ou outside

UL {list-style-position:inside}

list-style regroupe toutes les propriétés de liste

Chapitre 5 :JavaScript

Introduction

JavaScript est un langage de programmation très simple et constitue une excellente initiation à la programmation web.

La création de programmes JavaScript requiert très peu de connaissances.

Il est facile d'utiliser JavaScript pour améliorer les pages Web crées en HTML.

Les programmes en JavaScript peuvent être composés d'une seule ligne.

Cependant il est possible d'utiliser JavaScript pour créer des applications complexes.

Scripts et programmes

Un **script** en JavaScript peut être constitué d'une seule ligne ; il peut également être une application complète.

Le script s'exécutera dans un navigateur.

JavaScript, pour sa part est un langage interprété : le navigateur exécute chacune des lignes du programme au fur et à mesure.

Il est aussi simple de changer le contenu d'un script JavaScript que celui d'un document HTML.

Intégrer un script JavaScript à une page Web

<script> : interpréter la suite du document comme un script.

</script> : interpréter la suite du document comme un HTML.

Les instructions JavaScript doivent toujours être placées à l'intérieur des balises **<script></script>**.

Un script peut être placé à quatre endroits différents :

- Dans le corps de la page : le résultat du script s'affiche au sein du document HTML lorsque la page est chargée.
- Dans l'en-tête de la page, à l'intérieur des balises Head : lorsqu'un script est placé dans l'en-tête, il n'est pas exécuté immédiatement, mais d'autres scripts peuvent s'y référer. L'entête est souvent employé pour les fonctions.
- A l'intérieur d'une balise HTML : on appelle cela un gestionnaire d'événements ; il permet au script d'interagir avec des éléments HTML.
- Dans un fichier séparé : vous pouvez stocker des scripts dans des fichiers comportant

l'extension .js ; pour vous y référer vous indiquerez le nom de fichier dans la balise <script>.

Stockage et utilisation des valeurs

Les variables

Les variables sont des conteneurs nommés qui peuvent stocker des données : un nombre, une chaîne de caractères ou un objet.

Vous pouvez à tout moment modifier la valeur d'une variable.

Le nom de la variable prend les mêmes précautions vues avec C.

Affecter des valeurs à des variables

Pour affecter une valeur à une variable c'est le même principe qu'en langage C.

Exemples d'affectation :

```
lignes = 40;
lignes = lignes -30;
```

Exemple d'incrémentation / décrémentation:

```
lignes = lignes +1; lignes = lignes-1;  lignes++; lignes--;
```

Types de données en JavaScript

- les nombres : 25 ; 3.14 …. JavaScript permet l'utilisation à la fois des entiers et des nombres en virgule flottante.
- Les valeurs logiques ou booléennes : sont true et false.
- Les chaînes de caractères : "…" ou '…' ; les chaînes sont constituées d'une série de caractères.
- La valeur nulle : représentée par null, la valeur d'une variable non définie.

Remarque :

Javascript enregistre le type de donnée stocké dans chacune des variables, mais vous pouvez Modifier le type de variable en cours de script.

Conversion de données d'un type à un autre

JavaScript convertit automatiquement les données d'un type à un autre chaque fois que c'est possible.

parseInt() : convertit une chaîne en nombre entier.

parseFloat() : convertit une chaîne en nombre en virgule flottante.

Exemple :

chaîne = "30 arbres coupés";

Nombre= parseInt(chaîne);

Apres l'exécution de ces deux instructions la variable nombre contiendra la valeur 30.

Variables locales et globales

Fonctions

Une **fonction** est un ensemble d'instructions JavaScript qui peut être exécutée comme une seule unité.

On déclare généralement les **fonctions** dans l'en-tête **(<head> … </head>)** afin qu'elles puissent être appelées de n'importe où dans la page.

L'appel d'une fonction fait par son **nom**.

L'appel d'une fonction doit toujours comporter les **parenthèses**, même si la fonction ne prend pas d'argument.

Une fonction ne s'exécutera pas tant que l'on n'y aura pas fait appel quelque part dans la page.

Exemple : Définition d'une fonction :

```
functionbonjour([Arg1],[arg2],…)

{

[utilisation des arguments]
alert("bonjour tout le monde");

}

function Somme(a,b)
```

{

c=a+b;

return c;

}

Exemple :

<HTML><head>

<title> ma premiere page JavaScript </title>

<script language="JavaScript">

functionbonjour(arg)

{

alert("bonjour tout le monde" + arg);

}

functionSomme(a,b)

{

var c=a+b; // remarquer l'utilisation du mot clé var
return c;}

</script>

</Head>

<body>

<h1> ma première page JavaScript </h1>

 bienvenue sur notre page web. Elle en cours de construction, sa dernière modification remonte à la date suivant :

<script language="JavaScript">

bonjour("smi"); */* ceci est un commentaire : ici on fait appel à la fonction qu'on a définit dans le <Head> */*

var c=somme(1,2)*; // remarquer la deuxième utilisation du mot clé var*

```
</script>

</body></HTM
L>
```

Il existe deux types de variables :

- Les ***variables globales*** peuvent être utilisées dans tout le script, et dans d'autres scripts du même document HTML.

  Pour créer une variable globale, il suffit de la déclarer dans le script *principal* à ***l'extérieur*** de toute fonction.

  Vous pouvez utiliser le mot clé ***var*** pour déclarer la variable globale.

- Les ***variables locales*** peuvent être employées qu'à ***l'intérieur*** de la fonction où elles ont été crées.

  Toute variable déclarée ou utilisée pour la première fois dans une fonction est une variable locale.

  Les variables faisant partie de la liste des paramètres d'une fonction sont des variables locales.

  Utilisez le mot clé ***var*** pour déclarer une variable locale à l'intérieur d'une fonction, même s'il existe une variable globale qui porte le même nom.

Les commandes essentielles

L'objet String

Les chaînes permettent de stocker des ensembles de caractères.

JavaScript stocke les chaînes en tant qu'objets ***String***.

Exemple :

Les deux instructions créent une chaîne de caractère de deux manières différentes.

```
test  ="ceci est un test";

test = new String("ceci est un test");
```

Exemple : affectation de valeurs aux chaînes

```
<HTML><Head><title> les chaînes en JavaScript
```

```
</title></Head>

<body><h1> les chaînes en JavaScript </h1>

<script language="JavaScript">

test_1 = "ceci est un test";

test_2 = " ne vous inquiétez pas";

chaine=test_1 + test_2; // le signe + entre deux chaînes réalise la concaténation

document.write(test_1);

document.write("<p>"+test_2 + "</p>");

document.write("<p>"+chaine+ "</p>");

</script></body></HTML>
```

Quelques méthodes de l'objet string :

chaîne.***length***;	La longueur de la chaîne
Chaîne=chaîne.***toLowerCase***();	retourne une chaîne en minuscules
Chaîne =chaîne.***toUpperCase***();	retourne une chaîne en majuscules
Chaîne=chaîne.***charAt(n***);	Retourne un seul caractère d'indice n

document.write(chaîne.***substring(n,m)***); : renvoie une chaîne extraite d'une autre chaîne en fonction des deux valeurs d'indice que vous indiquerez comme paramètres.

Le premier caractère de la chaîne a une ***valeur d'indice de 0.***

Le second paramètre est ***exclusif***,

Position = chaîne.***indexOf***("char"); : rechercher la chaîne char au sein d'une chaîne et commence la recherche depuis le début de la chaîne mère.

Position = chaîne.***indexOf***("char",n); : le deuxième argument pour préciser d'où commencer la recherche.

La variable Position contient la valeur d'indice qui correspond au premier numéro d'indice de substring.

Position = chaîne.***lastIndexOf***("char",[valeur]); : de même que indexOf, mais celle là commence depuis la fin de la chaîne.

Les tableaux numériques

Un tableau est une série d'éléments de données qu'il est possible de manipuler comme une seule unité.

Les tableaux peuvent contenir des chaînes; des nombres, des objets ou d'autres types de données.

Les tableaux doivent être déclarés avant de pouvoir être utilisés.

Pour affecter des valeurs au tableau, utilisez des ***crochets*** et des numéros ***d'indice***.

La propriété ***length*** du tableau correspond au nombre d'éléments dont il est constitué.

Exemple : tableau numérique :

```
Scores = new Array(30) ; //créer un tableau
Scores[0]=39 ; //affecter des valeurs au tableau
Scores[8]=12; //affecter des valeurs au tableau
Scores[24]=2 ; //affecter des valeurs au tableau

document.write(Scores.length) ; //nombre d'éléments du tableau

Affichage_Scores= ''Scores : '' +Scores[0]+ ''  ''+Scores[24]+''  '';

Document.write(Affichage_Scores) ;
```

La méthode Chaîne.***split***('' '') divise une chaîne en plusieurs parties.

Elle divise la chaîne au niveau de chacun de caractères ***d'espacement***.

Les parties de chaînes résultantes sont stockées dans un ***tableau***.

La méthode chaîne.***join***('' '') réalise la ***jointure*** des chaînes .

La méthode **join** est l'inverse de **split**.

Exemple :

```
Chaîne_a_diviser=''pourquoi on aime la vie'' ;
Parties_de_la_chaîne=Chaîne_a_diviser.split(''  '') ;
Parties_de_la_chaîne[0]= ''pourquoi'' ;
Parties_de_la_chaîne[2]=''aime'' ;
Chaîne_jointure=parties_de_la_chaine.join('' '') ;
Document.write(chaîne_jointure) ;
```

Tri de tableau

La méthode ***sort*** permet de trier le contenu d'un tableau.

Cette méthode renvoie une version triée de votre tableau du ***petit vers le plus grand*** dans le cas d'un tableau numérique et par ordre ***alphabétique*** dans le cas d'un tableau de chaînes.

Exemple:

```
Noms[0]= ''amine'' ;
Noms[1]=''mohamed'' ;
Noms[2]=''ahmed'' ;
Noms[3]=''said'' ;
Noms=Noms.sort();
```

Test et comparaison

- If … then … else

```
If (condition)
        {faire ceci ;}
else
        {faire cela; }
```

Opérateurs de comparaison des valeurs:	Opérateurs logiques
= = : égale à != : Différent de. < : Inférieur à. > : Supérieur à. <= : Inférieur ou égale. >= : Supérieur ou égale	&& : et || : ou

- Conditions multiples ***switch*** :

```
Switch( condition)
{
Case 1 : faire ;
            Break ;
Case 2 : faire ;
            Break ;
Case 3 :faire ;
            Break ;
….
Default : faire ;
}
```

Break permet de sortir le bloc swith.

Les boucles:

Le boucle for :

La boucle for comporte son propre compteur.
for (initialisation; condition d'entrée ; évolution).
{
Instructions … }

Exemple :

Varxsum=0 ;
Var x ;
For (x=1 ;x<=10 ;x++)
{
Xsum+=x;
}
	Varxsum=0 ;
	For (x=1,inct=0 ;x<=10 ;x++,inct++)
	{
	Xsum+=x;
	}

La boucle *while* :

La boucle **while** est indiquée quand le nombre de boucles à exécuter est inconnu.

While (condition d'entrée)
{
Instructions …}
La boucle do … while :
Le code est exécuted'abord avant la vérification de condition.
Do
{
Instructions …}
While (condtion);

La boucle *for* … in

Cette boucle permet d'effectuer des opérations sur les propriétés d'un objet.

Exemple :

```
For (i in navigator)
{
Document.write(''la propriété est : " +i) ;
Document.write(''la valeur est : " +navigator[i]) ;
}
```

Fonctionnement de base des objets en JavaScript

- Un ***objet*** JavaScript est constitué de ses ***propriétés*** (sa description, ses caractéristiques) et de ***méthodes*** (ce qu'il sait faire).

  Les ***objets*** permettent de combiner différents types de données (propriétés) et de fonctions permettant d'agir sur ces données (méthodes) en un seul élément facile à manipuler.

- La ***propriété*** d'un objet est l'équivalent d'une variable qui serait contenue dans un objet.

  Syntaxe : ***objet.propriété***

- Les ***méthodes*** sont des fonctions stockées en tant que propriétés d'objets.

  Syntaxe : ***objet.méthode(argument)***

- Le mot clé ***with***permet de simplifier l'écriture des programmes en JavaScript ou du moins la quantité de code à saisir.

  Le mot clé ***with*** permet de spécifier un objet : il est suivi d'un bloc d'instructions placées entre accolades. Pour chacune des instructions du bloc, les propriétés mentionnées sans que l'objet correspondant soit indiqué se réfèrent à l'objet indiqué après ***with***.

Exemple :

```
prénom =
"Foulan";with(prénom)

{

document.write("<p>la longueur est : " + length);
ch=toUpperCase();

document.write("<p>la chaîne est " + ch);

}
```

- # L'objet Math

Math est un objet prédéfini de JavaScript qui comprend de nombreuses constantes et fonctions.

Les propriétés de l'objet ***Math*** sont des constantes mathématiques, ses méthodes des fonctions mathématiques.

Liste des principales méthodes

- Math.sqrt() , racine carrée.
- Math.log() , Math.exp() , Math.abs() ,Math.cos (), Math.sin() , Math.tan()
- Math.floor(), Math.ceil(),entier immédiatement inférieur /supérieur.
- Math.pow(base, exposant), fonction puissance, où base et exposant sont des expressions numériques quelconques évaluables.
- Math.max() , Math.min()
- Math.random(), nombre "réel" choisi au hasard dans [0 , 1[.
- Math.round(), arrondit à l'entier le plus proche.

Exemple :

```
<script>
r=10;
with (Math )
{
s = PI * pow(r , 2);
theta = PI / 3;
x = r * cos( theta );
y = r * sin( theta);
document.write("PI = ",PI, "<br>");
document.write("s = ", s,"<br> Coordonnées de M : x = ",x, " y = ",y)
}
</script>
```

- # Les dates

Date est un objet prédéfini de JavaScript qui permet de manipuler facilement dates et heures.

L'origine des dates a été choisie le 1er janvier 1900 et est exprimée en millisecondes.

Pour construire un objet de type **Date**, il faut utiliser un constructeurDate() avec le mot-clé new

variable = **new Date**(*liste de paramètres*)

Les secondes et les minutes sont notées de 0 à 59.

Les jours de la semaine de 0 (dimanche) à 6.

Les jours du mois de 0 à 30.

Les mois de 0 (janvier) à 11.

Les années sont décomptées depuis 1900.

Les méthodes ***set*** permettent définir les valeurs des objets Date.

setDate(): définit le jour du mois ;

setMonth(): définit le mois;

setFullYear(): définit l'année en quatre chiffres ;

setHours(), setMinutes() et setSecondes() : définit l'heure.

*L*es méthodes ***get*** permettent de lire les valeurs des objets Date. Le recours à ces méthodes est obligatoire dans la mesure où les valeurs de ces objets ne sont pas disponibles en tant que propriétés.

getDate(): permet de lire le jour du mois ;

getMonth(): permet de lire le mois ;

getFullYear(): permet de lire l'année;

getHours(), getMinutes() et getSecondes()permettent de lire lesheures, les minutes et les secondes.

JavaScript comprend plusieurs fonctions permettant de gérer les fuseaux horaires et les heures locales.

getTimeZoneOffset(): renvoie le décalage du fuseau horairelocal par rapport à l'heure GMT. En l'occurrence, le fuseau horaire local est celui du navigateur de l'utilisateur.

getGMTString() : convertit en texte la valeur de l'objet Date enfonction de l'heure de GMT.

toLocalString(): convertit en texte la valeur de l'objet Date enfonction de l'heure locale.

Les objets du navigateur

- ### Le fonctionnement des objets de navigateur

JavaScript permet de manipuler directement le navigateur Web.

Vous pouvez utiliser un script JavaScript pour charger une nouvelle page dans le navigateur, manipuler des parties de la fenêtre et même ouvrir de nouvelles fenêtres.

Pour effectuer ces manipulations, JavaScript fait appel à des objets de navigateur.

Les objets du navigateur ont des propriétés qui décrivent la page Web ou le document et des méthodes qui permettent de les modifier.

On déduit que les objets de navigateur sont structurés en une hiérarchie d'objets parents et filles. Lorsqu'on se réfère à un objet, on utilise l'objet parent suivi du ou des noms des objets filles.

L'objet ***window*** se situe au sommet de la hiérarchie des objets de navigateur.

- ### Manipulation des documents Web

L'objet ***document*** représente un document Web.

Les documents Web sont affichés dans la fenêtre du navigateur

Dans la mesure où l'objet ***window*** représente toujours la fenêtre en cours (celle qui contient le script), vous pouvez utiliser ***window.document*** pour vous référer au document en cours.

Vous pouvez également vous référer uniquement à ***document*** : vous désignez ainsi automatiquement le document de la fenêtre en cours.

Si plusieurs fenêtres ou cadres sont ouverts, il existera plusieurs objets ***window*** et

un objet ***document*** pour chacun d'eux.

Propriétés d'un objet *document :*

location : indique l'URL du document.

title : représente le titre du document.

referrer : l'adresse d'où l'on vient.

lastModified : est la date de dernière modification de la page.

linkcolor : la couleur des liens.

Quelques méthodes de l'objet document :

write() : écrit du texte ou de l'html dans le document.

writeln() : idem, mais suivi d'un retour-chariot.

close() : fermeture du document en cours.

• Accès à l'historique du navigateur

L'objet *history* est une propriété fille de l'objet ***window***.

Il contient des informations sur les URL qui ont été visitées avant et après la page courante, ainsi que des méthodes pour y accéder :

L'objet ***history*** dispose de trois méthodes :

- ***history.go()***: qui ouvre une URL de l'historique.

  Pour l'utiliser,indiquez un nombre positif ou négatif comme paramètre.

  Exemple :

  history.go(-2): équivaut à cliquer deux fois sur Précédente.

- ***history.back()***: qui affiche la page précédente.
- ***history.forward()***: qui affiche la page suivante.

L'objet ***history*** dispose de quatre propriétés :

- ***history.length***: indique la longueur de l'historique;ie : lenombre de pages différentes visitées par l'utilisateur.
- ***history.current***: qui contient la valeur courante de l'historique,l'URL de la page que l'utilisateur est en train de visualiser.
- ***history.next***: cette propriété ne contient pas de valeur que sil'utilisateur à déjà cliqué sur Précédente.
- ***history.previous***: qui est la valeur de l'élément précèdent delhistorique, ie : la page où sera envoyé l'utilisateur s'il clique sur Précédente.

- **L'objet *location***

L'objet ***location*** est lui même fille de l'objet ***window***.

Il stocke des informations concernant l'URL de la page affichée dans la fenêtre et permet de charger de nouvelles pages.

Ainsi, pour charger une URL dans la page courante, on utilisera ce qui suit : ***window.location.href = www.caramail.com***

L'objet location dispose de deux méthodes :

location.reload(): qui permet de recharger le document courant et quiéquivaut à un clic sur actualiser.

location.replace(): qui remplace la page courante par une nouvelle page.Cette méthode est similaire à l'emploi de ***location.href*** pour charger une nouvelle page, à la différence qu'avec ***location.replace()***, l'historique du navigateur n'est pas modifié.

Autrement dit, l'utilisateur ne pourra pas revenir à la page précédemment affiché à l'aide de Précédente.

Création d'objets personnalisés

- **Définir un objet**

La première étape de la création d'un objet consiste à lui donner un nom et à nommer ses propriétés.

Puis on doit créer une fonction *(ce qu'on avait appelé: Constructeur)*pour pouvoir instancier notre objet.

Exemple :

```
functionpersonne(nom,prenom,pro,dom)

{

this.nom         = nom;
this.prenom = prenom;
this.telpro              = pro;

this.teldom = dom;}
```

Le constructeur est une fonction simple acceptant des paramètres pour initialiser le nouvel objet, puis les affectant aux propriétés correspondantes.

La fonction s'appelant ***personne***, l'objet lui aussi portera le nom de personne.

Donc : *P = new personne("zhong","ahmed","0665557744","0553666992");*

Note 1 *: le mot clé* ***this*** *permet de se référer à l'objet courant, ie : celui crée par la fonction.*

Note 2 *: il faut très bien remarquer que la définition du constructeur ne contient pas de méthodes.*

- ## Définir une méthode pour un objet

On va créer à ce moment une méthode pour manipuler l'objet ***personne***.

On va créer une méthode qui sert à afficher les propriétés de l'objet ***personne***.

Exemple :

```
functionAfficher()

 {ligne1 = "<b> Nom : </b>" + this.nom + "<br>"; ligne2 = "<b>Prenom : </b>"
 + this.prenom + "<br>";

ligne3 = "<b><pre> Le téléphone professionnel : </pre></b>" + this.telpro +
"<br>";

ligne4 = "<b><pre> Le téléphone de domicile : </pre></b>" + this.teldom +
"<br>";

document.write(ligne1,ligne2,ligne3,ligne4);  }
```

La fonction d'affichage n'est pas encore reconnue par l'objet ***personne***, il faut qu'elle en fasse partie de la définition de l'objet ***personne***.

```
functionpersonne(par_nom,par_prenom,par_pro, par_dom)
{
this.nom            = par_nom;
this.prenom = par_prenom;
this.telpro         = par_pro;
this.teldom = par_dom;
this.Afficher() = Afficher();
}
```

- # Création d'instances d'objet

Pour créer une instance de l'objet défini, on utilise le mot clé ***new***. Donc :

```
M = new personne("zhong","ming","557744","556699");
S = new personne("pepoloni","stephano","5267744","5566699");
```

On peut créer une instance d'objet vide, puis lui affecter des valeurs, comme suit :

```
A = new personne();
A.nom ="maziane";
A.prenom="Abdel"; ....
```

Pour afficher les propriétés de ces objets, on peut utiliser la méthode Afficher de chaque objet.

Donc :*M.Afficher();S.Afficher(); A.Afficher();*

- # Personnalisation d'objets prédéfinis

Il est possible d'étendre les fonctions des objets prédéfinis de JavaScript.

Pour ajouter des propriétés ou des méthodes à un objet ***String***, on utilise le mot clé ***prototype***qui permet de modifier la définition d'un objet à l'extérieur de sonconstructeur.

Exemple : on ajoutera une méthode titre à l'objet ***String*** qui convertit une chaîne de caractères en titre html.

```
<HTML>
<Head>
<title> Personnalisation d'objets prédéfinis</title>
<script language="JavaScript">
functionajout_titre(niveau)
{
balise = "H" + niveau;
debut = "<" + balise + ">";
fin = "</" + balise + ">";
contenu = this.toString();
resultat = debut + contenu + fin;
returnresultat;
}
functionlength_nouvelle(c)
{
tab_temp_1 = new Array();
tab_temp_1=this.split("");
x=tab_temp_1.length
for(i=0;i<tab_temp_1.length;i++)
{
if(tab_temp_1[i] == c) x=x-1;
}
```

```
return x;
}
</script>
</Head>
<body>
<h1> Personnalisation d'objets prédéfinis </h1>
<script language="JavaScript">
String.prototype.titre = ajout_titre;
String.prototype.nv_length = length_nouvelle;
texte = "je suis un titre";
document.write("<p>" + texte.titre(1));
document.write("<p>" + texte.titre(2));
document.write("<p>" + texte.titre(3));
/*******************************/
tab=texte.split(""); document.write("<p>" + texte.length);
document.write("<p>" + tab.length + "<p>");
document.write("<p>" + texte.nv_length("s") + "<p>");
</script>
</body></HTML>
```