Plan du cours

0. Les bases de la programmation web

0.1. Le web et le site web

Plusieurs définitions sont proposées pour expliquer le terme web que les anglophones appellent « World Wide Web », d'où l'acronyme www, et les francophones la « toile mondiale » :

- Le web est un moyen simple de cliquer pour explorer le volume gigantesque des pages d'informations situées sur Internet.
- Et, le web est une des possibilités offertes par le réseau Internet de naviguer entre des documents reliés par des liens hypertextes.

Le principe de web repose sur l'utilisation d'hyperliens pour naviguer entre des documents (appelés « pages web ») grâce à un logiciel appelé navigateur, en anglais « browser ».

Une page web est un document électronique contenant du texte, du son, des images fixes ou animés, ainsi que des liens hypertextes vers d'autres pages web. Ces hyperliens permettent aux utilisateurs de naviguer, de « fureter » ou de « surfer » parmi les informations, d'une manière résolument non séquentielle.

Un site web (appelé également site internet) est un ensemble de pages reliées par des liens hypertextes, auquel on accède sur le web par une ou plusieurs adresses qui lui sont propres. C'est aussi un ensemble de fichiers HTML stockés sur un ordinateur connecté en permanence à internet et hébergeant les pages web (serveur web).

Un site web est habituellement architecturé autour d'une page centrale, appelée «page d'accueil» et proposant des liens vers un ensemble d'autres pages hébergées sur le même serveur, et parfois des liens dits «externes», c'est-à-dire de pages hébergées par un autre serveur.

Chaque site propose un ensemble plus ou moins important de documents, transmis sur le réseau par l'intermédiaire d'un programme serveur. Ce programme serveur dialogue avec un programme client qui peut être situé n'importe où sur le réseau. Le programme client prend le plus souvent la forme d'un navigateur, grâce

auquel un utilisateur du Web peut demander et consulter très simplement des documents.

Le dialogue entre un programme serveur et un programme client s'effectue selon des règles précises qui constituent un protocole. Le protocole du Web est HTTP, mais il est souvent possible de communiquer avec un site via d'autres protocoles, comme par exemple le FTP qui permet d'échanger des fichiers.

La création d'un site web est un projet à part entière comprenant 3 phases :

- La conception qui représente la formalisation de l'idée ;
- La réalisation qui correspond au développement du site web ;
- L'hébergement qui se rapporte à la mise en ligne du site, de manière permanente.

0.2. Le site web statique et le site web dynamique

On appelle « page web statique », une page web qui est représenté par un fichier texte ne contenant que des code (X)HTML et CSS, avec éventuellement des images animées, des éléments défilants et des liens vers d'autres documents.

Un site constitué de pages web statiques est ainsi qualifié de « site web statique ».

Ce site fonctionne très bien, mais son contenu ne peut pas être mis à jour automatiquement : il faut que le webmaster modifie le code source pour y ajouter des nouveautés. Ce n'est pas très pratique quand on doit mettre à jour son site plusieurs fois dans la même journée.

Et un site dynamique est alors constitué des pages dynamiques, c'est-à-dire des pages qui sont générées à la volée par des programmes (serveurs web). Leurs contenus peuvent changer sans l'intervention du webmaster.

La plupart de sites web que nous visitons aujourd'hui, sont des sites dynamiques, exemples Facebook, mediacongo etc.

Un site sur Internet est une entreprise coûteuse en termes de temps. Les mises à jour régulières sont parfois difficiles à réaliser. Or, un site qui n'évolue pas est condamné à disparaitre ou du moins à voir sa fréquentation diminuer. L'internaute aime le changement, il ne supporte pas longtemps un site statique qui n'évolue guère. Les langages dynamiques permettent de résoudre ces problèmes. Ils

facilitent les opérations de mise à jour, permettent plus d'interactivité sur les pages…

0.3. Les différents types de Sites

a) Le site « carte de visite »

Il s'agit de la forme la plus basique de site web. Un site de type « carte de visite » ne regroupe que les informations nécessaires pour entrer en contact avec vous et votre entreprise. Elle ne propose que très peu de contenu et n'a pas de valeur ajoutée en soit. Il existe même un domaine de premier niveau dédié à cet usage.

b) Le site vitrine

Comme son nom l'indique, un site vitrine est là pour exposer votre entreprise sur le web. Il peut être composé de plusieurs pages et donne à l'internaute toutes sortes d'informations sur votre entreprise : vos produits, vos services, vos coordonnées et éventuellement quelques fonctions supplémentaires comme par exemple une section « actualités », une galerie photos ou encore un formulaire de prise de contact ou de demande de devis. Pour le rendre plus efficace et visible, il est recommandé de le coupler à d'autres services tel un blog ou une newsletter.

c) Le site de e-commerce

Si vous souhaitez vendre des produits ou des services en ligne, vous devez alors opter pour un site e-commerce. Il est impossible de résumer en quelques lignes les caractéristiques complètes d'un site marchand car il existe autant de sites que de commerçants.

d) Le site mobile

De plus en plus de connexions au web se font depuis un téléphone mobile ou un smartphone. Si une grande partie de votre cible est composée d'internautes nomades, il est important de disposer d'un site web adapté aux petits écrans de ces appareils mobiles.

e) L'application web

Une application web est un programme qui s'exécute depuis votre navigateur web. Plus besoin d'installer un logiciel sur votre poste, d'assurer sa maintenance et ses

mises à jour : tout se fait depuis votre navigateur. De plus, l'application est accessible depuis n'importe quel ordinateur connecté au web. On peut citer comme applications web des services de facturation en ligne ou de gestion de projets.

f) Le site communautaire

C'est un site qui fait la part belle à ses membres et propose de nombreux services autour de la communication entre internautes. Le plus connu de ce site est bien entendu Facebook. On peut également citer des sites basés sur des forums ou des jeux en ligne comme Piplex.

g) Le blog

On ne présente plus le blog, ce symbole même du web 2.0 qui vous permet de publier, jour après jour des billets dans lesquels vous pouvez écrire vos humeurs mais aussi des articles d'actualité ou d'analyse d'un domaine particulier. Le blog est un outil formidable pour faire la promotion de votre activité, de renforcer votre réputation en ligne tout en améliorant votre référencement naturel.

h) Le site spécifique, sur mesure

Cette dernière catégorie regroupe en fait tous les autres types de sites qui n'entrent pas dans celles présentées précédemment. Ce sont des sites qui sont développés sur mesure pour un client qui a des besoins spécifiques après l'étude d'un cahier des charges.

0.4. Les outils de développement web

Nous présentons dans ce point certains outils permettant de faire du développement web. Il s'agit notamment de :

- Serveur web : c'est le programme le plus important qui est chargé de délivrer des pages web aux visiteurs. Les principaux serveurs web sont : apache, Internet Information Server (IIS) et Personal Web Server (PWS).
- Navigateur web : c'est le programme chargé d'afficher le contenu du fichier enregistré sur le serveur après l'avoir interprété. Les principaux navigateurs web sont : Internet Explorer, Firefox et Netscape.
- Langages de scripts côté serveur : PHP, Perl, Java, VBScript, …

- Langages de scripts côté navigateur : Javascript, VBScript, Perlscript, Java.

Le script étant un programme simple consistant en un ensemble d'instructions destinées à exécuter ou automatiser des tâches ou fonctions spécifiques.

A ceux-ci, on associe également une application de gestion de base de données.

Dans le cadre de ce cours, nous allons nous attarder sur les langages de scripts PHP côté serveur et Javascript côté navigateur. La base de données sera en MySQL. Nous signalons en pensant que Javascript peut également être utilisé côté serveur.

Le logiciel Wampserver que nous allons utiliser est très pratique en ce qu'elle amène dans un même paquetage : le serveur apache, le langage PHP, le système de gestion de bases de données MySQL et un outil d'administration de MySQL : PhpMyAdmin.

0.5. Les composantes d'une application web

Un site est constitué, matériellement, d'un ordinateur connecté à l'Internet, et d'un programme tournant en permanence sur cet ordinateur, le serveur. Le programme serveur est en attente de requêtes transmises à son attention sur le réseau par un programme client. Quand une requête est reçue, le programme serveur l'analyse afin de déterminer quel est le document demandé, recherche ce document et le transmet au programme client.

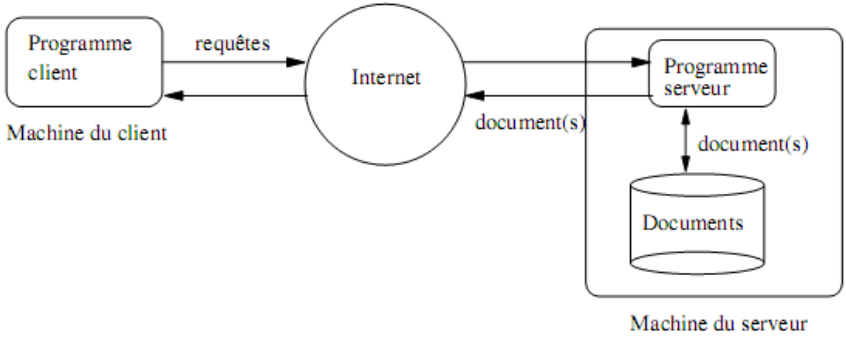


Fig. Architecture web

Généralement, le programme serveur est simplement désigné par le terme « serveur » ou par le nom du programme particulier (Apache, Tomcat, …). Les termes

« navigateur » et « client » sont également désignés tous deux par le programme client (Internet Explorer, Firefox, Safari, etc.). Enfin, le terme « utilisateur » (ou, parfois, « internaute »), est réservé à la personne physique qui utilise un programme client.

Un autre type important d'interaction consiste pour le programme client à demander au programme serveur d'exécuter un programme, en fonction de paramètres, et de lui transmettre le résultat.

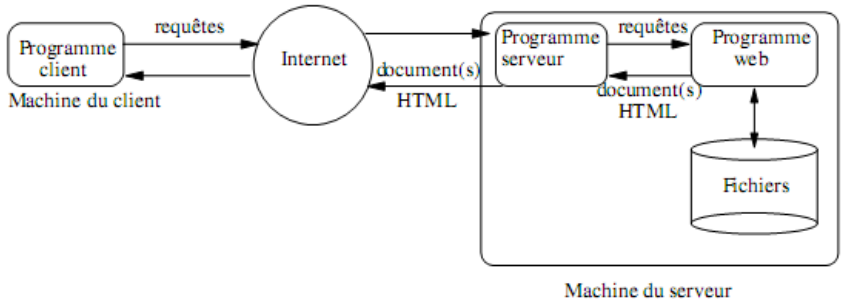


Fig. Architecture basique d'une application web

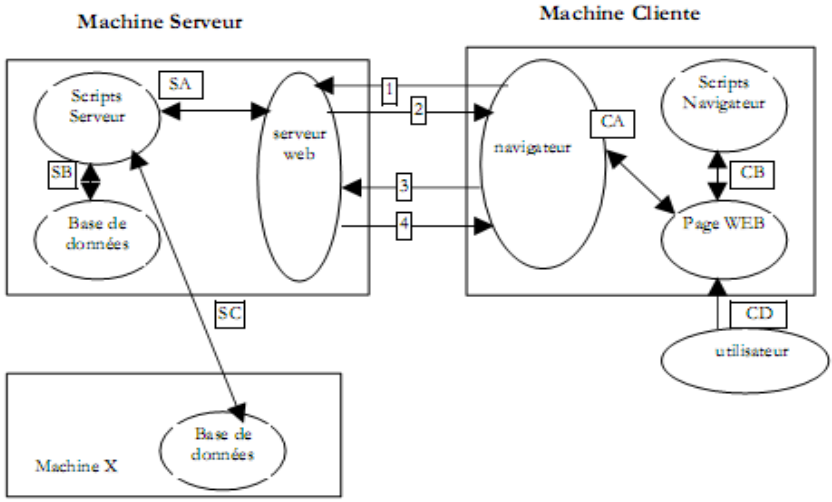
L'exécution du programme web par le serveur web se déroule en trois phases :

1. Constitution de la requête par le client : le navigateur construit une URL contenant le nom du programme à exécuter, accompagné, le plus souvent, de paramètres ;

2. Réception de la requête par le serveur : le programme serveur récupère les informations transmises par le navigateur et déclenche l'exécution du programme en lui fournissant les paramètres reçus ;

3. Transmission de la réponse : le programme renvoie le résultat de son exécution au serveur sous la forme d'un document HTML, le serveur se contentant alors de faire suivre au client.

0.6. Les échanges de données dans une application web avec formulaire



1 : Le navigateur demande une URL pour la première fois (http://machine/url). Aucun paramètre n'est passé.

2. le serveur web lui envoie la page web de cette url. Elle peut être statique ou bien dynamiquement générée par un script serveur (SA) qui a pu utiliser le contenu de bases de données (SB, SC). Ici, le script détectera que l'URL a demandée sans passage de paramètres et générera la page web initiale.

Le navigateur reçoit la page et l'affiche (CA). Des scripts côté navigateur (CB) ont pu modifier la page initiale envoyée par le serveur. Ensuite par des interactions entre utilisateur (CD) et les scripts (CB) la page web va être modifiée. Les formulaires vont notamment être remplis.

3. l'utilisateur valide les données de formulaire qui doivent alors être envoyées au serveur web. Le navigateur redemande l'URL initiale ou un autre selon les cas et transmet en même temps au serveur les valeurs du formulaire. Il peut utiliser pour ce faire deux méthodes appelées GET et POST. A réception de la demande du

client, le serveur déclenche le script (SA) associé à l'URL demandée, script qui va détecter les paramètres et les traiter.

4. le serveur délivre la page web construite par programme (SA, SB, SC). Cette étape est identique à l'étape 2 précédente. Les échanges se font désormais selon les étapes 2 et 3.

1. Introduction au Javascript

1.1. Les scripts côté navigateur

Les simples pages HTML (avec ou sans objets) ne fournissent des solutions aux besoins d'interactivité et de dynamisme des sites. Quand le visiteur demande une pareille page HTML en tapant une adresse URL, le serveur web lui renvoie simplement le contenu de la page demandée, c'est le navigateur web qui interprète le contenu HTML et les plug-ins qui interprètent les objets, en demandant à leur tour le contenu des objets au serveur web.

Si du code, côté client, est inséré dans la page, le serveur web renverra également la page au client, y compris le code. C'est donc, à nouveau le navigateur web qui doit interpréter le code et faire ce que le code demande.

La programmation côté client sera utilisée essentiellement dans le cas de validation de formulaires (champs obligatoires, petits calculs,...) et dans la création de petites animations (menus déroulants, ...).

La programmation côté client utilise des langages de scripts. À la différence d'autres langages de programmation, ces langages ne sont pas compilés, ils sont interprétés par le navigateur web du visiteur, et leur syntaxe est moins stricte que celle des langages de développeurs (variables non-typées a priori, le point-virgule à la fin des instructions est optionnel, ...).

Un script est une portion de code qui vient s'insérer dans une page HTML. Le code du script n'est toutefois pas visible dans la fenêtre du navigateur car il est compris entre des balises spécifiques qui signalent au navigateur qu'il s'agit d'un script écrit dans un langage spécifique.

Les langages de script côté client sont nombreux. En voici quelques uns :

- Javascript : qui est le plus utilisé. Il est reconnu par tous les navigateurs (dans les versions actuelles). Il se base sur le langage normalisé ECMAScript et possède une syntaxe issue du langage Java.
- JScript : qui est très proche du JavaScript. c'est simplement une adaptation de celui-ci par Microsoft.

- VBScript : qui est le langage de script développé par Microsoft sur base du Visual Basic. Ce langage n'est interprété que par les navigateurs basés sur ceux développés par Microsoft (soit Internet Explorer et Maxthon). Il est donc peu utilisé sur l'Internet (mais utilisé parfois sur des intranets).
- Perlscript : qui est développé sur base du langage Perl. Il n'est également interprété que par Internet Explorer et est peu utilisé.

1.2. Javascript dans les pages HTML

JavaScript a été initialement développé par Netscape et s'appelait à l'époque LiveScript. Adopté à la fin de l'année 1995, par la firme Sun (qui a aussi développé Java), il prit alors son nom actuel de JavaScript. Microsoft l'a aussi adopté à partir de son Internet Explorer 3.

Les scripts JavaScript sont gérés et exécutés en direct et sans retard par le navigateur lui-même sans devoir faire appel aux ressources du serveur. Par conséquent la page est chargée sur le navigateur avec le code source qui contient des scripts bruts de JavaScript en vue d'exécution. Il est alors évident que la confidentialité du code est « compromise » et il peut être copié et réutilisé par d'autres personnes.

On appelle souvent le langage JavaScript un langage événementiel. En effet, la plupart de ses scripts sont associés à des événements qui peuvent se produire sur le navigateur tel que le chargement de la page, la fermeture de la page, le clic, le survol, la sélection, la frappe au clavier… JavaScript est généralement utilisé pour contrôler les formulaires avant envoi au lieu d'attribuer ce travail à un langage coté serveur tel que le PHP.

La balise <script> informe le navigateur sur le début d'un script. Et pour préciser que c'est du JavaScript on ajoute l'attribut language="javascript".

La balise devient alors :

<script language="javascript">

La balise </script> informe de la fin du script JavaScript.

Il est toutefois possible de préciser au navigateur la version JavaScript à exécuter en ajoutant son identificateur.

Il y a plusieurs façons d'inclure du JavaScript dans une page HTML :

- Grâce à la balise <script>

- En mettant le code dans un fichier

- Grâce aux événements

Dans la balise script

Le code Javascript peut être inséré où on le désire dans la page Web, on doit toutefois veiller à ce que le navigateur est entièrement chargé le script avant d'exécuter une instruction. Ainsi, on place généralement le maximum d'éléments du script dans la balise d'en-tête (les balises <head> et </head>). Les évènements Javascript seront quant à eux placés dans le corps de la page (les balises <body> et </body>) comme attribut d'une commande HTML.

<script language="Javascript">

Le code de votre script

</script>

On informe ainsi au navigateur qu'on a affaire à un script en javascript.

Dans un fichier externe

Il est possible de mettre les codes de JavaScript en annexe dans un fichier. Le code à insérer est le suivant:

<script language=Javascript src="url/fichier.js"></script>

Où url/fichier.js correspond au chemin d'accès au fichier contenant le code en JavaScript, sachant que si celui-ci n'existe pas le navigateur exécutera le code inséré entre les 2 balises.

Grâce aux évènements

On appelle évènement une action de l'utilisateur, comme le clic d'un des boutons de la souris. Le code dans le cas du résultat d'un événement s'écrit:

<balise eventHandler="code Javascript à insérer">

eventHandler représente le nom de l'événement.

Un code javascript peut s'exécuter au chargement de la page (ex. affichage de la date courante, heure, …) ou après chargement de la page, lorsque l'utilisateur va interagir avec elle au moyen des différents objets qu'elle contient (liens, boutons, champs de texte,...), mais aussi par des actions sur l'environnement (déplacement de la fenêtre, activation d'une autre fenêtre, déplacement de la souris, frappe au clavier,...).

Exemple : Soit le code suivant qui permet l'affichage de la date et l'heure courantes du système.

```
<html>
<head><title>Date du jour</title></head>
<body>
<center>
<h1>Une page JavaScript générée dynamiquement</h1><hr/>
<h2><script language="JavaScript">
document.write("Bonjour, il est :<br/><br/>");
maintenant = new Date();
document.write(maintenant);
mois=maintenant.getMonth()+1;
document.write("<p>"+mois+"e mois");
</script></h2>
<h3>A chaque fois que vous rechargez la page, l'heure change</h3>
</body>
</html>
```

La fonction **new Date** retourne la date courante du système, tandis que la fonction **write** permet d'obtenir l'affichage dans le document ou page web.

Remarques

- JavaScript est un langage non typé. Il n'est donc pas utile de déclarer les variables qui vont être utilisées et encore moins d'indiquer leur type. En fait,

une variable est implicitement déclarée dès son apparition et typée par son contexte d'utilisation (affectation).

- Toute instruction se clôture par un point virgule.
- Il est également important de signaler que Javascript est sensible à la casse des caractères (différenciation entre les minuscules et majuscules, ex. Mois ≠ mois).Les méthodes toLowerCase() et toUpperCase() permettent de convertir tous les caractères d'une chaîne respectivement en minuscule et en majuscule.
- Une chaîne de caractères est présentée encadrée par des guillemets " ou en la délimitant par des apostrophes ' et en plaçant le tout entre parenthèses.
- Si vous souhaiter utiliser des guillemets dans vos chaînes de caractères, tapez \" ou \' pour les différencier vis à vis du compilateur.

1.3. Objets et leur hiérarchie

JavaScript repose sur les notions d'objets et de méthodes. Les objets sont classés hiérarchiquement, c'est-à-dire que pour accéder à un objet, il faut d'abord passer par l'objet parent.

Considérons l'exemple en HTML suivant :

```
<html>
<head></head>
<body>
<form name=''mon_form''>
<input type=''text'' name=''mon_text''>
</form>
</body>
</html>
```

Avec JavaScript, on ne peut pas accéder directement à la valeur du champ texte nommé « mon_text », car comme on peut le voir, on a la hiérarchie suivante :

- L'objet « text » se trouve dans l'objet « form » donc, pour arriver au texte on va écrire mon_form.mon_text (l'objet parent.l'objet cible).
- L'objet « form » se trouve dans l'objet « document ». L'objet document étant le corps du navigateur. Alors l'expression devient :

document.mon_form.mon_text

- L'objet « document » se troue dans l'objet « window ». L'objet window (ou fenêtre) est l'objet le plus haut en hiérarchie. C'est à partir de là que commence l'appel aux objets. L'expression précédente redevient:
  window.document.mon_form.mon_text

N.B. : Puisque le premier objet dans la hiérarchie est l'objet « window », alors on peut toujours ne pas le mentionner lors de l'appel hiérarchique aux objets. La dernière expression devient alors: document.mon_form.mon_text

On veut maintenant accéder à la valeur du champ texte, en mentionnant alors la méthode (ou la propriété) à la fin de la hiérarchie. On aura donc:

document.mon_form.mon_text.value

1.4. Quelques instructions de bases

Les fonctions d'entrée et de sortie

- La fonction **alert** est une méthode de l'objet window qui effectue une sortie d'expression à travers une boîte de message. Cette boite bloque le programme en cours tant que l'utilisateur n'aura pas cliqué sur "OK". Elle va aussi nous aider à débugger les scripts. Sa syntaxe est :

  Alert (expression) ;

  Où expression peut être un nombre, une chaîne de caractères ou une variable.

  Si vous souhaitez écrire sur plusieurs lignes, il faudra utiliser le signe \n.

- La fonction **prompt** est une autre méthode de l'objet window qui affiche une boîte d'entrée de données avec un texte dessus. Il est généralement utilisé pour saisir des données fournies par l'utilisateur. Sa syntaxe est :

  Rep=Prompt ("texte de la boite d'entrée") ;

  Où Rep est la variable qui reçoit la donnée saisie en entrée.

  On peut également proposer une valeur d'entrée par défaut. La syntaxe devient :

  Rep=Prompt ("texte de la boite d'entrée", "valeur par défaut") ;

En cliquant sur OK, la méthode renvoie dans Rep la valeur tapée par l'utilisateur ou la valeur par défaut. Si l'utilisateur clique sur Annuler ou Cancel, la valeur null est alors renvoyée.

N.B. : Le texte de la boite d'entrée ne doit pas dépasser 45 caractères sous Netscape et 38 sous Explorer 3.0.

Exemple : Soit le code suivant qui affiche la date courante du système à travers une boîte de dialogue, et demande à l'utilisateur d'entrer son nom avant de l'afficher.

```
<html>
<head><title>Exemple Boîtes de dialogue</title></head>
<body >
<h1>Utilisation des boîtes "alert et prompt" </h1><hr/>
<script language="Javascript">
date=new Date();
mois=date.getMonth()+1
alert("Nous sommes le "+ date.getDate()+" - "+ mois +" - "+ date.getYear());
rep=prompt('Tapez votre nom')
alert("Votre nom est : "+rep)
</script>
</body>
</html>
```

Nota : toute fois les boîtes de dialogues sont à éviter dans les pages web, car elles les rendent lourdes.

Exemple : Entrez une valeur quelconque dans la zone de texte d'entrée. Appuyer sur le bouton pour afficher cette valeur dans la zone de texte de sortie.

```
<html>
<head>
<script language="javascript">
```

```
function afficher(form2) {
var testin =document. form2.input.value;
document.form2.output.value=testin
}
</script>
</head>
<body>
<form name="form2">
Zone de texte d'entrée : <input type="text" name="input" value=""><br>
<input type="button" name="bouton" value="afficher"
onclick="afficher(form2)"><br>
Zone de texte de sortie : <input type="text" name="output" value="">
</form>
</body>
</html>
```

Grâce au gestionnaire d'évènement onclick, il est possible d'exécuter une fonction au click du bouton.

- La méthode **confirm** qui affiche une boite de dialogue de confirmation qui donne la possibilité de choisir entre 'OK' (renvoie la valeur true) ou 'Annuler' (renvoie la valeur false). Cette méthode est très utile pour confirmer les actions de l'utilisateur sur le navigateur. Sa syntaxe :

```
Rep=confirm ("texte de la boite d'entrée") ;
```

Une minuterie

Javascript met à votre disposition une minuterie (ou plus précisément un compteur à rebours) qui permettra de déclencher une fonction après un laps de temps déterminé.

La syntaxe de mise en route du temporisateur est :

nom_du_compteur = setTimeout("fonction_appelée()", temps en milliseconde)

Ainsi, setTimeout("demarrer()",5000) va lancer la fonction demarrer() après 5 secondes.

Pour arrêter le temporisateur avant l'expiration du délai fixé, il y a :

clearTimeout(nom_du_compteur)

Les Opérateurs

Les opérateurs sont des symboles qui permettent de manipuler des variables, c'est-à-dire effectuer des opérations, les évaluer, ... On distingue plusieurs types d'opérateurs : les opérateurs de calcul, les opérateurs d'assignation, les opérateurs d'incrémentation, les opérateurs de comparaison, les opérateurs logiques …

Les opérateurs de calcul : ils permettent de modifier mathématiquement la valeur d'une variable

Opérateur	Dénomination	Exemple	Résultat (avec X valant 7)
+	Addition	X + 2	9
-	soustraction	X - 2	5
*	multiplication	X * 3	21
/	Division	X / 2	3.5
=	affectation	X=3	Met la valeur 3 dans la variable X.

Les opérateurs d'assignation : ils permettent de simplifier des opérations telles qu'ajouter une valeur dans une variable et stocker le résultat dans la variable. Une telle opération s'écrirait habituellement de la façon suivante par exemple: x=x+2

Avec les opérateurs d'assignation il est possible d'écrire cette opération sous la forme suivante: x+=2 Ainsi, si la valeur de x était 7 avant opération, elle sera de 9 après...

Les autres opérateurs du même type sont les suivants:

Opérateur	Effet
- =	Soustrait deux valeurs et stocke le résultat dans la variable
* =	Multiplie deux valeurs et stocke le résultat dans la variable
/ =	Divise deux valeurs et stocke le résultat dans la variable

Les opérateurs d'incrémentation : ils permettent de facilement augmenter ou diminuer d'une unité une variable. Ces opérateurs sont très utiles pour des

structures telles que des boucles, qui ont besoin d'un compteur (variable qui augmente de un en un).

Un opérateur de type x++ permet de remplacer des notations lourdes telles que x=x+1 ou bien x+=1

Opérateur	Dénomination	Effet	Ex.	Résultat (avec X = 7)
++	incrémentation	Augmente d'une unité la variable	X++	8
- -	décrémentation	Diminue d'une unité la variable	X --	6

Les opérateurs de comparaison

Opérateur	Dénomination	Effet	Syntaxe	Résultat (avec X= 7)
==	Egalité	Vérifie que deux valeurs sont égales	X==3	Retourne 1 si X est égal à 3, sinon 0
<	Infériorité stricte	Vérifie qu'une variable est strictement inférieure à une valeur	X < 3	Retourne 1 si X est inférieur à 3, sinon 0
<=	Infériorité	Vérifie qu'une variable est inférieure ou égale à une valeur	X <= 3	Retourne 1 si X est inférieur ou égal à 3, sinon 0
>	Supériorité stricte	Vérifie qu'une variable est strictement supérieure à une valeur	X > 3	Retourne 1 si X est supérieur à 3, sinon 0
>=	Supériorité	Vérifie qu'une variable est supérieure ou égale à une valeur	X >= 3	Retourne 1 si X est supérieur ou égal à 3, sinon 0
= !	Différence	Vérifie qu'une variable est différente à une valeur	X = ! 3	Retourne 1 si X est égal à 3, sinon 0

Les opérateurs logiques (booléens) : ils permettent de vérifier si plusieurs conditions sont vraies :

Opérateur	Dénomination	Effet	Syntaxe
||	Ou logique	Vérifie qu'une condition est réalisée	((condition1)||(condition2))
&&	Et logique	Vérifie que toutes les conditions sont réalisées	((condition1)&&(condition2))
!	Non logique	Inverse l'état d'une variable booléenne (retourne la valeur 1 si la valeur vaut 0, 0 si elle vaut 1)	(!condition1)

Les structures conditionnelles

Ce sont des instructions qui permettent de tester si une condition est vraie ou non, ce qui permet de donner de l'interactivité aux scripts.

Remarque : la condition doit être entre des parenthèses et il est possible de définir plusieurs conditions à remplir avec les opérateurs ET et OU (&& et ||).

L'instruction if ... else

Cette forme permet de choisir les instructions à exécuter en cas de réalisation ou de non réalisation de la condition. Sa syntaxe est:

```
if (condition) {
liste d'instructions si vrai
}
else {
liste d'instructions si faux
}
```

Si la condition est vérifiée (true), la liste d'instructions si vrai s'exécute. Si elle ne l'est pas (false), la liste d'instructions si faux s'exécute.

La version simplifiée :

Il est possible de faire un test avec une structure beaucoup moins lourde grâce à la structure suivante:

(condition) ? instruction si vrai : instruction si faux

Remarque : la condition doit être entre des parenthèses ; lorsque la condition est vraie, l'instruction de gauche est exécutée, et lorsque la condition est fausse, l'instruction de droite est exécutée.

Dans sa formulation la plus simple, l'expression if se présente comme suit :

if (condition vraie) {

une ou plusieurs instructions;

}

Ainsi, si la condition est vérifiée, les instructions s'exécutent. Si elle ne l'est pas, les instructions ne s'exécutent pas et le programme passe à la commande suivant l'accolade de fermeture.

Exemple 1 :

```
<html><head><title>Saisie d'age</title></head>
<body>
<h1>Utilisation de la structure if then </h1><hr/>
<script language="javascript">
var age = prompt("Entrer votre age");
if ( age <= 0 || age > 150)
alert("Age non valide saisir à nouveau");
else
alert("vous avez saisi "+parseInt(age) + " votre age est valide");
</script>
</body>
```

```
</html>
```

Exemple 2 : Choix à l'aide des boutons radio.

```
<html>
<head>
<script language="javascript">
function choixprop(form3) {
if (form3.choix[0].checked) { alert("vous avez choisi la proposition " + form3.choix[0].value) };
if (form3.choix[1].checked) { alert("vous avez choisi la proposition " + form3.choix[1].value) };
if (form3.choix[2].checked) { alert("vous avez choisi la proposition " + form3.choix[2].value) };
}
</script>
</head>
<body>
entrez votre choix :
<form name="form3">
<input type="radio" name="choix" value="1">Choix numéro 1<br>
<input type="radio" name="choix" value="2">Choix numéro 2<br>
<input type="radio" name="choix" value="3">Choix numéro 3<br>
<input type="button" name="but" value="Quel et votre choix ?"
onclick="choixprop(form3)">
</form>
</body>
</html>
```

Exemple 3 : Affichage heure.

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

```
<script language="javascript">
<!--
function getdt(){
dt=new date();
hrs=dt.gethours();
min=dt.getminutes();
sec=dt.getseconds();
tm=" "+hrs+":";
tm+=min+":";
tm+=sec+" heure de Kinshasa";
document.horloge.display.value=tm;
settimeout("getdt()",1000);
}
// -->
</script>
</head>
<body onload="getdt()">
<form name="horloge">
<input type="text" name="display" size=25 value ="">
</form>
</body>
</html>
```

L'instruction switch

Elle permet d'effectuer un choix entre plusieurs en comparant la valeur test aux valeurs possibles. En cas de concordance avec un cas, les instructions associées à ce cas sont exécutées. Sa syntaxe est :

```
switch(valtest)
```

```
   {  case 1 : instruction; break;
      case 2 : instruction;    break;
…
   };
```

Exemple : Entrée par formulaire : choix dans un listbox

```
<html>
<head><title>Faire un choix</title>
<script language="JavaScript">
function Preferer(x)
{ switch(x)
{  case 1 : alert('Votre choix : '+'classique');break;
      case 2 : alert('Votre choix : '+'jazz');    break;
      case 3 : alert('Votre choix : '+'rock');    break;
      case 4 : alert('Votre choix : '+'pop');     break;
};
};
</script>
</head>
<body>
<form name="musique">
choisissez : <select name="choix"
 onChange="Preferer(document.musique.choix.selectedIndex)">
<option value = "choix">
<option value = "classique">classique
<option value = "jazz">jazz
<option value = "rock">rock
<option value = "pop ">pop
```

</select>

</form>

</body>

</html>

Les boucles

Les boucles sont des structures qui permettent d'exécuter plusieurs fois la même série d'instructions jusqu'à ce qu'une condition soit ou ne soit plus réalisée.

La façon la plus commune de faire une boucle, est de créer un compteur (une variable qui s'incrémente, c'est-à-dire qui augmente de 1 ou autre nombre à chaque tour de boucle) et de faire arrêter la boucle lorsque le compteur dépasse une valeur limite.

La boucle for

Elle permet d'exécuter plusieurs fois la même série d'instructions en fonction de la réalisation d'un certain critère. Sa syntaxe est :

for (valeur initiale du compteur; condition; progression du compteur) {

liste d'instructions

}

Par exemple:

for (i=1; i<6; i++) {

Alert(i)

}

Cette boucle affiche 5 fois la valeur de i, c'est-à-dire 1, 2, 3, 4, 5.

Elle commence à i=1, vérifie que i est bien inférieur à 6, puis progresse de 1 jusqu'à atteindre la valeur i=6, pour laquelle la condition ne sera plus réalisée, la boucle s'interrompra et le programme continuera son cours après l'accolade de fermeture.

Nota : il faudra toujours vérifier que la boucle a bien une condition de sortie (i.e le compteur s'incrémente correctement)

L'instruction while

Elle représente un autre moyen d'exécuter plusieurs fois la même série d'instructions tant que la condition est réalisée. Sa syntaxe est :

```
while (condition) {
liste d'instructions
 }
```

Attention, la condition de sortie pouvant être n'importe quelle structure conditionnelle, les risques de boucle infinie (boucle dont la condition est toujours vraie) sont grands, c'est-à-dire qu'elle risque de provoquer un plantage du navigateur!

Arrêt inconditionnel

L'instruction break permet d'interrompre prématurément une boucle for ou while.

Pour illustrer ceci, voyons cet exemple :

```
compt=1;
while (compt<5) {
if (compt == 4)
break;
document.write ("ligne : " + compt + "<br>");
compt++;
}
document.write("fin de la boucle");
```

La boucle while sera interrompue par le break, une fois que le compteur aura atteint la valeur 4. A ce moment, "fin de boucle" sera affiché.

Saut inconditionnel

Il peut être nécessaire de faire sauter à la boucle une ou plusieurs valeurs sans pour autant mettre fin à celle-ci.

La syntaxe de cette expression est "**continue**;" (cette instruction se place dans une boucle!), on l'associe généralement à une structure conditionnelle, sinon les lignes situées entre cette instruction et la fin de la boucle seraient obsolètes.

Exemple: Imaginons que l'on veuille imprimer pour x allant de 1 à 6, la valeur de 1/(x-4).Il est évident que pour x=4, il y aura une erreur.

```
<html><head><title>Boucle tant que</title>
</head>
<body>
<center><h1>Exemple d'une boucle</h1></center><hr>
<p>Vous allez gérer la division par 0 quand boucle atteindra la valeur 4</p>
<script language="JavaScript">
x=1
while (x<=6) {
if (x == 4) {
alert('division par 0')
x++;
continue;
}
a = 1/(x-4);
alert(x)
x++
}
</script>
</body>
</html>
```

Grâce à l'instruction continue il est possible de traiter la valeur x=4 à part puis de continuer la boucle.

En lieu et place de continue, l'instruction **break** arrête la boucle.

La fonction (sous-programme)

On appelle fonction un sous-programme qui permet d'effectuer un ensemble d'instructions par simple appel de la fonction dans le corps du programme principal.

La déclaration d'une fonction

Avant d'être utilisée, une fonction doit être définie car pour l'appeler dans le corps du programme il faut que le navigateur la connaisse, c'est-à-dire qu'il connaisse son nom, ses arguments et les instructions qu'elle contient. La définition d'une fonction s'appelle "déclaration". La déclaration d'une fonction se fait grâce au mot clé function selon la syntaxe suivante:

```
function Nom_De_La_Fonction(argument1, argument2, ...) {
liste d'instructions
}
```

Remarques:

- Un nom de fonction doit commencer par une lettre ; peut comporter des lettres, des chiffres et est sensible à la casse (différenciation entre les minuscules et majuscules)
- Les arguments sont facultatifs, mais s'il n'y a pas d'arguments, les parenthèses doivent rester présentes

Appel de fonction

Pour exécuter une fonction, il suffit de faire appel à elle en écrivant son nom (une fois de plus en respectant la casse) suivi d'une parenthèse ouverte (éventuellement des arguments) puis d'une parenthèse fermée:

```
Nom_De_La_Fonction();
```

Remarque:

Si jamais vous avez défini des arguments dans la déclaration de la fonction, il faudra veiller à les inclure lors de l'appel de la fonction (le même nombre d'arguments séparés par des virgules)

Pour éviter d'erreurs de référence, toute fonction doit être déclarée avant d'être appelée, et cela se fait généralement dans les balises <script> et </script>situées dans l'en-tête de la page.

Grâce au gestionnaire d'évènement onLoad (à placer dans la balise body), il est possible d'exécuter une fonction au chargement de la page, comme par exemple l'initialisation des variables pour votre script, et/ou le test du navigateur pour savoir si celui-ci est apte à faire fonctionner le script. Il s'utilise de la manière suivante:

```
<html>
<head>
<script language="Javascript">
function Chargement() {
alert('Bienvenue sur le site');
}
</script>
</head>
<body onLoad="Chargement();" >
Javascript qui ne sert absolument à rien si ce n'est déranger vos visiteurs...
</body>
</html>
```

Les paramètres d'une fonction

Il est possible de passer des paramètres à une fonction, c'est-à-dire lui fournir une valeur ou le nom d'une variable afin que la fonction puisse effectuer des opérations sur ces paramètres ou bien grâce à ces paramètres. Lorsque vous passez plusieurs paramètres à une fonction, il faut les séparer par des virgules, aussi bien dans la déclaration que dans l'appel et il faudra veiller à bien passer le bon nombre de paramètres lors de l'appel au risque sinon de créer une erreur.

Imaginons que l'on veuille créer une page web qui affiche une boîte de dialogue avec un texte différent selon le lien sur lequel on appuie. La méthode de base consiste à faire une fonction pour chaque texte à afficher:

```
<html>
<head>
<script language="Javascript">
function Affiche1() {
 alert('Clique sur Texte 1');
 }
function Affiche2() {
alert('Clique sur Texte2');
}
</script>
</head>
<body onLoad="Chargement();" >
<a href="javascript:;" onClick="Affiche1();">Texte1</a><br/>
<a href="javascript:;" onClick="Affiche2();">Texte2</a>
</body>
</html>
```

Il existe toutefois une méthode plus "confortable" qui consiste à créer une fonction qui a comme paramètre le texte à afficher:

```
<html>
<head>
<script language="Javascript">
function Affiche(Texte) {
alert(Texte);
}
```

```
</script>
</head>
<body onLoad="Chargement();" >
<a href="javascript:;" onClick="Affiche('Clique sur Texte1');">Texte1</a>
<a href="javascript:;" onClick="Affiche('Clique sur Texte2');">Texte2</a>
</body>
</html>
```

Exemple d'un lien qui ouvre d'une fenêtre dans laquelle s'affiche une image :

```
<html>
<head>
<title>nouvelle fenetre</title>
<script>
  function creerFenImage() {
    fiRef = window.open ("","fenImage", "width=290, height=200, scrollbars=no,
    toolbar=no, location=no, directories=no, status=no")
  }
</script>
<body>
<a href="JD.jpg" target="fenImage"onClick="creerFenImage()">
<b>cliquer pour ouvrir une nouvelle fenêtre </b></a>
</body>
</html>
```

Exemple de gestion d'un envoi de mail :

```
<html>
<head><title>Envoyer un mail</title></head>
<body>
<script>
function envoyer() {
```

```
  var text = "";
  var index = 0;
  var  identif = "monweb"
// création du sujet du mail qui contient l'identificateur (identif) et l'option
// selected(probleme, suggestion, commentaire)
  with (window.document.mail) {
    i = elements["sujet"].length - 1;
    for (;i >= 0; i--){
      if (elements["sujet"][i].selected == true) {
         index = i;
         break;
      }
    }
    // génération de l'action du formulaire mailto
    action = 'mailto:mathieu.decore@altavista.net?content-
type=text/html&subject=';
    action += identif + '(' + elements["sujet"][index].value + ')';
  }
  // création du corps du mail, récupération du type de browser utilisé
  // par la personne qui émet le mail
with (navigator) {
    text = appname +' ' + appversion +'\n';
text += appcodename + ' depuis' +  useragent + ' \n';
  }
  with (window.document.mail) {
 // l' information sur le browser est stocke dans un champ hidden  (browser)
    elements["browser"].value = text;
    // on régénère le texte saisie + le message "message envoyé"
    elements["message"].value += '\n\n____ message envoye ___\n\n';
}
  return true;
```

```
}
</script>
<form name="mail"  action="" method="get"  enctype="text/plain"
onsubmit="envoyer()">
<b>sujet du mail : </b><br>
<select name="sujet">
<option value="probleme">problème
<option value="suggestion"> suggestion
<option value="commentaire" selected> commentaire
</select>
<hr>
<p>
<b>message :</b><br>
<textarea name="message" rows="15" cols="50" wrap></textarea>
<input type="hidden" name="browser">
<hr>
<center>
<input type="submit" value="envoyer le message">
</center>
</form>
</body>
</html>
```

L'emploi de this

Lorsque vous faîtes appel à une fonction à partir d'un objet, par exemple un formulaire, le mot-clé this fait référence à l'objet en cours et vous évite d'avoir à définir l'objet en tapant window.objet1.objet2... ainsi lorsque l'on passe l'objet en cours en paramètre d'une fonction, il suffit de taper nom_de_la_fonction(this) pour pouvoir manipuler cet objet à partir de la fonction.

Pour manipuler les propriétés de l'objet il suffira de taper this.propriété (où propriété représente bien sûr le nom de la propriété).

Exemple d'utilisation du mot-clé this dans un formulaire :

```
<form name="form3">
<input type="radio" name="choix" value="1">Choix numéro 1<br>
<input type="radio" name="choix" value="2">Choix numéro 2<br>
<input type="radio" name="choix" value="3">Choix numéro 3<br>
<input type="button" name="but" value="Quel et votre choix ?" onClick="choixprop(form3)">
</form>
```

Au lieu d'employer choixprop(form3), on aurait pu utiliser choixprop(this.form) et éviter ainsi toute confusion avec les autres noms de formulaires. Dans cet exemple, this.form désigne le formulaire form3 complet. Par contre, choixprop(this) n'aurait désigné que l'élément de type bouton du formulaire form3.

Pour être complet, this est utilisé aussi pour créer une ou plusieurs propriétés d'un objet. Ainsi, pour créer un objet livre avec les propriétés auteur, éditeur et prix, cette opération peut être effectuée à l'aide de la fonction :

```
function livre(auteur, editeur, prix) {
this.auteur = auteur;
this.editeur = editeur;
this. prix = prix;
}
```

Exemple : Test de validité des données d'un formulaire avant leur envoie.

```
<html>
<head><title>Formulaire web</title>
<script language="javascript">
function effacer(){
document.frmpersonne.txtNom.value="";
document.frmpersonne.txtAge.value="";
}
```

```
function valeur(){
with (document.frmpersonne){
champs=/^\s*(\d+)\s*$/.exec(txtAge.value);
if (champs == null){
alert('Age incorrect');
txtAge.focus();
return;
}
txtAge.value=champs[1];
submit();
}
}
</script>
</head><body><center>
<h1>Un formulaire Web</h1>
<h2>Récupération des valeurs des champs par un script Javascript côté navigateur</h2>
<hr>
<form name="frmpersonne" method="post">
<br>
<table>
<tr><td>Nom</td>
<td><input type="text" value="" name="txtNom"></td>
</tr><tr><td>Age</td>
<td><input type="text" value="" name="txtAge"></td>
</tr><tr></tr>
```

```
<tr><td colspan=2><input type="submit" value="Envoyer" name="Effacer"
onclick="valeur();"></td>
<td><input type="button" value="Effacer" name="Effacer"
onclick="effacer();"></td>
</tr>
</table></form>
</body>
</html>
```

1.5. Exercices

1. Proposer une page qui permet la lecture de quatre valeurs puis calcule et affiche leur moyenne.

2. Modifier le formulaire précédent en exigeant que l'âge soit supérieur à 15 pour que l'envoie soit validé.

2. Programmation web en PHP

2.1. Introduction

PHP (officiellement: Hypertext Preprocessor, et au départ Personal Home Page) est un langage de scripts généraliste, spécialement conçu pour le développement d'applications web (pages dynamiques). Comme particularités :

- Il s'intègre facilement au HTML.
- Il s'exécute sur le serveur et permet d'accéder facilement aux nombreuses bases de données (dBase, Ingres, mSQL, Oracle, MySQL, …).
- C'est un produit "Open Source" (le code est accessible à tout développeur) et est gratuit.

Au delà des pages statiques que l'on crée avec le (X)HTML, le PHP peut se mettre dans du code XHTML pour rendre celui-ci dynamique

Le traitement côté serveur par PHP apporte comme intérêts :

- Réduction du temps de téléchargement puisque le client ne reçoit qu'une simple page HTML (=>diminution du trafic réseau) ;
- Absence de problème de compatibilité des navigateurs ;
- Offrir au client des données qui sont dans une base ;
- Et le code ne peut être vu par le client.

Combiné au système d'exploitation Linux, au serveur Apache et à la base de données MySQL (eux-mêmes gratuits), il permet de créer des sites Web à des coûts très réduits.

Ce qui distingue PHP des langages de script comme le Javascript est que le code est exécuté sur le serveur. Avec un script similaire sur votre serveur, le client ne reçoit que le résultat du script, sans aucun moyen d'avoir accès au code qui a produit ce résultat.

PHP est principalement conçu pour servir de langage de script coté serveur, ce qui fait qu'il est capable de réaliser tout ce qu'un script CGI quelconque peut faire, comme collecter des données de formulaire, générer du contenu dynamique, ou gérer des cookies. Mais PHP peut en faire bien plus :

- Langage de programmation en ligne de commande. Vous pouvez écrire des scripts PHP et l'exécuter en ligne de commande, sans l'aide du serveur web et d'un navigateur. Il vous suffit de disposer de l'exécutable PHP.
- Ecrire des applications clientes graphiques. Bien qu'il ne soit pas le meilleur langage pour écrire des applications clientes graphiques.

Pour l'édition des codes en PHP, on utilise soit des éditeurs simples comme Note pad ou Bloc notes, ou encore des outils spécialisés comme PHPEdit et PHPEclipse, puis l'enregistrement se fait avec l'extension php, php3, php4…

Contrairement à une page (X)HTML dont la navigateur interprète directement, le serveur doit d'abord traiter la page en PHP avant de l'envoyer.

4.1. Installation et Configuration du PHP

Le principal inconvénient du développement de site en PHP, c'est que les scripts ne sont interprétés que par un serveur. En effet, il est impossible de voir le résultat de ses lignes de code avant d'avoir mis ses pages en ligne. Pour remédier à cela, on installe un serveur de développement complet sur sa machine.

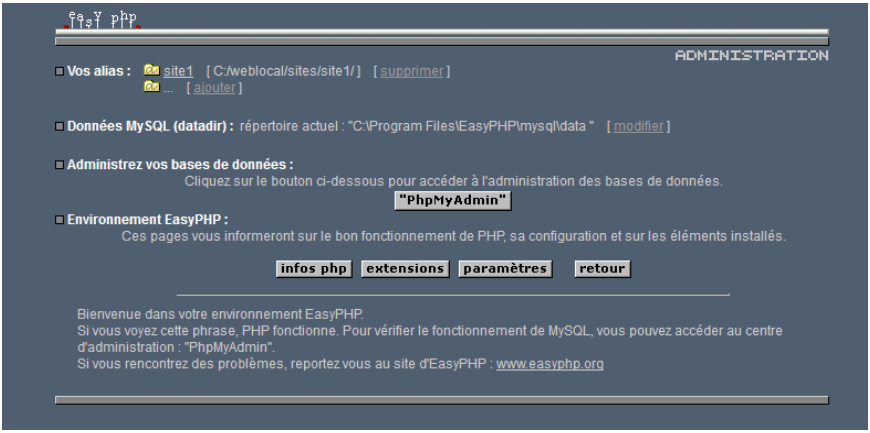
L'installation d'un serveur de développement exige du matériel : un serveur Apache, un certain nombre de bibliothèques PHP, une base de données MySQL et quelques autres outils qui sont actuellement organisés en kits d'installation et de configuration en quelques clics d'un serveur de développement complet. Certains d'entre eux, EasyPHP et Wampserver, contiennent l'ensemble des outils nécessaires au développement de pages PHP. Ils s'installent très facilement sur un ordinateur fonctionnant sous Windows.

Nous présentons dans cette partie l'EasyPHP et dans l'étude des cas, nous faisons recours au Wampserver. Comme vous allez le constater, les deux outils sont presque similaires.

Pour mettre en place d'EasyPHP, exécutez simplement le programme d'installation et choisissez le dossier dans lequel vous souhaitez le placer.

Il suffit d'exécuter EasyPHP à partir du menu Démarrer, et le serveur Apache se met en route. Une petite icône dans la zone de notification en témoigne. Et pour lancer l'interface d'administration d'EasyPHP, faites un clic droit sur cette

petite icône, et sélectionnez administration dans le menu qui apparaît. Une nouvelle fenêtre de votre navigateur par défaut s'ouvre sur l'interface d'EasyPHP.



Elle vous permet d'effectuer plusieurs manipulations : Gérer l'ensemble de vos projets PHP, Administrer votre base de données locale, Vérifier la version de PHP que vous possédez et les extensions qui y sont installées, Connaître les paramètres d'accès à votre base de données et Tester différentes extensions et fonctionnalités présentes.

La gestion de vos projets se fait, à travers la fenêtre qui s'ouvre en cliquant sur le bouton [Ajouter] de l'option vos alias.

Saisissez un nom pour l'alias (nom de votre site, ex. Urkim) et le chemin du répertoire créé (ex. C:\weblocal\sites\sites1).Cliquez sur valider pour terminer.

Il ne reste qu'à redémarrer EasyPHP pour qu'il prenne en charge votre nouveau site, faute de quoi vous aurez un message d'erreur 404 ('impossible d'afficher la page).

Revenez à l'interface d'administration d'EasyPHP, où avec un clic sur le nom de votre alias, vous visualisez votre site avec ses différentes pages.

4.2. Les bases du langage PHP

PHP ressemble aux langages C, C++ et Javascript, à savoir :

- ; est obligatoire à la fin de chaque ligne d'instructions ;
- {…} pour encadrer un bloc d'instructions ;
- les opérateurs de comparaison et d'affectation sont les mêmes (&&, ||, ==, …) ;
- les symboles des commentaires sont // et /* … */
- il est sensible à la casse (ex. la variable $NBRE, $nbre et $Nbre ne sont pas les mêmes).

Les entrées / sorties

Les saisies de données (entrée) vont être réalisées par l'intermédiaire des formulaires, et les éditions des résultats (sorties) vont s'inscrire dans la page par les fonctions suivantes :

- **echo** qui la fonction d'affichage la plus utilisée en PHP.

Exemple : echo ''Bonjour'' // Affiche Bonjour

Pour afficher un guillemet dans un texte, il faut le faire précéder par un backslash (soit \''). Et les autres caractères s'obtiennent à l'aide de : \' l'apostrophe, \t tabulation, \n alinéa ou retour chariot.

- **print()** qui est équivalent à echo.
  Exemple : print(''Bonjour \''Bruna \''') // Affiche Bonjour
- **printf()** qui donne un affichage formaté comme en C.
  Exemple : printf(''Junior %s'', ''Jude'') // Affiche Junior Jude

Intégration des scripts PHP dans une page HTML

Les pages web sont au format html, tandis que les pages web dynamiques générées avec PHP sont au format php (index.php par exemple). Le code source php est directement inséré dans le fichier HTML grâce aux balises suivantes :

- <?php …?>(balise standard)
- <? …?> (balise concise)
- <script language=''PHP''> … </script> (balise de script)

Exemple d'une page nommée index.php:

```
<html>
<head>
<title>Notre première instruction : echo</title>
</head>
<body>
<h2>Affichage de texte avec PHP</h2>
<p>
        Cette ligne a été écrite entièrement en (x)HTML.<br />
<?php echo("Celle-ci a été écrite entièrement en PHP.<br>");
                Printf("MWAMBA NZAMBI %s","Junior Jude");
?>
        </p>
</body>
</html>
```

Les variables

En PHP le typage des variables est implicite (ce n'est pas un langage très typé). Il n'est donc pas nécessaire de déclarer les types ni d'initialiser les variables. Il est même possible d'affecter à une variable des valeurs de types différents dans le même script.

Les identificateurs de variables sont précédées par le symbole « $ »

Exemple:

$nom=‘‘MUJINGA’’;

$c=$a+$b;

Les variables peuvent être de type entier (integer), réel (double), chaîne de caractères (string), tableau (array), booléen (boolean).

- Chaîne de caractères : la valeur est délimitée par des guillemets (simples ou doubles, on privilégie les simples pour question de rapidité) :

```
<?php
    $prenom = 'Anthony';
?>
```

- Un nombre : il peut s’agir d’un nombre entier ou d’un nombre réel (flottant) :

```
<?php
    $nombre = 7;
    $nombre = 3.02;
?>
```

- Un booléen : les variables booléennes peuvent prendre deux valeurs : la valeur true (vrai) et la valeur false (faux). Elles peuvent être utiles pour savoir par exemple si une option est activée ou pas :

```
<?php
    $afficher_options = false; //on n'affichera pas les options
    $utiliser_cache   = true;  //on utilisera un cache
?>
```

Il est possible de convertir le type d’une variable, settype($var, ’’type’’) : convertit la variable d’un type à un autre.

Exemple:

$str = ’’12’’; // $str vaut la chaîne ‘‘12’’

$nbr = (int)$str; // $nbr vaut le nombre 12

N.B. : Les constantes PHP ne sont préfixées par aucuns symboles. Elles sont déclarées de la façon suivante :

define(‘‘id_constante’’,‘‘valeur’’);

// ou bien define(id_constante, ‘‘valeur’’);

Exemple : define(‘‘nom’’, ‘‘UNIKIN’’); // la constante nom vaut la valeur UNIKIN

Les constantes ne sont définie qu'une seule fois dans le script (leurs valeurs restent inchangées et leurs identificateurs ne sont pas sensibles à la casse, contrairement aux variables).

Les opérateurs

Les opérateurs arithmétiques sont :+ (addition), - (soustraction), * (multiplié), /(divisé), %(modulo), ++ (incrément) et --(décrément).

Les opérateurs d'assignement := (affectation), *= ($x*=$y équivalent à $x=$x*$y), de même pour /=, +=, -=, %=

Les opérateurs logiques : and, && (et), or, || (ou), xor (ou exclusif) et ! (non)

Et les opérateurs de comparaison :== (égalité), < (inférieur strict), <= (inférieur large), >, >= et != (différence)

La concaténation de chaînes

L'opérateur de concaténation de chaînes est « . », et se présente sous 2 formes :

Exemple 1 :

```
$var1 = ''Bonjour'';
$var2 = '' à tous'';
echo $var1.$var2; // imprime Bonjour à tous
```

Exemple 2 :

```
$var = 'Ca va bien';
$var.= '' monsieur''
echo $var; // affiche Ca va bien monsieur
```

Les structures de contrôle

La syntaxe des structures de contrôle est la même que celle en langage C.

Structure conditionnelle si … alors … [sinon …]

```
if( condition1 )
{
traitement 1
}
```

```
elseif(condition 2-1)
{
traitement 2-1
}
else
{
traitement 2-2
}
```

Sélection de cas (branchement multiple)

```
switch( paramètre_index )
{
case valeur 1 : { traitement 1 } break;
case valeur 2 : { traitement 2 } break;
…
default : { traitement par défaut }
}
```

Structure de boucle en nombre défini

```
for( $i=$n ; $i<$m ; $i++ )
{
traitement
}
```

Où $i le compteur, reçoit sa valeur initiale $n ; $i<$m est la condition de sortie avec $m la limite à ne pas dépasser ; $i++ est l'expression de progression du compteur.

Structure de boucle en nombre non défini

- Tant que …

```
while( condition )
{
```

traitement

}

- Faire … Tant que …

do

{

traitement

}

while(condition)

N.B. : L'instruction **break** permet de quitter prématurément une boucle. Et L'instruction continue permet d'ignorer le traitement associé à une valeur de la boucle et de passer à l'occurrence suivante d'une boucle.

Exemple 1 :

```
while($nbr < 10) {
echo $nbr.''<br>'';
if($nbr == 5)
break;
$nbr++;
}
```

Exemple 2 :

```
for($i=1; $i<=10; $i++) {
if($i==5)
continue;
echo $i;
}
```

La valeur 5 ne sera pas affichée.

Les tableaux

La fonction **array()** permet de créer des tableaux dynamiques qui seront exploités par les scripts PHP. Un tableau array est temporaire et ne reste généré que le temps du déroulement du script.

Les tableaux array servent souvent à stoker les données provenant d'une base de données en attendant le traitement. Elles permettent aussi aux fonctions de retourner plusieurs résultats au lieu d'une seule.

Remarque : Les tableaux dynamiques ou arrays n'ont rien à voir avec les tableaux HTML, qui servent à la mise en forme des données de la page Web.

Pour initialiser un tableau on utilise plusieurs méthodes:

Méthode 1: (classique)

$tableau= array($val1,$val2,$val3,…);

Exemple:

$tab=array(''URKim'',2012,'Kinshasa');

dans ce cas $tab[0]=''URKim'', $tab[1]=2013 et $tab[2]='Kinshasa'

Méthode 2: (initialisation directe)

$tableau[0]=$val0;

$tableau[1]=$val1;

$tableau[10]=$val10; …

Méthode 3: (initialisation directe implicite)

$tableau[]=$val0; (sous-entend $tableau[0]=$val0)

$tableau[]=$val1; (sous-entend $tableau[1]=$val1)

$tableau[]=$val2; …

Ainsi, l'appel d'un élément du tableau se fait à l'aide de son indice.

Exemple:Echo $tab[0];

Parcours d'un tableau

La première méthode pour parcourir un tableau consiste à utiliser une boucle tant que.

```
$i=0;
while($i <= count($tab)) // la fonction count() retourne le nombre d'éléments
{
echo $tab[$i].''<br>'';
$i++;
}
```

La seconde la plus simple consiste à l'utilisation de la boucle foreach, dont la syntaxe est la suivante:

```
foreach($tableau as $element)
{
traitement;
}
```

La variable $element prend pour valeurs successives tous les éléments du tableau nommé $tableau.

Exemple:

```
foreach($tab as $elem)
{
echo $elem.''<br>'';
}
```

Quelques fonctions manipulant les tableaux

count(), sizeof() : retournent le nombre d'éléments du tableau ;

in_array($var,$tab) : dit si la valeur de $var existe dans le tableau $tab ;

list($var1,$var2…) : transforme un tableau en liste de variables ;

range($i,$j) : retourne un tableau contenant un intervalle de valeurs ;

shuffle($tab) : mélange les éléments d'un tableau ;

sort($tab) : trie alphanumérique les éléments du tableau ;

rsort($tab) : trie alphanumérique inverse les éléments du tableau ;

implode($str,$tab), join : retournent une chaîne de caractères contenant les éléments du tableau $tab joints par la chaîne de jointure $str ;

array_merge($tab1,$tab2,$tab3…) : concatène les tableaux passés en arguments ;

array_rand($tab) : retourne un élément du tableau au hasard ;

reset($tab) : place le pointeur sur le premier élément ;

current($tab) : retourne la valeur de l'élément courant ;

next($tab) : place le pointeur sur l'élément suivant ;

prev($tab) : place le pointeur sur l'élément précédant ;

each($tab) : retourne la paire clé/valeur courante et avance le pointeur.

Exemple :

```
$colors = array(''red'', ''green'', ''blue'');
$nbr = count($colors);
reset($colors);
for($i=1; $i<=$nbr; $i++) {
echo current($colors).''<br>'';
next($colors);
}
```

Les Fonctions

Définitions

- Les fonctions sont des sous-programmes qui regroupent un ensemble d'instructions réutilisables simplement et qui accomplissent un ensemble d'opérations par appel dans les corps de programmes principaux. Elles peuvent accepter des valeurs (appelées "arguments" ou "paramètres"). S'il y en a plusieurs, les arguments sont séparés par des virgules.

Une fonction réalise une succession d'instructions et renvoie principalement une (et une seule !) valeur issue d'un calcul à l'aide de l'instruction return, et spécialement en PHP, elle peut être formulée comme une procédure (c'est-à-dire elle réalise une succession d'instructions mais ne renvoie pas de valeur en retour).

La syntaxe est la suivante:

```
function ma_fonction($argument1, $argument2,…)
{
Liste des instructions;
Return ($valeur_de_retour);
}
```

N.B. : l'instruction return est omise en cas de sa formulation comme procédure.

Exemple 1 : une fonction qui calcule le prix total taxe comprise d'un produit à partir d'un argument passé, le prix total hors taxe.

```
Function PT_taxe_comprise ($Ptht) {
$Pttc=0.16 * $PTHT;
return $Pttc;      // ou return ($Pttc);
 }
```

Exemple 2 : une fonction qui calcule le prix total hors taxe d'un produit à partir de deux arguments passés, le prix unitaire et la quantité.

```
Function PT_hors_taxe ($Pu, $Qte) {
$Ptht= $Pu*$Qte;
return $Ptht;
}
```

Exemple 3 : Une fonction définie comme procédure à 3 arguments qui effectue l'affichage des prix totaux.

```
function affichage_prix ($s_nom_article, $Ptht, $Pttc) {
echo ("<b><u> $s_nom_article </u></b><br/>");
echo (" Le prix total hors taxe est : " . $Ptht ." Fc <br/>");
echo ("Le prix total taxe comprise est : <font color=red>  " . $Pttc . "Fc</font>");
echo "<hr>";      // ligne horizontale
 }
```

4.3. Récupérer les paramètres envoyés par un client web

Dans une page Html, l'interactivité entre l'internaute et le serveur se fera par l'intermédiaire des formulaires (zones particulières encadrées par les balises <form> et </form>, avec les attributs method et action qui définissent la procédure de leur soumission).

Associé aux formulaires, Php facilite grandement le remplissage des listes de choix et la récupération des valeurs d'un formulaire dans la même ou dans une autre page.

Exemple de remplissage d'une liste de choix

```
<html>
<head>
<title>Définition d'une liste de choix</title>
</head>
<body>
<table><tr><td valign="top">Liste de choix </td><td>
<select name="lst_valeur" size=4>
<?
// création d'un tableau à 2 dimensions (nom et value)
$nom = array( array("Brunette", "B"),    // creation Nom et value
              array ("Redodine", "R"),
            array ("Junior", "JJ"),
array ("Espé", "E"));
$nb = count($nom); // compte le nombre d'éléments
// création des 4 options
sort ($nom);
for ($i = 0 ; $i < $nb ; $i++)
{   echo "<option  value ='" .
```

```
$nom[$i][1] . "'>" .      // affectation de la "value"
        $nom[$i][0] .       // affichage du nom dans la liste
"</option>";
 }
 ?>
</select>
</td></tr>
</table>
</body>
</html>
```

Exemple de récupération des valeurs d'un formulaire dans la même page

Pour récupérer un champ de formulaire HTML appelé par exemple « nom », on utilise la variable $_POST["nom"] s'il est envoyé par la méthode POST et la variable $_GET["champ"] s'il est envoyé par la la méthode GET.

Cas d'utilisation de la méthode Get :

```
<html>
<head><title>Formulaire web</title></head>
<body>
<center><h3>Un formulaire Web Méthode Get </h3>
<h4>Récupération des valeurs des champs d'un formulaire</h4><hr>
<form name="frmPersonne" method="get">
<table>
<tr><td>Nom</td><td><input type="text" value="<?php echo $nom ?>"
name="txtNom" size="20"></td><td>Age</td>
<td><input type="text" value="<?php echo $age ?>" name="txtAge"
size="3"></td>
<tr>
```

```
</table>
<input type="submit" name="cmdUtiliser" value="Envoyer">
</form>
</center><hr>
<h4>Valeurs récupérées</h4>
<table border="1"><tr>
<td width=30>Nom</td><td width=80><?php echo $_GET["txtNom"] ?></td></tr><tr>
<td width=30>Age</td><td width=80><?php echo $_GET["txtAge"] ?></td></tr>
</table>
</body>
</html>
```

N.B. : L'existence d'une donnée dans un champ peut être testée avec la fonction isset(donnée) qui rend la valeur true si la donnée existe, false sinon. Cfr exemple suivant.

Cas d'utilisation de la méthode Post :

```
<html>
<head><title>Formulaire web</title>
<?php
$post=isset($_POST["txtNom"]) && isset($_POST["txtAge"]);
if($post){
$nom=$_POST["txtNom"];
$age=$_POST["txtAge"];
}
else {
$nom="";
$age="";
```

```
}
?>
</head>
<body><center><h3>Un formulaire Web 2</h3>
<h4>Récupération des valeurs des champs d'un formulaire</h4><hr>
<form name="frmPersonne" method="post">
<table><tr><td>Nom</td>
<td><input type="text" value="<?php echo $nom ?>" name="txtNom"
size="20"></td><td>Age</td>
<td><input type="text" value="<?php echo $age ?>" name="txtAge"
size="3"></td><tr>
</table><input type="submit" name="cmdUtiliser" value="Envoyer">
</form></center><hr>
<?php
if ($post) {
?>
<h4>Valeurs récupérées</h4>
<table border="1"><tr>
<td>Nom</td><td><?php echo $nom ?></td></tr>
<tr><td>Age</td><td><?php echo $age ?></td></tr></table>
<?php
 }
?>
</body>
</html>
```

Exemple de récupération des valeurs d'un formulaire dans une autre page

Considérons le code HTML suivant qui après envoi est traité par le fichier reponse.php

```
<html>
<head><title>Centre Informatique Infonet</title></head>
<body><center>
<form name="Inscription" Method="post" Action="reponse.php">
Enregistrement d'un candidat
<table border=0>
<tr><td>Nom</td>
<td><input type="text" name="nom" size="20" tabindex="1"></td></tr>
<tr><td>Sexe</td>
<td>Homme : <input type=radio name="sexe" value="M" checked>
<br/>Femme : <input type=radio name="sexe" value="F">     </td></tr>
<tr><td>Cochez les Cours que vous comptez suivre :</td>
</tr><tr><td></td><td>
        <input type="checkbox" name="choix1" value="ok"> Word<br/>
        <input type="checkbox" name="choix2" value="ok"> Excel<br/>
        <input type="checkbox" name="choix3" value="ok"> Access
</td></tr><tr></tr><tr><td>Niveau d'études :</td>
<td><select name="fonction">
                <option value="primaire">Primaire</option>
                <option value="D6">Diplome d'état</option>
                <option value="G3">Graduat</option>
                <option value="L2">Licence</option>
                <option value="D">DEA ou Thèse</option>
</select></td></tr><tr><td>Joindre un fichier :</td><td>
```

```
<input type="file" name="mon_fichier"></td></tr><tr>
<td align=center><input type="submit" value="Envoyer">     </td>
<td align=center><input type="reset" value="Rétablir"></td>
</tr></table>
</form></center>
</body>
</html>
```

Le code correspondant au fichier reponse qui affiche simplement les données envoyées est :

```
<html>
<head><title>Formulaire de saisie</title></head>
<body>
<p><b>Résultats du formulaire</b></p>
<?
    echo "zone de texte nom : $nom <br>" ;
    echo "bouton radio sexe : $sexe <br>" ;
echo "case à cocher word : $word <br>" ;
echo "case à cocher excel : $excel <br>" ;
    echo "case à cocher access : $access <br>" ;
    echo "liste de choix études : $etudes <br>" ;
    echo "fichier : $mon_fichier <br>" ;
echo "bt_submit : $cmdenvoyer <br>" ;
    echo "bt_reset : $cmdretablir <br>" ;
?>
</body>
</html>
```

4.4. Gérer les bases de données MySQL

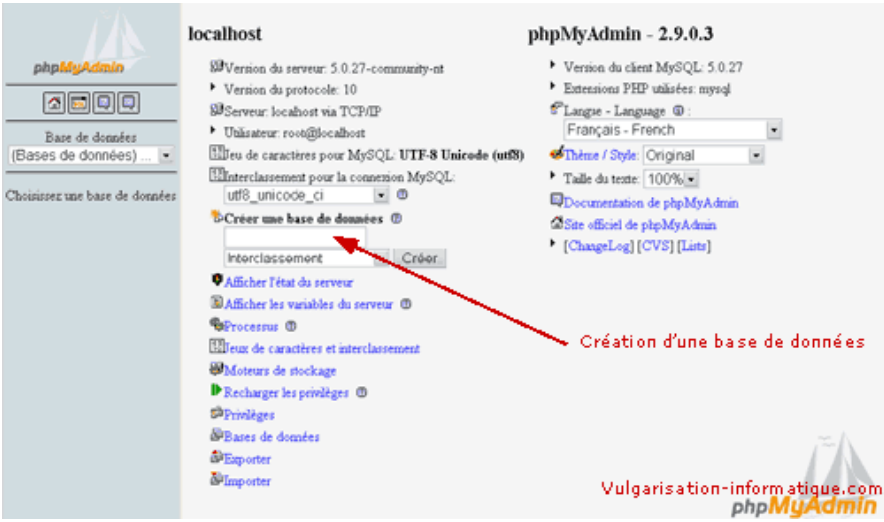
PHPMyAdmin

PHPMyAdmin est une application PHP qui va vous permettre de gérer vos bases de données. Vous allez pouvoir créer des bases, des tables, etc... sans vous préoccuper du langage SQL qu'il faut écrire. Pour accéder à PHPMyAdmin, tapez dans votre navigateur l'adresse suivante :

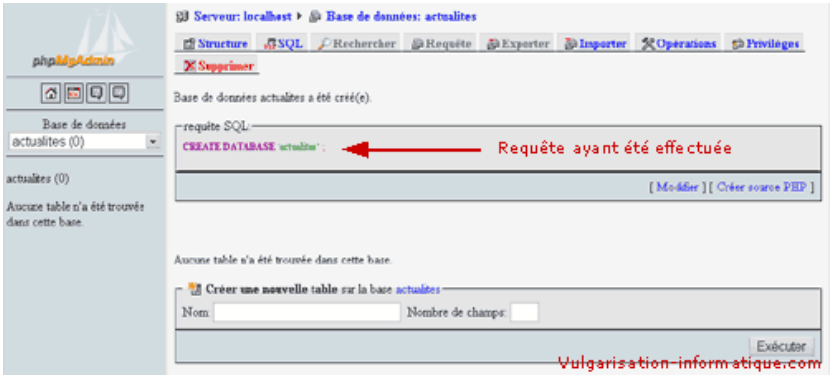
http://localhost/phpmyadmin

Si vous avez installé EasyPHP, tapez ceci : http://localhost/mysql/ (le / de fin est important)

Si tout s'est bien passé, vous allez arriver devant un écran de ce style :

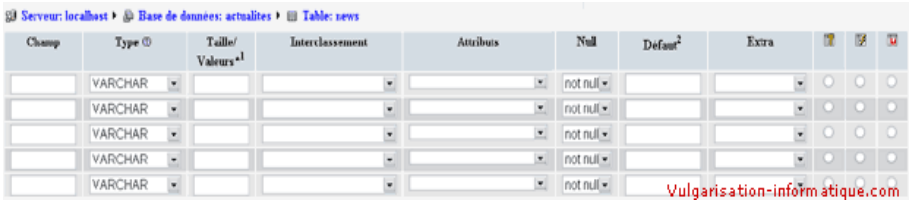


Vous allez maintenant pouvoir créer votre base de données. Pour l'exemple, nous allons créer une base de données intitulée actualites. Mettez donc actualites dans la case Créer une base de données et cliquez sur Créer. Ne vous préoccupez pas du champ interclassement.



La base de données a été créée. Comme vous pouvez le constater, le langage SQL que PHPMyAdmin a utilisé pour créer cette base est CREATE DATABASE actualites

Il s'agit de ce qu'on appelle une requête (My)SQL. Nous allons maintenant créer une table news qui va contenir des actualités pour notre site Web. Réfléchissez maintenant au nombre de champs que nous pourrions mettre. Tout d'abord, il faut pouvoir identifier chacune des actualités postées. Nous allons utiliser un champ de type numérique que l'on va nommer id_news. Chaque news aura donc un id différent. Ensuite, il faut que l'on connaisse le titre de l'actualité, sa date de parution, l'auteur de la news et son texte. Cela nous fait donc 5 champs au total. Dans la case nom, indiquez news, dans la case nombre de champs, mettez 5, puis cliquez sur Exécuter.



Vous voici maintenant devant une liste de champs qu'il va falloir déterminer. Le premier champ sera notre champ id_news. Indiquez donc dans la première case id_news. Pour le type de champ, nous allons donc utiliser un type SMALLINT avec comme attribut UNSIGNED où vous pourrez stocker 65536 news.

Dans le champ extra indiquez auto-incrément, pour permettre un ajout automatique du numéro par MySQL. Quand vous ajouterez une news, vous n'aurez donc pas à aller récupérer le numéro maximum existant pour lui ajouter un et l'insérer dans la base. Sélectionnez ensuite la troisième option (icône avec un U), qui indique que ce champ sera un champ ayant une valeur unique (il s'agit également d'un INDEX qui va vous permettre d'accélérer les recherches sur la base).

Pour le second champ, il s'agit du titre de l'actualité. On peut donc nommer ce champ titre, et lui mettre un type VARCHAR (mettez 255 dans l'option "taille/valeurs").

Pour le troisième champ, il s'agit d'une date de parution. Nous allons utiliser un type INT avec l'attribut UNSIGNED, qui permettra de stocker des dates provenant de la fonction time() de PHP.

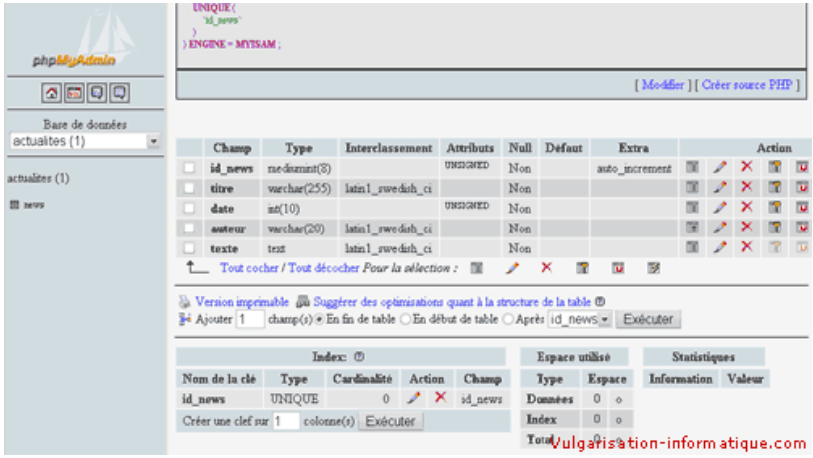
Pour le quatrième champ, nous allons utiliser le nom auteur, qui contiendra le pseudo du posteur de la news, avec le type VARCHAR et 20 caractères dans la zone "taille/valeurs".

Le dernier champ sera de type TEXT et portera le nom de texte. Il servira à contenir le texte de la news.

Choisissez dans la zone "moteur de stockage" l'option MyISAM, car il est supporté par plusieurs hébergeurs. Une fois que tout est rempli, cliquez sur le bouton Sauvegarder. Si tout va bien, PHPMyAdmin exécutera la requête suivante :

```
CREATE TABLE `news` (
   `id_news` MEDIUMINT UNSIGNED NOT NULL AUTO_INCREMENT ,
   `titre` VARCHAR( 255 ) NOT NULL,
   `date` INT UNSIGNED NOT NULL ,
   `auteur` VARCHAR( 20 ) NOT NULL ,
   `texte` TEXT NOT NULL , UNIQUE ( `id_news` ) )
ENGINE = MYISAM ;
```

Vous vous retrouvez devant cet écran :



Maintenant, nous allons insérer des données dans cette table news. Pour ce faire, remarquez les onglets présents en haut du cadre de droite :

Cliquez sur l'onglet Insérer. Vous obtenez à l'écran une représentation des différents champs avec des zones de texte pour vous permettre de les remplir :

Il va maintenant falloir remplir les champs. Voici ce que vous pouvez mettre :

Tout d'abord dans le champ id_news, ne mettez rien car c'est un champ automatique. Pour le titre, à vous de trouver un titre comme par exemple « Bienvenue sur mon site ». En ce qui concerne la date, faites dans un fichier PHP ceci :

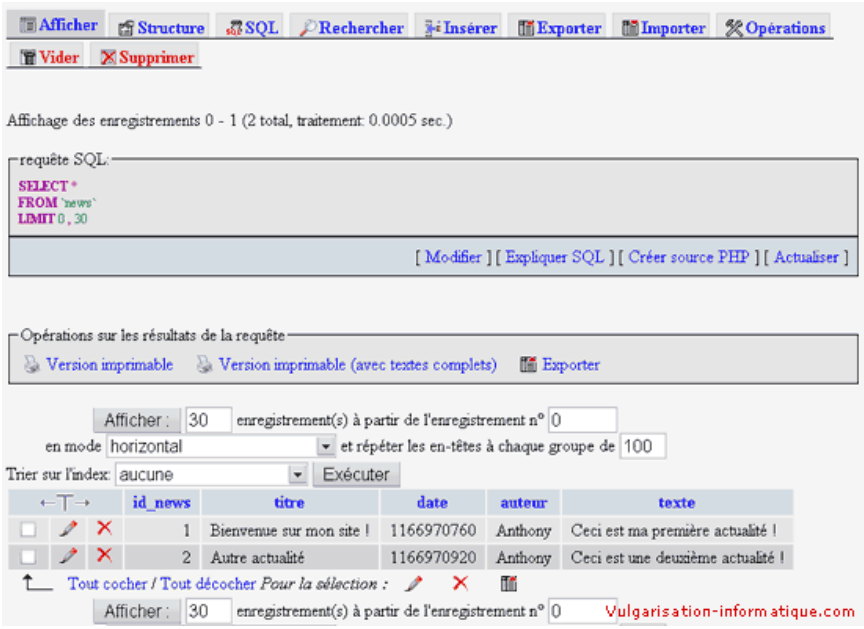
```
<?php
    echo time();
?>
```

Notez la valeur obtenue et renseignez-la dans le champ date. Pour le champ auteur, indiquez votre pseudo. Mettez ensuite un texte dans la zone de texte correspondant au champ texte, ce sera le texte de la news. Cliquez ensuite sur Exécuter.

Voilà ce qu'a pu donner la requête :

INSERT INTO `news` (`id_news` , `titre` , `date`, `auteur` , `texte`) VALUES (NULL , 'Bienvenue sur mon site !', '1166970760', 'Anthony', 'Ceci est ma première actualité !');

Si vous souhaitez maintenant consulter les données présentes dans la table, vous pouvez cliquer sur l'onglet Afficher. Exemple :



PHP et les requêtes MySQL

Introduction

Après avoir vu ce qu'était une base de données et découvert comment fonctionnait PHPMyAdmin dans le chapitre précédent, nous allons maintenant nous intéresser à la partie PHP, car n'oubliez pas que PHP va vous permettre de communiquer avec votre base de données. Dans ce chapitre, on utilisera toujours PHP pour communiquer avec MySQL, mais sachez qu'il est tout à fait possible d'entrer les requêtes à la main dans ce qu'on appelle une console (pour voir la console Windows, cliquez sur **démarrer**, **Exécuter** et tapez **cmd**).

Avant toute chose, il faut que vous sachiez qu'on ne fait pas directement une requête avec une fonction PHP. Il faut d'abord passer par plusieurs étapes. Un SGBD dispose ainsi de droits d'accès et peut comporter plusieurs bases. Il faut donc tout d'abord s'identifier au serveur MySQL que l'on souhaite utiliser. Une fois cette identification faite, il faut sélectionner la base sur laquelle on souhaite travailler. Et après ça, vous pourrez effectuer vos requêtes ! Voyons comment cela fonctionne.

Connexion au serveur MySQL

La première étape consiste à se connecter au serveur MySQL. Pour cela, on utilise la fonction mysql_connect() de PHP. Elle prend trois arguments :

- L'adresse IP (ou l'alias pointant vers l'ip) du serveur MySQL ;
- Le nom d'utilisateur ;
- Le mot de passe.

Ces trois paramètres vous sont fournis par votre hébergeur. Lorsque vous travaillez avec EasyPHP ou Wamp, les paramètres à utiliser sont :

- localhost
- root

et il n'y a pas de mot de passe à renseigner.

Voilà comment vous pouvez coder ça pour un usage local :

```
<?php
   $connexion = mysql_connect('localhost', 'root', '') OR die('Erreur de connexion');
?>
```

La partie **OR die('Erreur de connexion')** n'est pas indispensable, mais elle permet de couper le script si il n'a pas réussi à se connecter au serveur. Il est important pour une application sécurisée de bien gérer les erreurs que vous pouvez avoir à la connexion ou lors de la sélection de base de données (on va voir ça tout de suite)

Sélection de la base de données

Une fois que vous êtes connecté au serveur MySQL, sachant qu'il peut contenir une infinité de bases de données, il faut bien qu'il sâche sur laquelle vous souhaitez travailler. On utilise pour cela la fonction mysql_select_db() en PHP. Elle prend comme paramètre le nom de la base de données que vous souhaitez utiliser.

Pour ceux qui ont suivi le chapitre précédent, j'avais utilisé une base **actualites**. Tous mes exemples vont donc s'appuyer sur cette base que nous avons créée ensemble. Voici comment dire à PHP que nous allons travailler sur cette base :

```
<?php
   mysql_select_db('actualites') OR die('Sélection de la base impossible');
?>
```

Encore une fois, la partie **OR die('Sélection de la base impossible')** n'est pas indispensable mais conseillée.

Déconnexion du serveur MySQL

Il s'agit d'une étape très importante et trop souvent négligée dans de nombreux scripts. Il faut savoir que MySQL dispose d'un paramètre spécifiant le nombre maximum de connexions simultannées qu'il peut traiter. En local, vous n'aurez pratiquement jamais de problème avec ce paramètre (qui se manifeste par une erreur de **Max user connections** et vous empêche donc d'effectuer vos requêtes). En revanche, chez un hébergeur, ce paramètre est souvent placé à une valeur de 3 ou de 5 (5 étant préférable). Cela veut dire que 5 connexions pourront avoir lieu quasiment simultanément. Vous vous dites "c'est énorme, il n'y aura jamais personne en même temps qui pourra cliquer sur mon site". Le problème, est que la connexion est par défaut, si vous ne la fermez pas, active pendant toute la durée de génération de la page. Si vous ouvrez votre connexion tout en haut de la page et que le serveur met 1 seconde (ce qui est énorme) pour générer la page, votre connexion restera ouverte pendant une seconde. Vous imaginez qu'il devient alors très facile d'avoir 5 connexions à la même seconde pour peu que vous ayez un peu de visiteurs ou des scripts très lents. Il faut donc fermer la connexion le plus tôt possible, après avoir effectué la dernière requête, et **AVANT TOUT TRAITEMENT**.

Vous allez voir tout à l'heure que nous allons utiliser la fonction mysql_query() pour effectuer des requêtes MySQL, et bien voici un schéma qu'il faudrait adopter pour bénéficier d'une optimisation maximum :

```
<?php
    $connexion = mysql_connect('localhost', 'root', '') OR die('Erreur de connexion');
    mysql_select_db('actualites') OR die('Erreur de sélection de la base');
    $requete1 = mysql_query('....');
    $requete2 = mysql_query('....');
    //Ici vous placez vos autres requêtes
    mysql_close(); //On ferme la connexion à MySQL
    //Ici vous mettez le code PHP qui va aller récupérer les données provenant des r
equêtes (fonction mysql_fetch_row() par exemple)
?>
```

Comme vous pouvez le voir, on utilise la fonction **mysql_close()** pour fermer la connexion au serveur MySQL. Par défaut, elle ne prend pas de paramètre.

Lire des données dans une table

Avant de lire des données dans une table, il faut que votre table contienne des enregistrements. On va se servir toujours de la même table que précédemment, c'est à dire la table **news**. Vous pouvez y insérer quelques enregistrements via PHPMyAdmin, comme je vous l'ai déjà montré au chapitre précédent.

Qu'il s'agisse d'une lecture, d'une écriture ou d'une modification de données ou de paramètres, il vous faudra passer par la fonction mysql_query(). Query signifie requête en anglais. C'est cette fonction qui va vous permettre d'interagir avec MySQL.

```
<?php
    $requete = mysql_query('Ici votre requête SQL');
?>
```

Le résultat de la requête sera retourné dans la variable $requete. Attention, il s'agit d'une variable de type resource. Vous ne pourrez donc pas faire un echo de cette variable, ça ne vous renverra pas ce que vous attendez. On va en reparler juste après ;)

Voyons comment effectuer une requête de sélection de certains champs :

En SQL, lorsque vous souhaitez sélectionner des données provenant d'une table, on utilise tout d'abord le mot SELECT, qui veut dire que vous vous apprêtez à récupérer des données. Ensuite, vous devez indiquer à MySQL la liste des champs de la table que vous souhaitez voir dans votre résultat, séparés par des virgules. Vous devez ensuite indiquer à quelle table vous souhaitez prendre les données via le mot clé FROM suivi du nom de la table

Si nous souhaitons récupérer les champs titre et texte de la table news, voici comment nous allons procéder :

```
<?php
    $requete = mysql_query('SELECT titre, texte FROM news');
?>
```

Bon c'est bien beau tout ça, mais comment fait-on pour récupérer le résultat d'une requête sous forme textuelle ? Et bien on utilise pour cela quatre fonctions de PHP (en fait on en utilise une parmi les quatre). Voilà les trois fonctions que vous pouvez utiliser :

- mysql_fetch_row() : cette fonction va retourner un tableau avec des indices numériques (l'indice 0 correspond au premier champ de la requête, l'indice 1 au second champ ...). Je vous conseille d'utiliser cette fonction pour des tables dans lesquelles vous sélectionnez peu de champs (un ou deux), car c'est la fonction la plus rapide.

- mysql_fetch_array() : cette fonction va retourner un tableau avec par défaut les indices numériques et associatifs. Elle est plus lente que mysql_fetch_row() mais plus simple d'utilisation. Pour la rendre plus rapide, on peut utiliser la fonction mysql_fetch_assoc() qui va uniquement retourner un tableau avec les indices associatifs.

- mysql_fetch_object() : cette fonction retourne un objet qui aura pour attributs les champs de la requête MySQL.

Voyons voir ce que ça donne pour sélectionner tous les enregistrements de la table actualites, et les afficher :

```
<?php
   mysql_connect('localhost', 'root', '') OR die('Erreur de connexion à la base');
   mysql_select_db('actualites') OR die('Erreur de sélection de la base');
   $requete = mysql_query('SELECT titre, texte FROM news') OR die('Erreur de la r
equête MySQL');
   mysql_close();
   /**
   * On récupère les données
   * Tant qu'une ligne sera présente, la boucle continuera
   */
   while($resultat = mysql_fetch_object($requete))
   {
      echo '<p>Titre : '.$resultat->titre.'. Texte : '.$resultat->texte.'</p>';
   }
?>
```

Avec la fonction **mysql_fetch_row()**, voici ce que ça aurait donné :

```
<?php
   mysql_connect('localhost', 'root', '') OR die('Erreur de connexion à la base');
   mysql_select_db('actualites') OR die('Erreur de sélection de la base');
   $requete = mysql_query('SELECT titre, texte FROM news') OR die('Erreur de la r
```

```
equête MySQL');
    mysql_close();
    /**
    * On récupère les données
    * Tant qu'une ligne sera présente, la boucle continuera
    */
    while($resultat = mysql_fetch_row($requete))
    {
        echo '<p>Titre : '.$resultat[0].'. Texte : '.$resultat[1].'</p>';
    }
?>
```

Et avec la fonction **mysql_fetch_assoc()** (équivalente à la fonction **mysql_fetch_array()** à laquelle on passe une constante en second paramètre : **MYSQL_ASSOC**) :

```
<?php
    mysql_connect('localhost', 'root', '') OR die('Erreur de connexion à la base');
    mysql_select_db('actualites') OR die('Erreur de sélection de la base');
    $requete = mysql_query('SELECT titre, texte FROM news') OR die('Erreur de la r
equête MySQL');
    mysql_close();
    /**
    * On récupère les données
    * Tant qu'une ligne sera présente, la boucle continuera
    */
    while($resultat = mysql_fetch_assoc($requete))  //équivalent à  while($resultat
= mysql_fetch_array($requete, MYSQL_ASSOC))
    {
        echo '<p>Titre : '.$resultat['titre'].'. Texte : '.$resultat['texte'].'</p>';
    }
?>
```

Pour tout ce qui est insertion de données, il est conseillé de lire les requêtes MySQL associées dans ce cours : requêtes MySQL d'insertion de données. Le principe est le même, sauf que là on ne récupère aucun résultat.

Exemple : Considérons une base de données MySql « Annuaire » constituée de la table « personne ». Nous présentons dans les lignes qui suivent la procédure qui permet d'afficher le contenu de cette base.

```
<?
$sql="select* from personne" // requête
// connexion au serveur
$maconnexion=mysql-connect("localhost", "root", "") ;
if ($maconnexion==FALSE)
{
die("la connexion a échouée") ;
}
//déclaration du nom de la base
$connectBase=mysql_select_db("Annuaire", $maconnexion);
if (=connectBase==FALSE) // si le base est inaccessibilité
{
die("base inaccessible") ;
}
//envoi au serveur le nom de la requête
$result=mysql_query($sql) ;
if ($result==FALSE) // si la requête est incorrecte
{
die("requête incorrecte : $sql") ;
}
//boucle sur les enregistrements retournés
while (($ligne=mysql_fetch_array($result))==TRUE)
{
// Affichage des champs nom et telephone
```

```
echo ("<h3>". $ligne["nom"]."  ". $ligne["telephone"]."</h3>") ;
}
//libération de l'espace mémoire pris au serveur MySql
mysql_free_result ($result) ;
//libération de la connexion au serveur MySql
mysql_close ($maconnexion) ;
?>
```

3. Systèmes de gestion de contenu (SGC)

3.1. Introduction

La gestion de contenu permet de gérer des sites web, que ce soient des sites

Internet ou des sites Intranet, et de partager l'information d'une manière efficace.

Les systèmes de gestion de contenu (SGC en anglais, Content Management Systems – CMS), également liés à la notion de gestion de contenu web, sont des plates-formes entièrement paramétrables, dédiées au contenu, pouvant être déployées rapidement. Les CMS améliorent la productivité des intervenants et la réactivité des sites web car ils rationalisent et automatisent des tâches répétitives.

Ce sont des outils qui offrent la possibilité à des non-techniciens de gérer des sites web, sans compétence informatique particulière, et de favoriser ainsi le travail collaboratif.

Concernant les organisations, les CMS doivent être adaptés à leurs besoins et des compétences en programmation sont alors nécessaires.

Les responsables de sites web des organisations deviennent donc responsables de la coordination des différents intervenants (développeurs, graphistes, rédacteurs, documentalistes, utilisateurs).

Les CMS peuvent être répartis en deux familles de logiciels :

- les CMS payants, très sophistiqués tels que : Documentum, Vignette, Interwoven, Tridion qui sont utilisés dans les organisations pour construire de véritables référentiels de contenu qui vont au-delà des sites web,
- les CMS issus du mouvement des logiciels libres plutôt orientés vers la seule gestion de sites web : SPIP, ZOPE, PhpNuke, Typo3, WordPress, Drupal, etc … Ils ont donc été utilisés en premier lieu à un niveau individuel ou pour des associations. Les organisations sont maintenant de plus en plus intéressées par leur intégration dans leur système d'information.

3.2. Avantages des logiciels de gestion de contenu

Les CMS permettent aux webmestres et aux développeurs de :

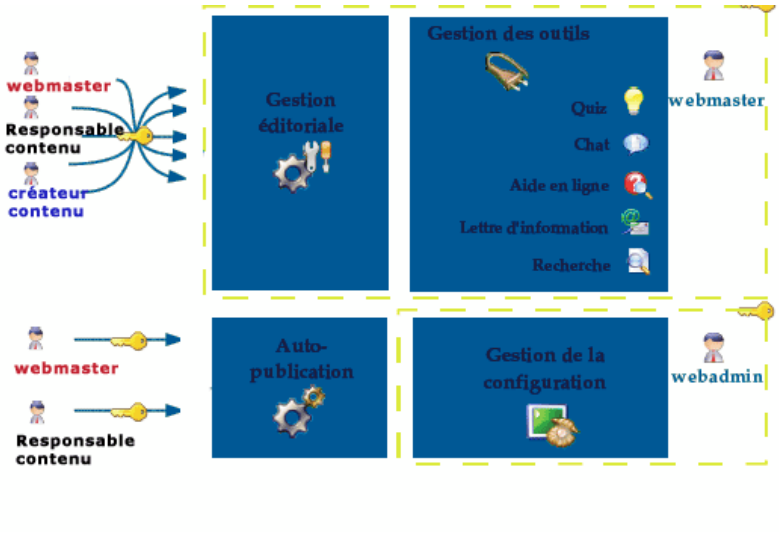
- contrôler l'entrée et la sortie du contenu au moyen d'une interface (souvent un navigateur web),

- maîtriser l'aspect et la publication de l'ensemble des pages d'un site,
- séparer totalement le contenu de la structure et de la mise en page,
- permettre une gestion automatisée des liens amont et aval d'une page (notamment en cas de suppression de page),
- diffuser le contenu sous plusieurs formats de manière automatique (exemples : HTML, PDF, TXT, DOC,…),
- permettre à chaque utilisateur de créer et gérer du contenu.

L'utilisation d'un CMS au sein d'une organisation peut lui permettre de :

- réduire les coûts de maintenance de son ou ses sites web, notamment grâce aux gains de productivité du travail des webmestres,
- réduire les délais de publication : le goulot d'étranglement des sites statiques n'existe plus sur le plan technique (mais il peut demeurer si le système de validation du contenu est trop contraignant). De plus, une information saisie une seule fois peut être mise en ligne sur plusieurs sites simultanément,
- augmenter la valeur ajoutée du contenu du site grâce à la pertinence et à l'actualisation des informations : les rédacteurs sont des spécialistes des questions traitées et ils peuvent contribuer à l'alimentation du site via Internet, quelle que soit leur position géographique.

3.3. Fonctionnalités attendues pour les logiciels de gestion de contenu



Gestion des intervenants

Les rédacteurs contribuant à la vie du site peuvent, en fonction des droits qui leur ont été alloués, créer, modifier ou supprimer du contenu via leur navigateur web avec l'utilisation de formulaires.

Toutes les modifications sont répercutées sur le site, après validation éditoriale éventuelle, via un système de pages dynamiques.

Les annuaires d'entreprises peuvent être utilisés pour définir des groupes d'utilisateurs et l'attribution des droits individuels et collectifs.

Les CMS offrent donc la possibilité à chaque membre d'une organisation de participer à la vie d'un site web, tout en respectant les règles hiérarchiques existantes. Il en découle une valorisation du travail des collaborateurs qui peuvent se traduire par des gains de productivité.

Certaines informations peuvent avoir un caractère confidentiel. Or, un CMS permet de respecter la confidentialité en mettant en place des restrictions d'accès au contenu du site en fonction du rôle et des droits de chaque utilisateur. Par exemple, une note d'un directeur général ne pourra être lue que par les responsables de l'organisation.

Enfin, les rédacteurs n'étant pas des spécialistes du langage HTML, les CMS proposent des interfaces utilisateurs simples et intuitives, accessibles par le biais des navigateurs web. Il est aussi important que les CMS puissent transformer automatiquement tout document sous format bureautique habituel (.doc, .pdf, .txt, .xls,…) produit préalablement par les rédacteurs.

Il faut distinguer deux types d'interfaces :

- l'interface de « back office »

Cette interface a pour rôle de simplifier la gestion des sites web, elle peut être appelée également partie privée ou espace privé d'un site. Elle est utilisée par les administrateurs des sites, les webmestres, les contributeurs/rédacteurs…

- l'interface de « front office »

Cette interface est la partie publique du site, elle permet de gérer les visiteurs et de leur faciliter l'accès aux informations dont ils ont besoin.

La plupart des logiciels libres CMS offrent la possibilité d'associer des forums de discussion (modérés ou non) aux articles publiés afin de transformer les sites web en outil de communication bilatérale. Les commentaires des internautes peuvent apporter une valeur ajoutée à l'information publiée.

Gestion des versions du contenu

Un CMS permet de conserver et d'archiver les différentes versions d'un document avec le jour, l'heure et l'auteur de la modification.

Le contrôle des versions permet à plusieurs intervenants de travailler sur un même fichier, sans que les modifications des uns « n'écrasent » le travail des autres.

De plus, des forums de discussion internes peuvent être disponibles et les commentaires sont utiles au moment de l'élaboration des documents.

Utilisation des métadonnées et recherche

La complexité et la variété des systèmes d'information s'étant accrues, les métadonnées constituent des structures et des descriptions émises à un niveau d'abstraction supérieur (méta) et relatives à un niveau inférieur (ou référence).

Ainsi, les métadonnées sont « des données sur les données » de type :

- Identification (titre, auteur, mots-clés,...),
- Administration (droits),
- Localisation (objet physique, URL),
- Utilisation (caractéristiques physique, format de fichier, …).

Les métadonnées générées par un CMS peuvent de plus faciliter ou améliorer la diffusion de données de base. Elles peuvent ainsi s'interfacer avec les logiciels de gestion documentaire existants et tenir compte des pratiques des documentalistes.

L'utilisation de métadonnées peut ainsi servir de base de référencement pour les moteurs de recherche.

En effet, les metatags, balises HTML décrivant le contenu des pages web, sont des informations utilisées par les serveurs ou les moteurs de recherche. Ces metatags peuvent être alimentées automatiquement par les métadonnées du CMS.

Ces informations sont visibles en faisant un « clic-droit » sur une page web et en choisissant « Afficher la source ». Le code HTML de la page s'ouvre dans le bloc-notes de l'ordinateur.

Exemples de balises HTML :

<title>Titre de la page</title>

<meta name="Description" content="Description du contenu de la page">

<meta name="Author" content="Nom de l'auteur du site, de la page">

<meta name="Keywords" content="Liste des mots-clés de la page">

Un CMS doit proposer, en outre, un outil de recherche interne puissant permettant aussi bien d'effectuer des recherches en fonction des métadonnées que des recherches en texte intégral sur l'ensemble du site.

Validation du contenu

Un CMS doit pouvoir gérer le cycle de vie des documents et donc remplir des actions du type : soumettre un document, le renvoyer pour correction ou l'approuver.

L'organisation du processus de validation des documents doit pouvoir s'effectuer en fonction de différents critères tels que la cible du document, son type, son auteur, etc…

Stockage des documents

En général, les CMS très sophistiqués utilisent une base de données documentaire pour stocker l'ensemble des documents avec leurs métadonnées. Ces dernières sont plus adaptées à ce type de stockage que les bases de données relationnelles.

Cependant, dans la plupart des cas, ce sont des bases de données relationnelles traditionnelles (MySql, Oracle, Sybase,…) qui sont utilisées.

Les documents ou les fragments de documents sont stockés en XML dans des BLOB (Big Large Objects Binary) et un certain nombre d'informations des documents est remonté dans des tables relationnelles afin de pouvoir effectuer rapidement des recherches.

Intégration d'autres sources de données

Lorsqu'une organisation prend la décision d'acquérir un CMS, elle doit vérifier qu'il peut être compatible avec le système d'information existant.

Le CMS doit pouvoir retrouver et stocker des données en provenance d'autres disques, d'autres serveurs.

L'utilisation du langage XML permet de séparer le contenu de sa présentation et donc de décrire les informations et les organiser finement. L'utilisation de gabarits (templates) permet de créer des modèles de pages dynamiques, de documents dont le contenu et la forme peuvent être modifiés indépendamment l'un de l'autre.

Les CMS basés sur le XML proposent des services de transformation, de présentation et de validation du contenu.

Une connexion doit être établie avec une base de données des utilisateurs LDAP (annuaire d'entreprise, cf. glossaire) pour gérer l'accès au CMS ainsi que les droits affectés à chaque personne ou groupe de personnes.

Les sources d'information pouvant également provenir d'autres sites web, il est alors possible de mutualiser les contenus de sites web différents grâce au mécanisme de syndication de site (exemple : les journaux qui diffusent les dépêches des agences de presse).

3.4. Quelques solutions existantes

Le tableau suivant recense sept projets de logiciels libres de gestion de contenu web.

Tableau : Logiciels libres de gestion de contenu web

CMS	Caractéristiques	Commentaires
Slash	- Perl&MySQL -Articles, catégories, commentaires, sondages -Interface modulable (système de thèmes et modèles) - Extensible (système de plug-in) -Moteur de recherche - Rapide, bonne montée en charge, sécurisé	A l'origine le CMS utilisé par le site Slashdot, il a néanmoins tendance à être moins bien vu dans un web où l'optimisation et la standardisation prennent le pas.
phpNuke / postNuke	- PHP & MySQL - Articles, catégories, commentaires, sondages - Extensible (système de modules et de blocs) - Moteur de recherche	phpNuke a été l'un des premiers CMS en PHP, et a été adopté par de nombreux sites. Réputé pour son manque de flexibilité, il est remplacé peu à peu par postNuke
SPIP	- PHP3 & MySQL - Articles, brèves, rubriques, forums de discussion, pétitions, statistiques - Multilinguisme - Interface modulable - Extensible (système de boucles) - Moteur de recherche - Système de correction typographique	L'un des premiers CMS français qui permet de gérer facilement un site complet.
Drupal	- PHP & MySQL/PostgreSQL/SQL Server...	Un couteau suisse de la gestion de contenu : il peut

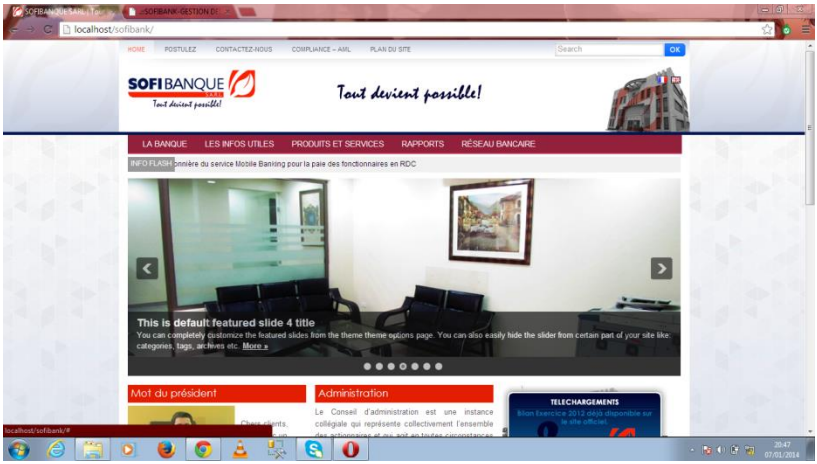
	- Articles, catégories, sondages, forums, wiki, weblog - Interface modulable - Extensible (modules) - Moteur de recherche	pour ainsi dire tout faire (CMS, weblog, wiki, forum...),et se révèle très puissant.
Typo3	- PHP & MySQL+AdoDB/PEAR::DB - Articles, catégories, newsletter - Extensible - XHTML et CSS en standard - Editeur WYSIWYG - Import de documents Word - Moteur de recherche - Interface modulable - Conservation des anciennes versions d'articles…	Un outil complet qui permet de rapidement créer plusieurs sites. La phase d'apprentissage peut se révéler longue pour l'administrateur.
EZ Publish	- PHP - Toutes les fonctions classiques - Support PDF, WebDAV, LDAP, Unicode...	Sa conception sous forme d'Objets en fait un CMS très naturel à gérer.
Mambo	- PHP & MySQL - Editeur WYSIWYG - Interface modulable - Extensible (modules et composants) - Gestion de publicités - Moteur de recherche	Très complet et simple d'utilisation, Mambo est l'une des références des CMS.

4. Etude de cas

Nous présentons ici une application web sur la gestion des licences d'exportation et d'importation des biens et services via une banque locale, la sofibanque qui se présente comme client de la banque centrale du Congo.

Les **interfaces** proposées sont :

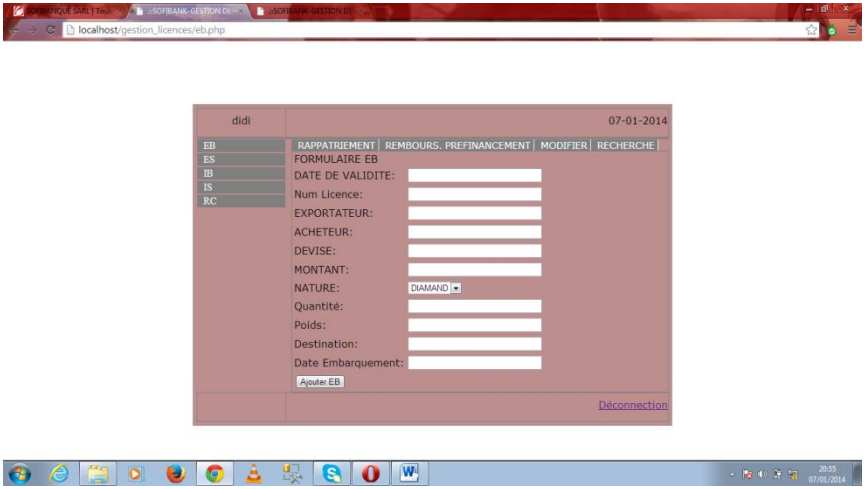
Interface d'accueil du site web



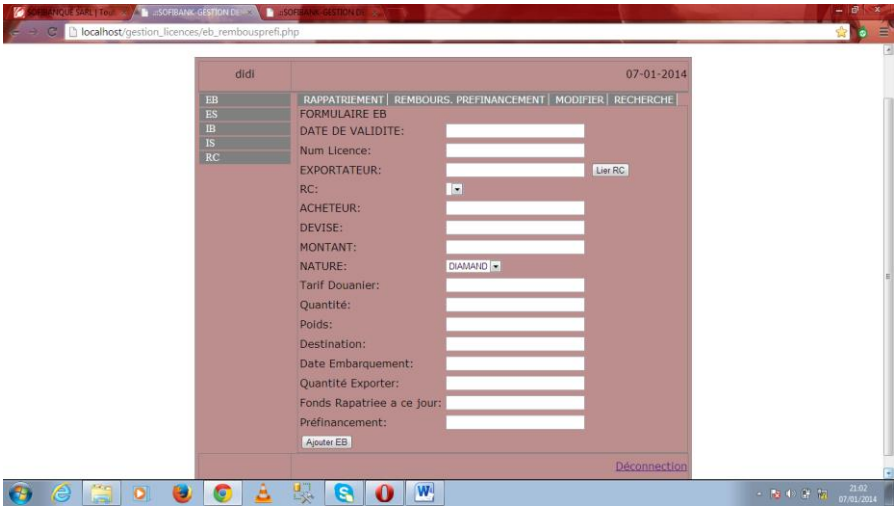
Interface d'accès

Menu de gestion des licences

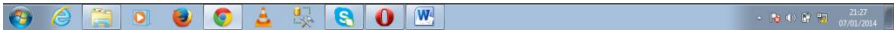
Interface de la gestion de EB (Exportation des Biens)


localhost/gestion_licences/eb.php
didi
07-01-2014
EB
ES
IB
IS
RC
RAPPATRIEMENT
REMBOURS. PREFINANCEMENT
MODIFIER
RECHERCHE
FORMULAIRE EB
DATE DE VALIDITE:
Num Licence:
EXPORTATEUR:
ACHETEUR:
DEVISE:
MONTANT:
NATURE:
DIAMAND
Quantité:
Poids:
Destination:
Date Embarquement:
Ajouter EB
Déconnection

Menu Remboursement préfinancement

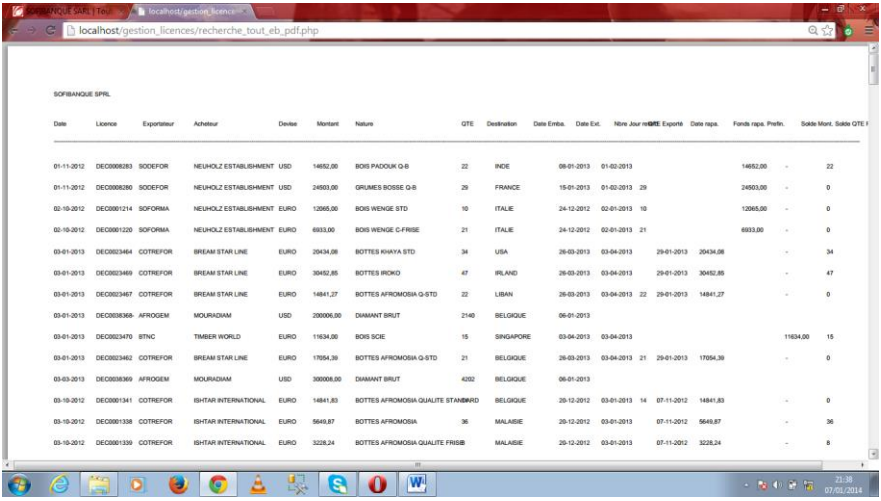


Interface de modification

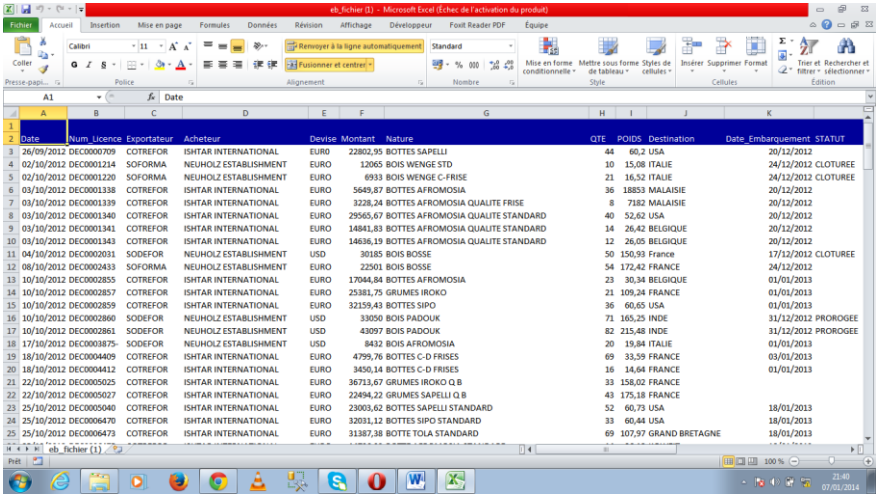


Interface de recherche

Exportation des données format PDF



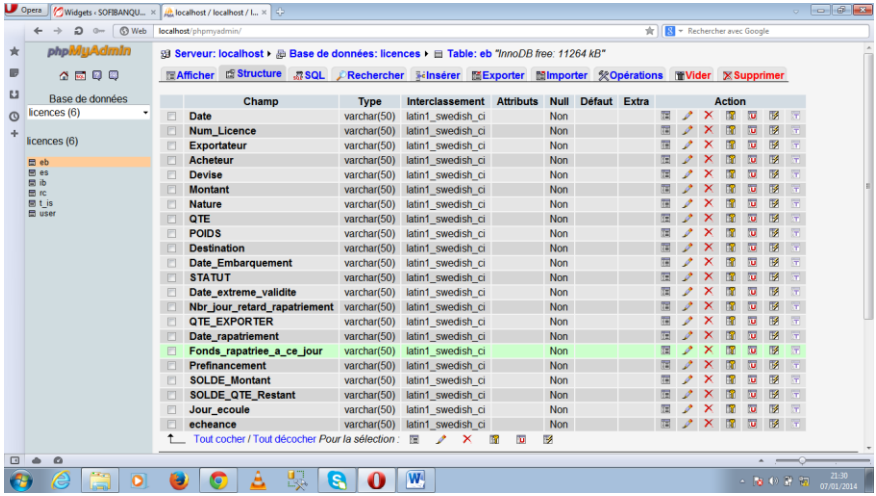
Exportation des données Format Excel



Date	Num_Licence	Exportateur	Acheteur	Devise	Montant	Nature	QTE	POIDS	Destination	Date_Embarquement	STATUT
26/09/2012	DEC0000709	COTREFOR	ISHTAR INTERNATIONAL	EUR0	22802,95	BOTTES SAPELLI	44	60,2	USA	20/12/2012	
02/10/2012	DEC0001214	SOFORMA	NEUHOLZ ESTABLISHMENT	EURO	12065	BOIS WENGE STD	10	15,08	ITALIE	24/12/2012	CLOTUREE
02/10/2012	DEC0001220	SOFORMA	NEUHOLZ ESTABLISHMENT	EURO	6933	BOIS WENGE C-FRISE	21	16,52	ITALIE	24/12/2012	CLOTUREE
03/10/2012	DEC0001338	COTREFOR	ISHTAR INTERNATIONAL	EURO	5649,87	BOTTES AFROMOSIA	36	18853	MALAISIE	20/12/2012	
03/10/2012	DEC0001339	COTREFOR	ISHTAR INTERNATIONAL	EURO	3228,24	BOTTES AFROMOSIA QUALITE FRISE	8	7182	MALAISIE	20/12/2012	
03/10/2012	DEC0001340	COTREFOR	ISHTAR INTERNATIONAL	EURO	29565,67	BOTTES AFROMOSIA QUALITE STANDARD	40	52,62	USA	20/12/2012	
03/10/2012	DEC0001341	COTREFOR	ISHTAR INTERNATIONAL	EURO	14841,83	BOTTES AFROMOSIA QUALITE STANDARD	14	26,42	BELGIQUE	20/12/2012	
03/10/2012	DEC0001343	COTREFOR	ISHTAR INTERNATIONAL	EURO	14636,19	BOTTES AFROMOSIA QUALITE STANDARD	12	26,05	BELGIQUE	20/12/2012	
04/10/2012	DEC0002031	SODEFOR	NEUHOLZ ESTABLISHMENT	USD	30185	BOIS BOSSE	50	150,93	France	17/12/2012	CLOTUREE
08/10/2012	DEC0002433	SOFORMA	NEUHOLZ ESTABLISHMENT	EURO	22501	BOIS BOSSE	54	172,42	FRANCE	24/12/2012	
10/10/2012	DEC0002855	COTREFOR	ISHTAR INTERNATIONAL	EURO	17044,84	BOTTES AFROMOSIA	23	30,34	BELGIQUE	01/01/2013	
10/10/2012	DEC0002857	COTREFOR	ISHTAR INTERNATIONAL	EURO	25381,75	GRUMES IROKO	21	109,24	FRANCE	01/01/2013	
10/10/2012	DEC0002859	COTREFOR	ISHTAR INTERNATIONAL	EURO	32159,43	BOTTES SIPO	36	60,65	USA	01/01/2013	
10/10/2012	DEC0002860	SODEFOR	NEUHOLZ ESTABLISHMENT	USD	33050	BOIS PADOUK	71	165,25	INDE	31/12/2012	PROROGEE
10/10/2012	DEC0002861	SODEFOR	NEUHOLZ ESTABLISHMENT	USD	43097	BOIS PADOUK	82	215,48	INDE	31/12/2012	PROROGEE
17/10/2012	DEC0003875-	SODEFOR	NEUHOLZ ESTABLISHMENT	USD	8432	BOIS AFROMOSIA	20	19,84	ITALIE	01/01/2013	
18/10/2012	DEC0004409	COTREFOR	ISHTAR INTERNATIONAL	EURO	4799,76	BOTTES C-D FRISES	69	33,59	FRANCE	03/01/2013	
18/10/2012	DEC0004412	COTREFOR	ISHTAR INTERNATIONAL	EURO	3450,14	BOTTES C-D FRISES	16	14,64	FRANCE	01/01/2013	
22/10/2012	DEC0005025	COTREFOR	ISHTAR INTERNATIONAL	EURO	36713,67	GRUMES IROKO Q B	33	158,02	FRANCE		
22/10/2012	DEC0005027	COTREFOR	ISHTAR INTERNATIONAL	EURO	22494,22	GRUMES SAPELLI Q B	43	175,18	FRANCE		
25/10/2012	DEC0005040	COTREFOR	ISHTAR INTERNATIONAL	EURO	23003,62	BOTTES SAPELLI STANDARD	52	60,73	USA	18/01/2013	
25/10/2012	DEC0006470	COTREFOR	ISHTAR INTERNATIONAL	EURO	32031,12	BOTTES SIPO STANDARD	33	60,44	USA	18/01/2013	
25/10/2012	DEC0006473	COTREFOR	ISHTAR INTERNATIONAL	EURO	31387,38	BOTTE TOLA STANDARD	69	107,97	GRAND BRETAGNE	18/01/2013	

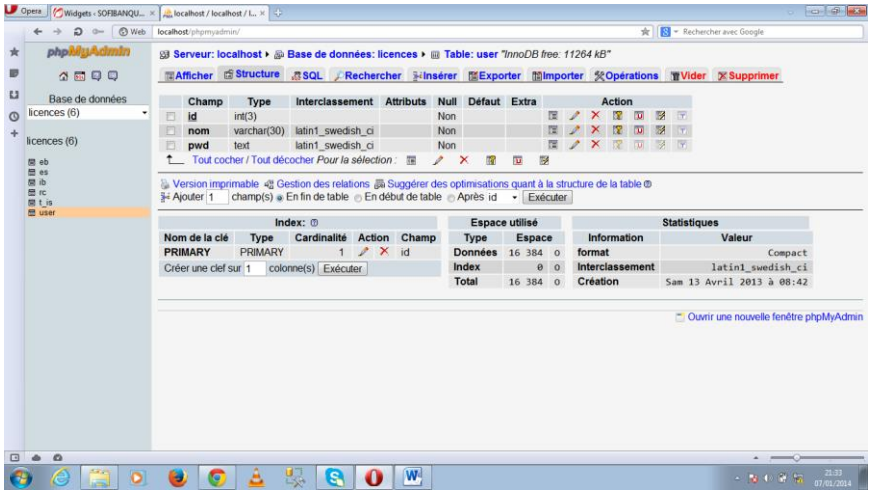
Notre base de données tourne sur Mysql qui est un SGBD léger.

Notre table EB avec les champs et les types de données utilisées



Champ	Type	Interclassement	Attributs	Null	Défaut	Extra	Action
Date	varchar(50)	latin1_swedish_ci		Non			
Num_Licence	varchar(50)	latin1_swedish_ci		Non			
Exportateur	varchar(50)	latin1_swedish_ci		Non			
Acheteur	varchar(50)	latin1_swedish_ci		Non			
Devise	varchar(50)	latin1_swedish_ci		Non			
Montant	varchar(50)	latin1_swedish_ci		Non			
Nature	varchar(50)	latin1_swedish_ci		Non			
QTE	varchar(50)	latin1_swedish_ci		Non			
POIDS	varchar(50)	latin1_swedish_ci		Non			
Destination	varchar(50)	latin1_swedish_ci		Non			
Date_Embarquement	varchar(50)	latin1_swedish_ci		Non			
STATUT	varchar(50)	latin1_swedish_ci		Non			
Date_extreme_validite	varchar(50)	latin1_swedish_ci		Non			
Nbr_jour_retard_rapatriement	varchar(50)	latin1_swedish_ci		Non			
QTE_EXPORTER	varchar(50)	latin1_swedish_ci		Non			
Date_rapatriement	varchar(50)	latin1_swedish_ci		Non			
Fonds_rapatriee_a_ce_jour	varchar(50)	latin1_swedish_ci		Non			
Prefinancement	varchar(50)	latin1_swedish_ci		Non			
SOLDE_Montant	varchar(50)	latin1_swedish_ci		Non			
SOLDE_QTE_Restant	varchar(50)	latin1_swedish_ci		Non			
Jour_ecoule	varchar(50)	latin1_swedish_ci		Non			
echeance	varchar(50)	latin1_swedish_ci		Non			

La structure de notre table utilisateur pour la gestion d'accès des utilisateurs dans notre application



Affichage des données pour la table utilisateur

Affichage des données de la table EB

Les codes associés

Code « Ajout »

```
<?php
session_start();
$base=mysql_connect('localhost','root','');
mysql_select_db('licences',$base) or die(mysql_error());
$duree_mois = 3;
$duree_jour = 10;
$duree_jour_autres = 30;
$date_du_jour = date ("d-m-Y");
$date=$_POST['date'];
$num_licence=$_POST['num_licence'];
$exportateur=$_POST['exportateur'];
$acheteur=$_POST['acheteur'];
$devise=$_POST['devise'];
$montant=$_POST['montant'];
$nature=$_POST['nature'];
```

```
$quantite=$_POST['quantite'];
$poids=$_POST['poids'];
$destination=$_POST['destination'];
$date_embarquement=$_POST['date_embarquement'];
$fonds_repartie=$_POST['fonds_repartie'];
$qte_exporter=$_POST['qte_exporter'];
$prefinancement=$_POST['prefinancement'];
$date_embarquementTimestamp=strtotime($date_embarquement);
//date extreme
$date_extreme=date('d-m-Y', strtotime('+'.$duree_mois.'month',
$date_embarquementTimestamp));
//date rapatriement
if($nature=='DIAMAND'){
        $date_rapatriement=date('d-m-Y', strtotime('+'.$duree_jour.'day',
$date_embarquementTimestamp));
}
else{
        $date_rapatriement=date('d-m-Y', strtotime('+'.$duree_jour_autres.'day',
$date_embarquementTimestamp));
}

$date_extreme_nombre=strtotime($date_extreme);
$date_today=strtotime($date_du_jour);
//solde montant
$solde_montant=$montant - $fonds_repartie;
//comparaison temps pour le statut
if($date_today< $date_extreme_nombre){
                if($solde_montant == $montant ){
                        $statut='encours/payé totalement';
                        }
                else{
                        $statut='encours/payé partiel';
                }
```

```
}
else{
        if($solde_montant == $montant ){
                $statut='expiré/payé totalement';
                }
            else{
                $statut='expiré/payé partiel';
            }
}
//nombre des jours en retard
$date_rapatriement_ennombre=strtotime($date_rapatriement);
$nbre_jour=round(($date_rapatriement_ennombre - $date_today)/86400);
$solde_qte_restant=$quantite - $qte_exporter;
//jour ecoulé
$jour_ecoule=round(($date_embarquementTimestamp- $date_today)/86400);
$echeance=$date_embarquement;
$rekette = 'INSERT INTO eb VALUES("'.$date.'", "'.$num_licence.'",
"'.$exportateur.'", "'.$acheteur.'", "'.$devise.'", "'.$montant.'", "'.$nature.'",
"'.$quantite.'", "'.$poids.'", "'.$destination.'", "'.$date_embarquement.'",
"'.$statut.'", "'.$date_extreme.'", "'.$nbre_jour.'", "'.$qte_exporter.'",
"'.$date_rapatriement.'", "'.$fonds_repartie.'", "'.$prefinancement.'",
"'.$solde_montant.'", "'.$solde_qte_restant.'", "'.$jour_ecoule.'", "'.$echeance.'")';
mysql_query ($rekette) or die ('ErreurSQL !'.$rekette.'<br />'.mysql_error());
mysql_close();

echo '<body onLoad="alert(\'Données EB insérées...\')">';
echo '<meta http-equiv="refresh" content="0;URL=eb.php">';
?>

<?php
session_start();
?>
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
```

```
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr">
<head>
<title>.::SOFIBANK-GESTION DES LICENCES::.</title>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
<link rel="stylesheet" media="screen" type="text/css" title="style_didier"
href="style.css" />

Code Remboursement
</head>
<body>
<div class="corps">
        <div class="un">
                <div class="un_gauche"><?phpecho $_SESSION['nom']; ?></div>
                <div class="un_droit">
                <?php
                        $date_du_jour = date ("d-m-Y");
                        echo $date_du_jour;
                        $base=mysql_connect('localhost','root','');
                        mysql_select_db('licences',$base) or die(mysql_error());
                ?>
                </div>
        </div>
        <div class="deux">
        <div class="deux_gauche">
        <ul>
                <li><a href="eb.php">EB</a></li>
                <li><a href="es.php">ES</a></li>
                <li><a href="ib.php">IB</a></li>
                <li><a href="is.php">IS </a></li>
                <li><a href="rc.php">RC </a></li>
        </ul>
```

```
</div>
<div class="deux_droit">
<div class="sous_menu">
<ul>
<li><a href="eb.php">RAPPATRIEMENT</a></li>
<li><a href="eb_rembousprefi.php">REMBOURS.
PREFINANCEMENT</a></li>
<li><a href="eb_modification.php">MODIFIER</a></li>
<li><a href="eb_recherche.php">RECHERCHE</a></li>
</ul></div>
<div class="fomulaire_selection">
<!--<form action="ajout_eb_prefi.php" method="post" >-->
<table>
<form action="validation_rc.php" method="post">
<tr><td><span class="nom_user">FORMULAIRE
EB</span></td><td></td></tr>
<tr><td><span class="nom_user">DATE DE
VALIDITE:</span></td><td><input type="text" name="date" value=""
size="30"/></td></tr>
<tr><td><span class="nom_user">NumLicence:</span></td><td><input
type="text" name="num_licence" value="" size="30"/></td></tr>
<tr><td><span class="nom_user">EXPORTATEUR:</span></td><td><input
type="text" name="exportateur" value="" size="30"/><input type="submit"
value="Lier RC" /></td></tr></form>
<tr><td><span class="nom_user">RC:</span></td><td>
<select id="form_promo" name="licence_rc">
                    <option></option>
                    </select></td></tr>
<tr><td><span class="nom_user">ACHETEUR:</span></td><td><input
type="text" name="acheteur" value="" size="30"/></td></tr>
<tr><td><span class="nom_user">DEVISE:</span></td><td><input
type="text" name="devise" value="" size="30"/></td></tr>
<tr><td><span class="nom_user">MONTANT:</span></td><td><input
type="text" name="montant" value="" size="30"/></td></tr>
```

```
<tr><td><span class="nom_user">NATURE:</span></td><td>
<select id="operations" name="nature">
        <option value="DIAMAND">DIAMAND</option>
        <option value="BOIS">BOIS</option>
        <option value="AUTRES">AUTRES</option>
</select></td></tr>
<tr><td><span class="nom_user">TarifDouanier:</span></td><td><input
type="text" name="tarif" value="" size="30"/></td></tr>
<tr><td><span class="nom_user">Quantité:</span></td><td><input
type="text" name="quantite" value="" size="30"/></td></tr>
<tr><td><span class="nom_user">Poids:</span></td><td><input
type="text" name="poids" value="" size="30"/></td></tr>
<tr><td><span class="nom_user">Destination:</span></td><td><input
type="text" name="destination" value="" size="30"/></td></tr>
<tr><td><span class="nom_user">Date
Embarquement:</span></td><td><input type="text"
name="date_embarquement" value="" size="30"/></td></tr>
<tr><td><span
class="nom_user">QuantitéExporter:</span></td><td><input type="text"
name="qte_exporter" value="" size="30" /></td></tr>
<tr><td><span class="nom_user">FondsRapatriee a
cejour:</span></td><td><input type="text" name="fonds_repartie" value=""
size="30" /></td></tr>
<tr><td><span
class="nom_user">Préfinancement:</span></td><td><input type="text"
name="prefinancement" value="" size="30" /></td></tr>

<tr><td><input type="reset" value="Ajouter EB" /></td></tr>
                        </table>
                </form>
                </div>
        </div>
</div>
<div class="un">
```

```
                <div class="un_gauche"></div>
                <div class="un_droit">
                <?php
                        echo '<a href="logout.php">Déconnection</a>';
                ?>
                </div>
        </div>
</div>
</body>
</html>
```

Code Recherche

```
<?php
session_start();
// Connexion MySQL
// Obligatoire pour la suite !
$base=mysql_connect('localhost','root','');
mysql_select_db('licences',$base) or die(mysql_error());
// la variable qui va contenir les données CSV
$outputCsv = '';
// Nom du fichier final
$fileName = 'exportateur-csv.csv';
$date_debut=$_POST['date_debut'];
$date_fin=$_POST['date_fin'];
$statut=$_POST['statut'];
$exportateur=$_POST['exportateur'];
if ($statut == 'encour_total'){
        $statut_bd = 'encours/payé totalement';
}
else if ($statut == 'encour_partiel'){
        $statut_bd = 'encours/payé partiel';
}
```

```
else if ($statut == 'expire_total'){
            $statut_bd = 'expiré/payé totalement';
}
else {
            $statut_bd = 'expiré/payé partiel';
}
$requete='SELECT
Date,Num_Licence,Exportateur,Acheteur,Devise,Montant,Nature,QTE,POIDS,Destination,Date_Embarquement,STATUT,Date_extreme_validite,Nbr_jour_retard_rapatriement,QTE_EXPORTER,Date_rapatriement,Fonds_rapatriee_a_ce_jour,Prefinancement,SOLDE_Montant,SOLDE_QTE_Restant,Jour_ecoule,echeance  FROM  eb
WHERE  Exportateur="'.$exportateur.'" or Num_Licence="'.$exportateur.'"';
//$requete = "SELECT * FROM user ORDER BY nom";
$sql = mysql_query($requete);
if(mysql_num_rows($sql) > 0)
{
   $i = 0;
while($Row = mysql_fetch_assoc($sql))
{
      $i++;
      // Si c'est la 1er boucle, on affiche le nom des champs pour avoir un titre pour chaque colonne
if($i == 1)
      {
foreach($Row as $clef => $valeur)
            $outputCsv .= trim($clef).';';
         $outputCsv = rtrim($outputCsv, ';');
$outputCsv .= "\n";
      }
      // On parcours $Row et on ajout chaque valeur à cette ligne
foreach($Row as $clef => $valeur)
$outputCsv .= trim($valeur).';';
      // Suppression du ; qui traine à la fin
```

```
$outputCsv = rtrim($outputCsv, ';');
// Saut de ligne
      $outputCsv .= "\n";
   }
}
else
exit('Aucune donnée à enregistrer.');
// Entêtes (headers) PHP qui vont bien pour la création d'un fichier Excel CSV
header("Content-disposition: attachment; filename=".$fileName);
header("Content-Type: application/force-download");
header("Content-Transfer-Encoding: application/vnd.ms-excel\n");
header("Pragma: no-cache");
header("Cache-Control: must-revalidate, post-check=0, pre-check=0, public");
header("Expires: 0");
echo $outputCsv;
exit();
?>
```

(Réserver au lecteur) Concevoir une page web qui est constituée d'une zone d'affichage où on peut observer les différents pourcentages des choix opérés par les visiteurs concernant le niveau d'appréciation du site et par ailleurs d'un formulaire contenant des rubriques suivantes : nom, prénom, ville, âge, sexe (Masculin et féminin), niveau d'appréciation du site (mauvais, médiocre, assez bon, bon, très bon, excellent). Les données remplies peuvent être soumises ou annulées. Après toute soumission, le tableau des pourcentages est tout de suite mis à jour. (Une base de données en MySql est chargée de conserver les informations envoyées par les visiteurs).

Annexes
(Rappels sur XHTML)

Rappels sur le html

1. Présentation du html

Le HTML (HyperText Markup Language) comme langage de marquage hypertexte constitue le langage de base du Web (World Wide Web). Ce n'est pas un langage de programmation proprement dit, mais plutôt un code de marquage qui permet de décrire la page Web élément par élément en se servant de balises de description.

Le code HTML est déchiffré par le navigateur du client. Tous les navigateurs existant reconnaissent parfaitement la syntaxe HTML.

Les pages web faites uniquement avec HTML sont suffixées par l'extension .htm ou encore .html.

Le XHTML (eXtensible Hypertext Markup Language) comme langage de balisage hypertexte extensible est une recommandation du W3C (www Consortium). Il s'agit d'une version de HTML respectant la syntaxe XML (eXtensible Markup Language, langage HTML amélioré permettant de définir de nouvelles balises) et de laquelle sont exclues toutes les imprécisions que l'on rencontre généralement dans les pages web.

L'édition d'une page web en code HTML ou XHTML est tellement fastidieuse. Il existe des logiciels d'édition des pages Web (éditeurs web) qui permettent de générer le code HTML tout en offrant au développeur une interface conviviale en mode création (outils de traitement de texte et de mise en page, outils de dessin…). Par exemple, les éditeurs professionnels suivants : DreamWeaver, FrontPage, Golive, WebExpert, Nvu…

2. Syntaxe html

Le langage HTML est un langage de marquage qui ne connaît que l'alphabet ASCII standard, limité à 128 caractères. Le marquage, réalisé par des balises, décrit la structure logique du document et est interprété par les logiciels de navigation (navigateurs ou browsers).

- Une balise prend la forme suivante:

<balise> objet décrit par la balise </balise>

Exemple : Programmation web

Sur le navigateur, on peut visualiser le texte « Programmation web » placé entre les balises écriture en gras)

- Certaines balises HTML s'écrivent sous la forme :

<balise attribut1=''valeur1'' attribut2=''valeur2'' … >

Objet décrit par la balise

</balise>

Exemple:

Bonjour

Font étant la balise de mise en forme du texte, sur le navigateur, on visualise « Bonjour » avec la police verdana, la couleur bleue et la taille 3.

- Il est possible de regrouper plusieurs balises à la fois, et il est recommandé que la dernière balise ouverte soit la première qui doit être fermée.

Exemple :   <u> Bonjour </u>

- En HTML les commentaires sont délimités par <!-- et --> tout ce qui se trouve à l'intérieure des délimiteurs est ignoré par le navigateur.

Exemple:

Bonjour

<!-- Les balises et mettent 'Bonjour' en gras -->

- Les balises **obligatoires** au sein d'un document HTML sont :

<html></html> : début et fin du document HTML

<head></head> : Entêtes du document

<body></body> : corps du document (c'est la partie visualisée sur le navigateur)

Typiquement, un document HTML se présente avec la structure suivante :

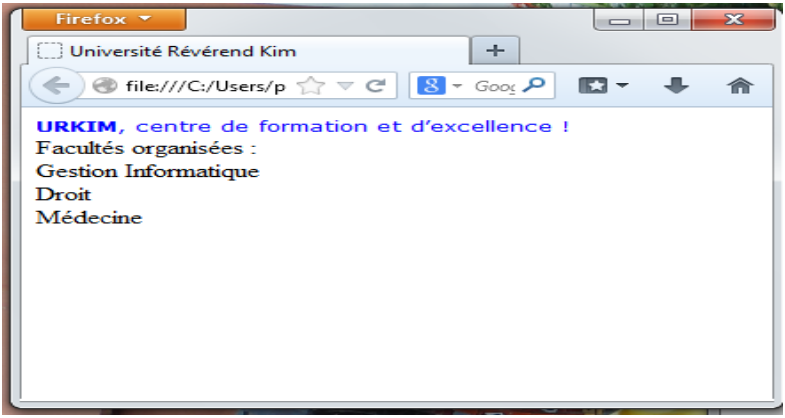
```
<html>
        <head>
```

```
        <title>Titre de la page</title>
    </head>
    <body>
        Contenu de la page
    </body>
</html>
```

Exemple d'un document HTML :

```
<html> <!-- Début du document nommé test.htm -->
<head>
<title>Université Révérend Kim</title> <!-- Titre du document -->
</head>
<body> <!-- La partie du document affichée sur le Navigateur -->
<font face="verdana" size="2" color="#0000FF">
<b>URKIM</b>, centre de formation et d'excellence !<br/>
</font>
Facultés organisées : <br/>
Gestion Informatique <br/>
Droit <br/>
Médecine <br/>
</body>
</html>
```

Saisir ce code sur bloc notes, puis l'enregistrer avec l'extension html (ou htm). Par exemple index.html.



La balise
 permet d'effectuer le saut de ligne.

Remarque : Avec XHTML, toutes les balises HTML peuvent être employées, mais leur nom doit être saisi en minuscules, par exemple h1 et non H1. Une déclaration de DOCTYPE (Document Type) doit être présente, référençant un des documents suivants : strict, transitional ou frameset.

Quelques balises de base

Liste des balises de base

Balise	Description
<title></title>	Titre du document (affiché sur la barre de titre)
<meta>	Echange d'infos avec le navigateur (sans </meta>)
<script>	Insertion de script (JavaScript, VbScript…)
<bgsound>	Fond sonore de la page (sans </bgsound>)
	Nouveau paragraphe
 	Saut de ligne (sans </br>)
	Insertion d'une ligne horizontale après l'élément.
	Texte en gras
	Texte en italique
<u></u>	Texte souligné

<s></s>	Texte barré
<marquee></marquee>	Texte défilant (souvent utilisé sur les pages web)
,	Liste ordonnée, et liste non ordonnée.
<a>	Lien hypertexte (ou hyperlien)
<img/>	Insertion d'image (sans </img>)
	Mise en forme du texte (police, couleur et taille)
	Mise en forme (alignement centré, justifié…)
	Décoration du texte (surlignage…)
<table></table>	Définition d'un tableau HTML
<tr></tr>	Définition d'une ligne du tableau
<td></td>	Définition d'une colonne du tableau
<form></form>	Création d'un formulaire

Les attributs importants des quelques balises de base

- Balise <body> (corps du document)

La balise **body** permet de définir le contenu de la page Web. Ses attributs associés sont :

- Bgcolor : Couleur de l'arrière plan de la page. Cet attribut peut prendre comme valeur le nom de la couleur tel que green, white, red, pink … Cependant, le codage de couleurs demeure la solution la plus adéquate pour exprimer toutes les nuances disponibles dans l'environnement RVB (ou RGB en anglais).

- Background : Image d'arrière plan de la page

- Topmargin, Bottommargin, Leftmargin, Rightmargin: Marges de la page respectivement supérieure, inférieure, gauche et droite.

- Text : Couleur par défaut du texte de la page

- Link : Couleur par défaut des liens hypertextes sur la page

- Alink : Couleur par défaut des liens hypertextes actifs (qui ont le focus)

- Vlink : Couleur par défaut des liens hypertextes visités

Si la couleur des liens et textes n'est pas définie, ceux-ci prennent les couleurs par défaut déclarées dans la balise body.

Exemple :

<body topmargin=0 leftmargin=0 bgcolor=''#0000ff'' text=''blue''>

…

</body>

- Balise <marquee>

La balise **marquee** permet d'insérer un texte (ou une image) défilant sur la page Web. Cette balise n'est pas reconnue par Netscape qui ne rendra son contenu qu'en élément fixe et immobile.

Les attributs associés sont :

- Id: identificateur de l'élément

- Behavior: comportement de l'élément (alternate, slide, scroll)

- Direction : sens de défilement (left, right, up, down)

- Loop: nombre du défilement (si rien n'est mentionné: infinie)

- Scrollamount: fixe le pas de déplacement en pixel (par défaut 6)

- Scrolldelay: vitesse de défilement (par défaut 90)

- Width: largeur de l'élément (zone réservée au défilement)

- Height: hauteur de l'élement

Exemple : <marquee direction="right"> URKIM, centre de formation et d'excellence !</marquee>

- Balise <p>

La balise **p** permet de définir un nouveau paragraphe. Ses attributs associés sont :

- Style=''margin-left: n px'' marge gauche

- Style=''margin-right: n px'' marge droite

- Style=''margin-top: n px'' marge supérieure

- Style=''margin-bottom: n px'' marge inférieure

Avec n un nombre entier définissant la valeur de la marge en pixel.

Exemple:

<p style=''margin:10px''> Une marge de 10 pixels est définie en haut, en bas, à gauche et à droite </p>

- Balises <ol> et

Les balises **o>** et **ul** permettent de définir respectivement la liste numérotée (ordonnée) et non numérotée. Ses attributs associés sont :

- Type : prenant les valeurs (disc, circle, square) pour une liste à puces. Et les valeurs (1, 2, … ou a, b, …ou A, B, … ou i, ii, … ou I, II, …) pour une liste numérotée.

- Start : détermine la valeur du début de la liste.

Chaque élément de la liste est placé dans ….

Exemple : Définir les différentes facultés en liste à puces.

- Balise (surlignage)

La balise **span** utilise une information de style locale de type:

- Style=''background-color:couleur'' : marquage du texte

- Style=''text-decoration:underline'' : soulignage du texte

- Style=''text-decoration:overline'' : surlignage du texte

Exemple :

<span style=''background-color:#00ff00; text-decoration:overline''>

Ce texte est marqué en vert et est surligné

- Balise <img/>

La balise **img** permet d'insérer une image extension jpg, png ou gif dans une page web. Ses attributs associés sont :

- Src : chemin relatif (ou absolu) de l'image

- Border : taille de la bordure de l'image en pixel

- Width : largeur de l'image en pixel

- Height : hauteur de l'image en pixel

- Alt (ou Title) : description de l'image qui s'affiche sur une info bulle

Exemple:

- Balise <bgsound>

L'élément **bgsound**, placé entre <head> et </head>, permet de créer des pages avec un fond sonore et de paramétrer la durée de diffusion, le volume et la balance.

Cet élément admet les types de fichiers sons extension .wav, .au et .mp3 et les séquences d'extension .mid.

Les attributs associés sont :

- Id: identificateur de l'élément

- Src: chemin relatif du fichier son

- Loop: nombre de répétitions (-1 veut dire infini)

- Volume: volume du son entre -10 000 et 0 (0: maximum)

- Balance: balance entre les deux haut-parleurs entre -10 000 et +10 000 (-10 000: le son provient entièrement du haut-parleur gauche)

N.B. : La balise <embed> est utilisée avec les mêmes attributs pour insérer dans une page web un fichier audio ou vidéo (le multimédia). Son attribut autostart, qui a pour valeurs true et false, spécifie si l'élément va se lancer automatiquement à l'affichage de la page web ou pas.

- Balise <a>

La balise **a** permet de définir un lien hypertexte vers une autre page ou un autre site.

- Href: lien de destination (chemin relatif de la page appartenant au site courant ou chemin absolu commençant par http:// si la page existe sur un autre site Web)

- Target: carde de destination (fenêtre courante ou nouvelle fenêtre)

- Title: titre affiché dans l'info bulle si l'on survole le lien avec le pointeur de la sourie.

Quelques valeurs courantes de l'attribut « target »:

- _self: même cadre

- _blank: nouvelle fenêtre

- _top: page entière

Exemple : Créer page2.html qui sera liée à « Gestion informatique » de la page index.html.

Gestion Informatique

- Balises <table> <tr> et <td>

La balise **table** permet de créer un tableau HTML dont les lignes sont définies à l'aide de la balise tr et les colonnes à l'aide de la balise td.

Les tableaux HTML permettent d'établir une mise en page facile de la page Web. Elles permettent de diviser la page en plusieurs partie, chacune accueille un objet (text, script, image, objet de formulaire…)

<table> (balise de déclaration d'un tableau)

- Id : identificateur du tableau

- Border : taille de la bordure du tableau

- Width : largeur du tableau en pixel ou en pourcentage

- Height : hauteur du tableau en pixel ou en pourcentage

- Cellspacing : espacement entre cellules en pixel

- Cellpadding : marge intérieure des cellules en pixel

- Bgcolor : couleur de l'arrière plan du tableau

- Bordercolor : couleur de la bordure du tableau

<tr> (balise de définition d'une nouvelle ligne du tableau)

- Bgcolor: couleur de l'arrière-plan de la ligne

- Width: largeur de la ligne en pixel ou en pourcentage (par défaut c'est la largeur définie en paramètres de la balise <table>

- Height: hauteur de la ligne en pixel ou en pourcentage

<td> (balise de définition d'une nouvelle colonne de la ligne)

- Bgcolor: couleur de l'arrière-plan de la colonne

- Width: largeur de la colonne en pixel ou en pourcentage

- Height: hauteur de la colonne en pixel ou en pourcentage

- Align: alignement horizontal des objets de la colonne

- Valign: alignement vertical des objets de la colonne

Exemple :

<table border=''0'' width=''760'' cellspacing=''0'' cellpadding=''0''>

<tr>

<td bgcolor=''blue'' height=''50'' width=''380'' valign=''top''>

C'est la colonne 1 de la ligne 1 du tableau 1

</td>

<td bgcolor=''green'' height=''50'' width=''380'' valign=''top''>

C'est la colonne 2 de la ligne 1 du tableau 1

</td>

</tr>

</table>

- Balise meta

La balise **meta** qui ne possède pas de balise de fin, est toujours déclarée à l'entête du document (entre <head> et </head>).

Les balises meta permettent de donner des informations sur la page (description, auteur, outils…). Elles sont souvent utilisées pour permettre aux moteurs de recherche de mieux analyser le contenu d'une page et de faciliter la recherche sur Internet en présentant des sujets connexes au contenu du document, l'auteur du document, une description courte du document…

Chaque balise meta possède un nom qui caractérise sa fonction :

Infos sur le créateur ou le propriétaire du site

- Balise meta retournant le nom du créateur du site

<meta name=''author'' lang=''fr'' content=''nom prénom''>

Lang: spécifie la langue utilisée sur la page courante.

- Balise meta retournant le nom du publieur du site (propriétaire du site)

<meta name=''publisher'' content=''nom prénom''>

Exemple :

<head>

<meta name=''author'' lang=''fr'' content=''Batu Jude''>

<meta name=''publisher'' content=''Labo NTIC''>

</head>

Infos sur le site

Balise meta retournant les mots-clés sur lesquels se basent les moteurs de recherche pour référencer la page courante.

<meta name=''keywords'' content=''mots clés séparés par des virgules''>

Balise meta retournant la description du site

<meta name=''description'' content=''brève description du site''>

Balise meta donnant l'URL complet de votre site

<meta name=''identifier-URL'' content=''URL complet du site''>

Exemple:

<head>

<meta name=''keywords'' content=''Informatique, Rdf, Biométrie, Empreintes''>

<meta name=''description'' content=''Reconnaissance des formes biométriques''>

<meta name=''identifier-URL'' content=''http://www.laboinfo.unikin.cd''>

</head>

Infos sur la création

Balise meta retournant les copyrights

<meta name=''copyright'' content=''text du copyright''>

Balise meta retournant les outils de développement.

<meta name=''generator'' content=''liste des logiciels utilisés à la création''>

Balise meta retournant la date de la création de la page.

<meta name=''date'' content=''date de création de la page''>

Exemple:

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

Orienter les robots (robots de moteurs de recherche)

Si vous désirez que le robot indexe toutes les pages de votre site, c'est-à-dire toutes celles vers lesquels vous avez placé des liens, utilisez la propriété All :

<meta name=‘’robots’’ content=‘’all’’>

Si le site est en construction et que vous voulez que le robot n’indexe pas vos pages vous mettez ‘’none’’ à la place de ‘’all’’. Avec ‘’follow’’ les destinations des liens sont indexées. ‘’Nofollow’’ empêche le robot d’indexer ces destinations. Avec ‘’Index’’ seule la page courante est indexée. ‘’NoIndex’’ La page courante n’est pas indexée

Vous pouvez aussi demander aux robots de réindexer automatiquement votre site après n jours :

<meta name="Revisit-after" content="n">

Exemple:

<meta name="Robots" content=‘’Index,Follow">

<meta name="Revisit-after" content="5">

Rafraîchissement et redirection

Pouvoir faire recharger une page périodiquement peut s'avérer très utile surtout si l'on affiche des bannières publicitaires qu’on veut changer souvent. Ainsi, il est possible d'ordonner au navigateur de recharger une page toutes les n secondes grâce à l'attribut Refresh :

<meta http-equiv="Refresh" content="n">

On peut même procéder au chargement d'une page différente, très utile lorsqu'on change d'hébergeur afin de rediriger le navigateur vers un autre site:

<meta http-equiv="Refresh" content="n; url=url souhaité">

Exemple :

<meta http-equiv="Refresh" content="60; url=http://www.google.com’’>

Date d'expiration

Vous savez qu'il existe des ordinateurs qui, pour faciliter l'accès aux pages web, conserve dans leurs disques durs les pages déjà visitées et donc susceptibles d'être redemandées. Ainsi, lorsqu'on accède à une seconde reprise au même site, on a des chances de retomber sur les mêmes pages déjà visitées. Ainsi, lorsqu'on veut empêcher cela, on indique une date d'expiration qui, si elle est dépassée, ordonnera au Proxy d'aller récupérer les vrais pages et de supprimer celles expirées. Attention, la date est au format anglophone!

<meta http-equiv="Expires" content=''Sat, 2 Jun 2008 14:30:00">

En plus, vous pouvez interdire aux navigateurs de conserver en mémoire-cache vos pages:

<meta http-equiv="Pragma" content="no-cache">

- Les formulaires

Les formulaires HTML (Forms en anglais) sont des ensembles de composants, appelés aussi champs, qui permettent à l'utilisateur d'entrer des informations, d'exprimer ses choix, de saisir du texte…

En général, un site dynamique contient des formulaires, car les champs de formulaires permettent à l'internaute de communiquer avec le site (Il ne se contente pas de voir le contenu en passant d'un lien à autre).

La balise **<form> …</form>** déclare un formulaire sur la page web.

Tous les champs (zone de texte, boutons, listes de choix, cases à cocher…) doivent être placés entre <form> et </form>.

N.B. : Une page peut contenir plusieurs formulaires à la fois, ceux-ci sont généralement traités par des scripts tels que JavaScript ou PHP…

Balise:

<form name=''nom_formulaire'' Method=''méthode d'envoi'' Action=''URL''>

Name : qui spécifie le nom du formulaire, est utile si on utilise plusieurs formulaires sur la même page.

Method (valeurs possibles post et get) : **post** permet l'envoi des valeurs du formulaire dans l'entête du document (les valeurs ne sont pas visibles) ; alors que **get** les envoie avec l'URL (les valeurs sont visibles sur la barre d'adresse, ce qui

peut compromettre la confidentialité des données envoyées tels que les mots de passe).

Action : Spécifie la page qui se charge du traitement du formulaire.

Le champ TEXTE

Syntaxe : <input type=''text''>

Ses attributs sont :

Name : nom du champ texte (Identificateur du champ)

Value : valeur par défaut

Size : taille en caractère

Tabindex : ordre de tabulation (définit l'ordre de déplacement du curseur entre les champs de formulaire suite à l'appuie sur la touche TABULATION du clavier).

Exemple :

<input type=''text'' name=''login'' size=''20'' tabindex=''1''>

Le champ ZONE DE MOT DE PASSE

Syntaxe : <input type=''password''>

Ses attributs sont :

Name : nom du champ texte

Value : valeur par défaut

Size : taille en caractère

Tabindex : ordre de tabulation

Exemple :

<input type=''text'' name=''login'' size=''20'' tabindex=''2''>

Le champ ESPACE DE TEXTE

Syntaxe : <textarea></textarea>

Ses attributs sont :

Name : nom de l'espace de texte

Colls : largeur de l'espace de texte en caractères

Rows : hauteur de l'espace de texte en lignes

Tabindex : ordre de tabulation

Exemple:

<textarea name=''commentaire'' colls=''30'' rows=''6'' tabindex=''3''>

Ceci est un commentaire

</textarea>

Le champ BOUTON RADIO

Syntaxe : <input type=''radio''>

Ses attributs sont :

Name : nom du bouton radio

Value : valeur alphanumérique de l'élément

checked : dit si l'élément est sélectionné

Tabindex : ordre de tabulation

Noter que les boutons radio forment des groupes. C'est-à-dire qu'ils portent le même nom mais on les différencie par leurs valeurs.

Exemple :

<input type="radio" name=''rad'' value=''1'' checked>

<input type="radio" name=''rad'' value=''2''>

Le champ CASE A COCHER

Syntaxe : <input type=''checkbox''>

Ses attributs sont :

Name: nom de la case à cocher

Value : valeur alphanumérique de l'élément

Checked : dit si l'élément est sélectionné

Tabindex : ordre de tabulation

Les cases à cocher peuvent être traitées comme des éléments indépendants, des groupes d'éléments ou encore des tableaux d'éléments (apprécié en PHP)

Exemple :

<input type=''checkbox'' name=''che'' value=''1'' checked>

<input type=''checkbox'' name=''che'' value=''2''>

Le champ BOUTON

Syntaxe : <input type=''type_de_bouton''>

Il existe trois types :

- type=''submit'' ce sont les boutons d'envoie de formulaires

- Type=''button'' boutons ordinaires (peuvent être personnalisés avec

JavaScript)

- Type=''reset'' boutons rétablir (rétablie les valeurs par défaut des champs de formulaire)

Ses attributs sont :

Type: type de bouton

Name: nom du bouton

Value: label du bouton (texte écrit dessus)

Tabindex: ordre de tabulation

Exemple:

<input type=''button'' name=''imp'' value=''Imprimer cette page''>

<input type=''submit'' name=''valider'' value=''Envoyer''>

<input type=''reset'' name=''RES'' value=''Rétablir le formulaire''>

Si le bouton est de type submit (le plus utilisé d'ailleurs), le fait de cliquer dessus redirige le navigateur vers la page définie en valeur de l'attribut action de la balise form. En effet, c'est le bouton submit qui soumet le formulaire au traitement.

Le champ LISTE DE SELECTION

Syntaxe :

<select name=''nom_de_l_element''>

<option>option1</option>

<option>option2</option>

…

</select>

Les attributs de la balise select sont :

Name : nom de la liste

Tabindex : ordre de tabulation

Et les attributs de la balise option sont :

Value : valeur de l'option (par défaut c'est le contenu de l'option)

Selected : mentionne si l'option est sélectionnée par défaut.

Exemple :

<select name=''secteur_activité''>

<option selected>Télécoms</option>

<option>Électricité</option>

<option>Informatique</option>

<option>Mécanique</option>

<option>Métallurgie</option>

</select>

L'attribut value de la balise option est facultatif. Si rien n'est mentionné alors le contenu de la balise passe en valeur de l'attribut value.

Le champ CHARGEMENT DE FICHIER

Syntaxe : <input type=''file''>

Ses attributs sont :

Name : nom du champ

Tabindex : ordre de tabulation

Les champs **file** permettent de parcourir un fichier sur le poste de travail et le soumettre au formulaire. Ce dernier, grâce à des scripts coté serveur comme PHP, charge le fichier sur le serveur (UPLOAD)

Exemple :

<input type= "file" name="mon_fichier">

Exemple d'un formulaire :

```
<html>
<head>
<title>Université Révérend Kim</title>
</head>
<body>
<center>
<form name="nom_formulaire" Method="get" Action="mbukanga@yahoo.fr">
Enregistrement d'un utilisateur
<table border=0>
<tr>
        <td>Nom</td>
        <td>
        <input type="text" name="login" size="20" tabindex="2">
        </td>
</tr>
<tr>
        <td>Sexe</td>
<td>
        Homme : <input type=radio name="sexe" value="M" checked>
        <br/>Femme : <input type=radio name="sexe" value="F">
        </td>
</tr>
<tr>
        <td>Les loisirs que vous intéressent :</td>
</tr><tr>
```

```
<td> </td><td>
        <input type="checkbox" name="choix" value="1"> Sport<br/>
        <input type="checkbox" name="choix" value="2">Théatre <br/>
        <input type="checkbox" name="choix" value="3" >Musique
        </td>
</tr>
<tr></tr>
<tr>
        <td>Fonction :</td>
        <td>
        <select name="fonction">
                <option value="enseignant">Enseignant</option>
                <option value="etudiant">Etudiant</option>
                <option value="ingenieur">Ingénieur</option>
                <option value="retraite">Retraité</option>
                <option value="autre">Autre</option>
        </select>
        </td>
</tr>
<tr>
        <td>Commentaires :</td>
        <td>
        <textarea name="commentaire" colls="30" rows="6" tabindex="3">
Ceci est un commentaire
</textarea>
        </td>
</tr>
<tr>
```

```
        <td>Joindre un fichier :</td>
        <td>
        <input type="file" name="mon_fichier">
        </td>
</tr>
<tr>
        <td align=center>
        <input type="submit" value="Envoyer">
        </td><td align=center>
        <input type="reset" value="Rétablir">
        </td>
</tr>
</table>
</form>
</center>
</body>
</html>
```