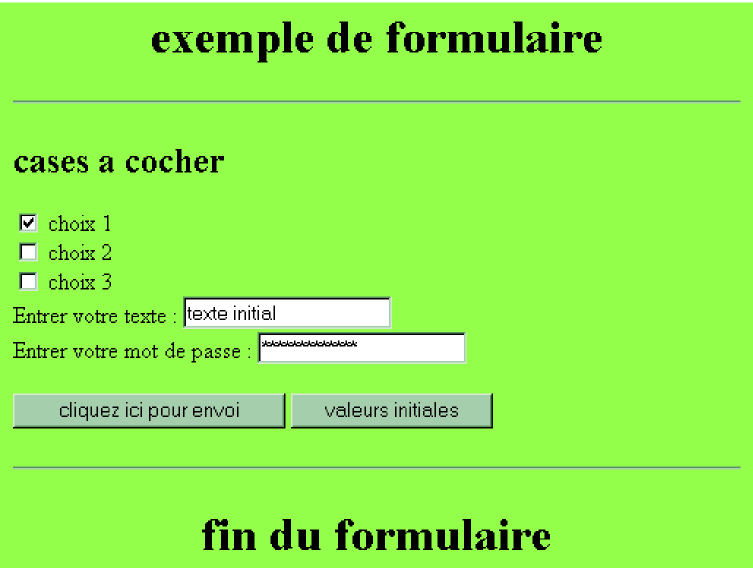
Séance d'Exercices Dirigés

HTML, CGI et PHP

Exercice 1 : Formulaire et script CGI

Soit le formulaire suivant rempli par un utilisateur :



1°) Construire la page HTML qui présente ce formulaire

Une réponse :
Il est obtenu par le document HTML :

```
<HTML>
<HEAD>
  <TITLE>Essai de script cgi</TITLE>
</HEAD>
<BODY>
<H1 ALIGN=CENTER>exemple de formulaire</H1>
<HR WIDTH="100%">

<FORM A_COMPLETER_QUESTION_SUIVANTE>
<H2>cases a cocher</H2>
<INPUT TYPE="CHECKBOX" NAME="case1" VALUE="la case 1 est cochee"> choix 1<BR>
<INPUT TYPE="CHECKBOX" NAME="case2"> choix 2<BR>
<INPUT TYPE="CHECKBOX" NAME="case3"> choix 3<BR>
Entrer votre texte : <INPUT TYPE="TEXT" NAME="wtexte" VALUE="texte initial"><BR>
Entrer votre mot de passe : <INPUT TYPE="PASSWORD" NAME="wpass" VALUE="texte initial"><BR>
```

```
<P>
<INPUT TYPE="SUBMIT" NAME="Envoi" VALUE="cliquez ici pour envoi"> <INPUT TYPE="RESET"
NAME="Reset" VALUE="valeurs initiales"><BR>
</FORM>
<HR WIDTH="100%">
<H1 ALIGN=CENTER>fin du formulaire</H1>
</BODY>
</HTML>
```

2°) On veut que, lorsque l'utilisateur clique sur le bouton "cliquez ici pour envoyer", ce formulaire demande à lancer le programme
`post-query` sur la machine serveur `cedric.cnam.fr.`
Compléter le formulaire précédent : on précisera les attributs ACTION et METHOD de la balise FORM, le programme `post-query` a été placé à l'emplacement convenu des scripts CGI du serveur web.

Une réponse :
Le document ci dessus a été compléter (en gras ci dessous).

```
<HTML>
<HEAD>
  <TITLE>Essai de script cgi</TITLE>
</HEAD>
<BODY>
<H1 ALIGN=CENTER>exemple de formulaire</H1>
<HR WIDTH="100%">

<FORM METHOD="POST" ACTION="http://cedric.cnam.fr/cgi-bin/post-query">
<H2>cases a cocher</H2>
<INPUT TYPE="CHECKBOX" NAME="case1" VALUE="la case 1 est cochee"> choix 1<BR>
<INPUT TYPE="CHECKBOX" NAME="case2"> choix 2<BR>
<INPUT TYPE="CHECKBOX" NAME="case3"> choix 3<BR>
Entrer votre texte : <INPUT TYPE="TEXT" NAME="wtexte" VALUE="texte initial"><BR>
Entrer votre mot de passe : <INPUT TYPE="PASSWORD" NAME="wpass" VALUE="texte initial"><BR>
<P>
<INPUT TYPE="SUBMIT" NAME="Envoi" VALUE="cliquez ici pour envoi"> <INPUT TYPE="RESET"
NAME="Reset" VALUE="valeurs initiales"><BR>
</FORM>
<HR WIDTH="100%">
<H1 ALIGN=CENTER>fin du formulaire</H1>
</BODY>
</HTML>
```

3°) En utilisant les données tapées par l'utilisateur, indiquer sous quelle forme le serveur Web reçoit ces données.

Une réponse :
En utilisant les noms des champs ci dessus et les données tapées par l'utilisateur, il a été envoyé au serveur, la chaîne :

```
case1=la+case+1+cochee&wtexte=sqfdkhfkq&wpass=bonjour&Envoi=cliquez+ic
i+pour+envoi
```

4°) En supposant que la méthode GET a été utilisé, quel URL a construit le client Web ? Comment le script CGI récupère les valeurs tapées par l'utilisateur.
En supposant que la méthode POST a été utilisé, préciser comment les valeurs tapées par l'utilisateur ont été envoyées au serveur Web puis comment le serveur Web les passe au script CGI.

Une réponse :
Si c'est la méthode GET qui a été utilisée, les données ont été envoyée codées dans l'URL qui est alors :
`http://cedric.cnam.fr/cgi-bin/post-query?case1=la+case+1+cochee&wtexte=sqfdkhfkq&wpass=bonjour&Envoi=cliquez+ici+pour+envoi`

Le serveur Web lance le processus CGI `post-query` après lui avoir préparé un environnement contenant des variables dont la variable `QUERY_STRING` qui contient la valeur
`case1=la+case+1+cochee&wtexte=sqfdkhfkq&wpass=bonjour&Envoi=cliquez+ici+pour+envoi`
Il suffit que le script `post-query` lise cette variable pour récupérer les valeurs saisies par l'utilisateur.

Si c'est la méthode POST, la requête a été envoyée au serveur Web sous la forme :

```
POST /cgi-bin/post-query  HTTP/1.0
                une ligne blanche
case1=la+case+1+cochee&wtexte=sqfdkhfkq&wpass=bonjour&Envoi=cliquez+ici+pour+envoi
```

Le serveur Web lance le script CGI après avoir passé à l'entrée standard de ce script la chaine :
`case1=la+case+1+cochee&wtexte=sqfdkhfkq&wpass=bonjour&Envoi=cliquez+ici+pour+envoi`
Il suffit alors que le script fasse une analyse de ce que lui donne l'entrée standard (découpage en paquet de caractères séparé par & puis découpage en paquets séparés par =).

5°) On veut maintenant que le script CGI lancé par la méthode POST,renvoie une page HTML qui affiche les couples `nomVariable, valeur`. Ecrire en langage C un tel script CGI.

une réponse :
Voici un script complet (donné par le serveur Web NCSA) qui fait ce travail. Il peut être écrit de manière beaucoup plus simple. Il utilise les routines langage C comme :

fmakeword(stdin,'&',&cl); qui lit les couples jusqu'au caractère &

plustospace(entries[x].val); qui convertit les + en espaces (le client Web ayant fait la conversions contraire car les espaces sont.

unescape_url(entries[x].val); qui convertit les codes hexadéc. en caractères pour les caractères spéciaux des données de l'utilisateur (l'utilisateur a tapé & ou =)

entries[x].name = makeword(entries[x].val,'='); qui sépare nom et valeur lié par le symbole =.

ainsi que la variable CONTENT_LENGTH qui indique le nombre de caractères envoyés par le serveur Web au script.

```
#include <stdio.h>
#include <stdlib.h>

#define MAX_ENTRIES 10000

typedef struct {
    char *name;
    char *val;
} entry;

char *makeword(char *line, char stop);
char *fmakeword(FILE *f, char stop, int *len);
char x2c(char *what);
void unescape_url(char *url);
void plustospace(char *str);

main(int argc, char *argv[]) {
    entry entries[MAX_ENTRIES];
    register int x,m=0;
    int cl;

    printf("Content-type: text/html%c%c",10,10);
if(strcmp(getenv("REQUEST_METHOD"),"POST")) {
        printf("This script should be referenced with a METHOD of POST.\n");
        printf("If you don't understand this, see this ");
        printf("<A HREF=\"http://www.ncsa.uiuc.edu/SDG/Software/Mosaic/Docs/fill-out-
forms/overview.html\">forms overview</A>.%c",10);
        exit(1);
    }
if(strcmp(getenv("CONTENT_TYPE"),"application/x-www-form-urlencoded")) {
        printf("This script can only be used to decode form results. \n");
        exit(1);
    }
    cl = atoi(getenv("CONTENT_LENGTH"));
    for(x=0;cl && (!feof(stdin));x++) {
        m=x;
/* lit les couples suivants jusqu'à & */
        entries[x].val = fmakeword(stdin,'&',&cl);
/* convertit les + en espaces */
        plustospace(entries[x].val);
/* convertit les codes hexadéc. en caractères */
        unescape_url(entries[x].val);
/* sépare nom et valeur */
     entries[x].name = makeword(entries[x].val,'=');
    }

    printf("<H1>Query Results</H1>");
    printf("You submitted the following name/value pairs:<p>%c",10);
    printf("<ul>%c",10);

    for(x=0; x <= m; x++)
        printf("<li> <code>%s = %s</code>%c",entries[x].name,
               entries[x].val,10);
    printf("</ul>%c",10);
}
```

Exercices sur PHP

Au sommaire de cet ED :

1. **Introduction avec la notion de modularisation, utilisée notamment pour la réalisation des interfaces utilisateurs,**
2. **Exercice avec la modularisation du site Web "Mode",**
3. **Accès et utilisation d'une base de données avec l'exemple de l'authentification.**

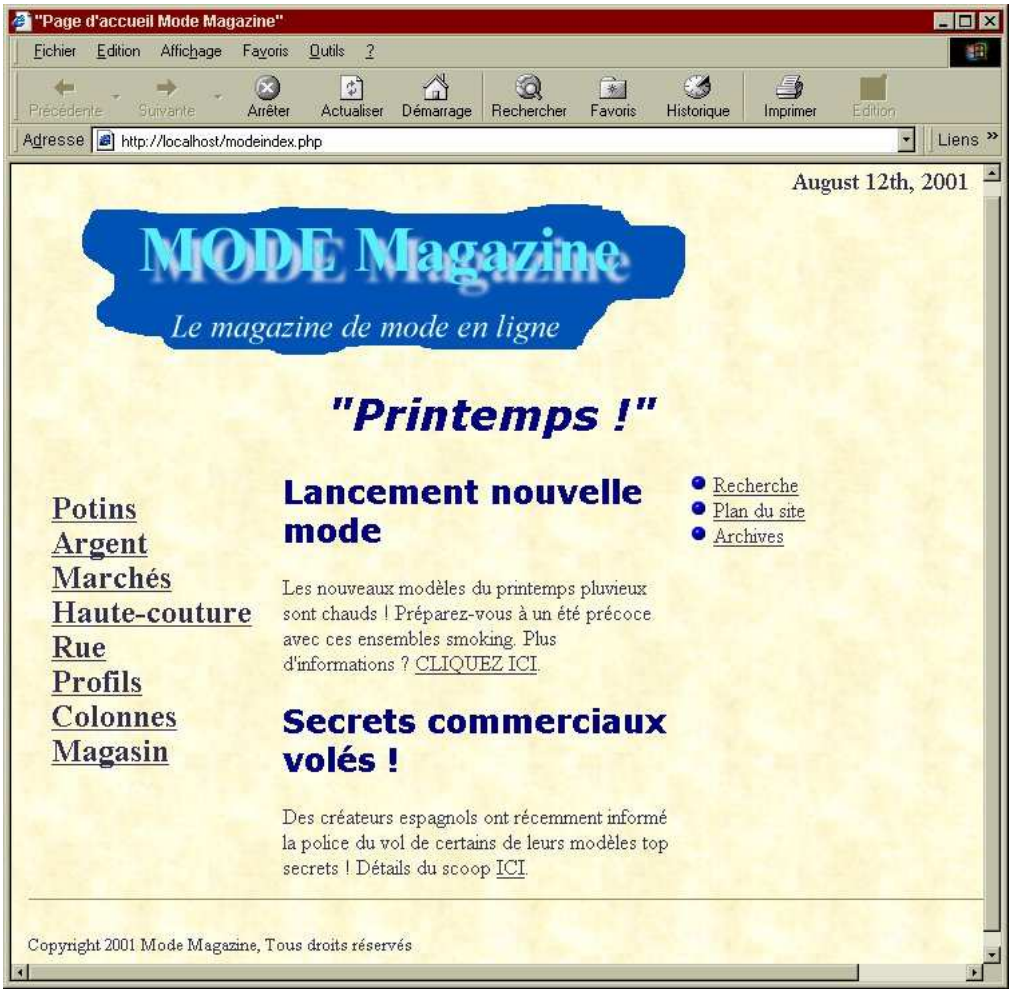
Enoncés

1. La notion de modularisation

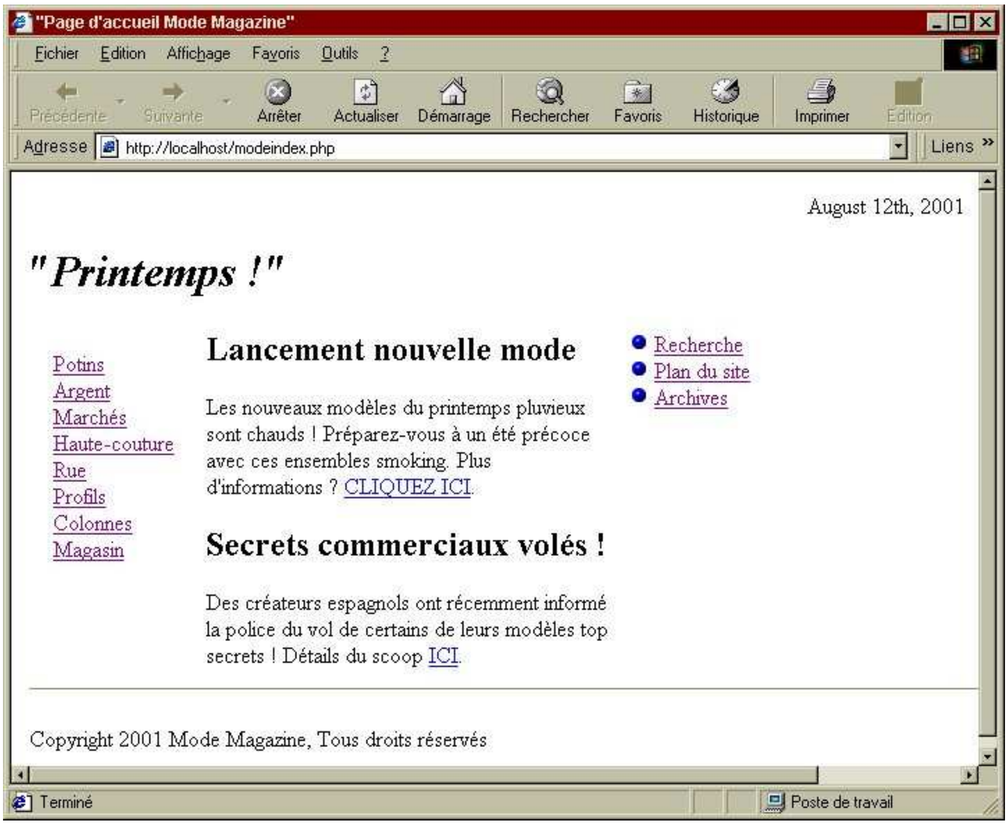
Que faut-il entendre par "modularisation" ?
Quelle est l'intérêt de la modularisation des pages Web ?
Introduction au mécanisme de la modularisation.

2. Exercice : modulariser le site Web de la société "Mode"

Cet exercice est un exemple de site modulaire. Il s'agit de la page d'accueil du magazine de mode en ligne fictif dont le titre est "Mode Magazine". Toute la page est modularisée. Les seuls composants de la page qui ne se trouvent pas dans un module sont le titre de la page, la manchette (*Nouveautés*) et le contenu du centre de la page. La figure suivante montre la présente page d'accueil.

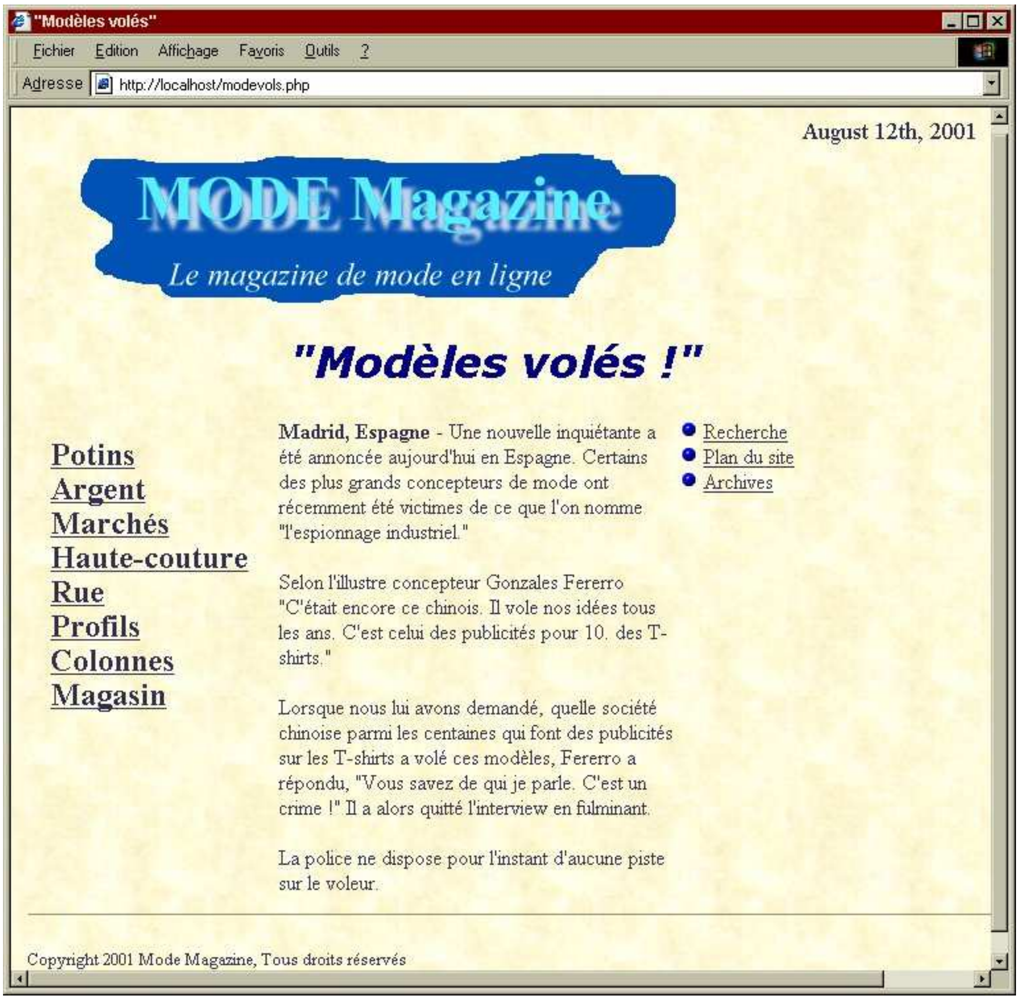


La Figure ci-après montre la même page pour laquelle la feuille de style et les graphismes ont été placés en commentaires dans le script pour qu'ils n'apparaissent pas à l'écran. Remarquez la différence qu'engendre la mise en commentaires de quelques lignes. Imaginez ce que vous pouvez faire si vous pensez à remodeler votre site et qu'il soit, comme dans cet exemple, configuré de manière modulaire. Vous pourriez remanier votre site entièrement et lui appliquer de nouvelles polices et autres graphismes en quelques minutes.



Plusieurs scripts composent le modèle de ce site, chacun ayant un but précis et certains étant des modules d'autres scripts. Combinés, ils constituent le squelette du site Web. Les voici :

- **modepage.php : il s'agit des pages de contenu qui accueillent toutes les autres pages comme, par exemple, *modeindex.php* et *modevols.php*.**
- **principal.php : il s'agit du fichier principal appelé par les fichiers ci-dessus.**
- **entete.php : cette partie imprime les informations d'entête HTML ainsi que le titre de chaque page. Elle affiche également la date en cours dans la partie supérieure droite de la page ainsi que l'image du logo Mode.**
- **style.php : il s'agit de la page de style du site. Elle est appelée dans *entete.php* pour être incluse dans la zone <head> HTML des pages de la partie *principal*.**
- **liensnav.php : ce fichier contrôle les liens de navigation qui s'affichent sur le côté gauche de la page.**
- **recherche.php : ce fichier contrôle les liens Recherche, Plan du site et Archives qui s'affichent sur le côté droit de la page.**
- **pieddepage.php : cette page contrôle le pied de page de chaque page. Sur ce site, elle contient une ligne et les informations de copyright.**



Question :

En vous aidant des informations précédentes et de la figure ci-dessus, essayez de déterminer l'architecture du présent modèle afin de voir comment chaque élément y est assemblé et commencez a écrire le code des scripts énoncés précédemment.

3. Authentification de l'utilisateur

La gestion d'une base de données avec PHP est simple. On peut ajouter ou supprimer des données à la base à partir de n'importe quel navigateur Web. Toutefois, cette simplicité a un revers puisque des personnes non autorisées peuvent modifier votre base de données. La solution consiste à mettre en Place une authentification simple des utilisateurs.

PHP comprend des méthodes simples qui fonctionnent parfaitement pour les sites Web ordinaires ne contenant pas de données sensibles, comme un bulletin d'informations ou un petit site d'articles. Le but de cet exercice est de montrer l'accès à une base de données MySql au travers de l'acte d'authentification de l'utilisateur.

Soit par exemple la table des utilisateurs suivante :

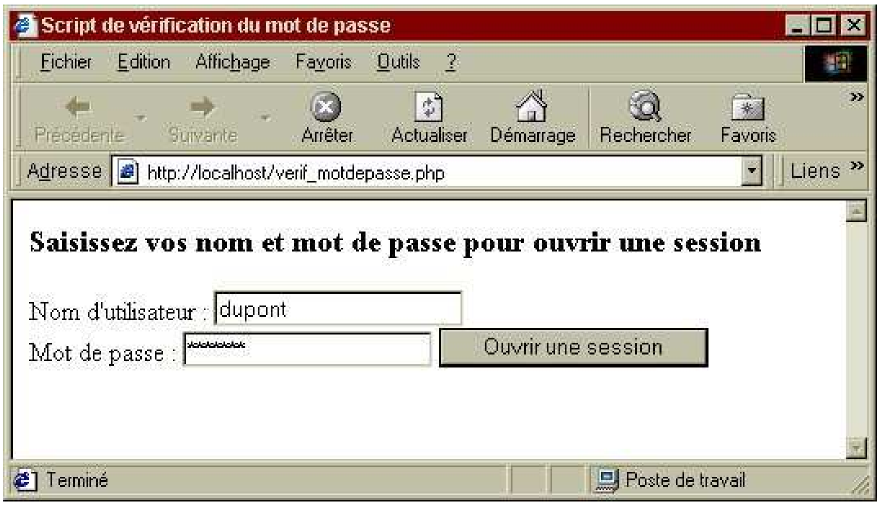
MYSQL> describe utilisateurs;

Field	Type	Null	Key	Default	Extra
id_utilisateur	Int(11)		PRI	0	Auto_increment
nom_utilisateur	varchar(32)	YES		NULL	
email_utilisateur	varchar(64)	YES		NULL	
motdepasse_utilisateur	varchar(8)	YES		NULL	

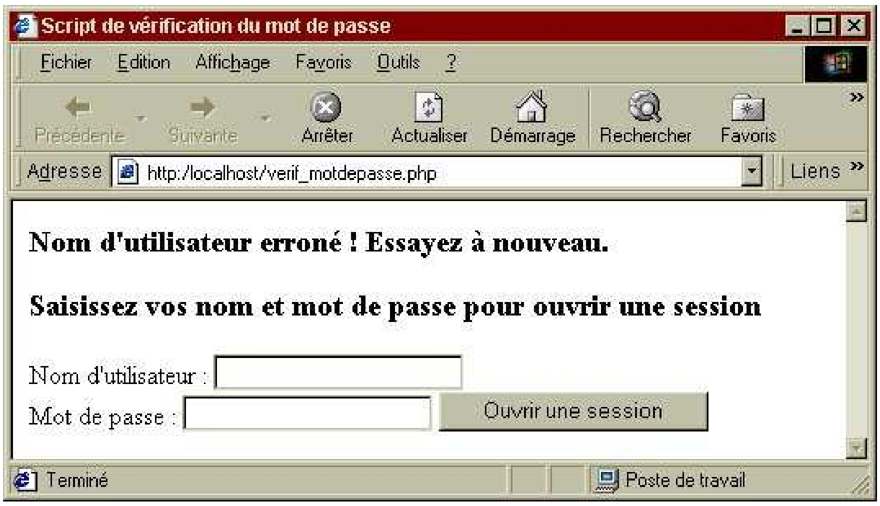
Le site Web est composé d'un script nommé *verif_motdepasse.php* qui sert à vérifier le mot de passe d'un utilisateur. Si le mot de passe est correct, le script retourne un message indiquant qu'il est accepté. Dans le cas contraire, le script retourne un message signalant que le mot de passe est erroné et invite à nouveau l'utilisateur à saisir son adresse électronique et son mot de passe.

Les figures ci-après décrivent le principe de fonctionnement.

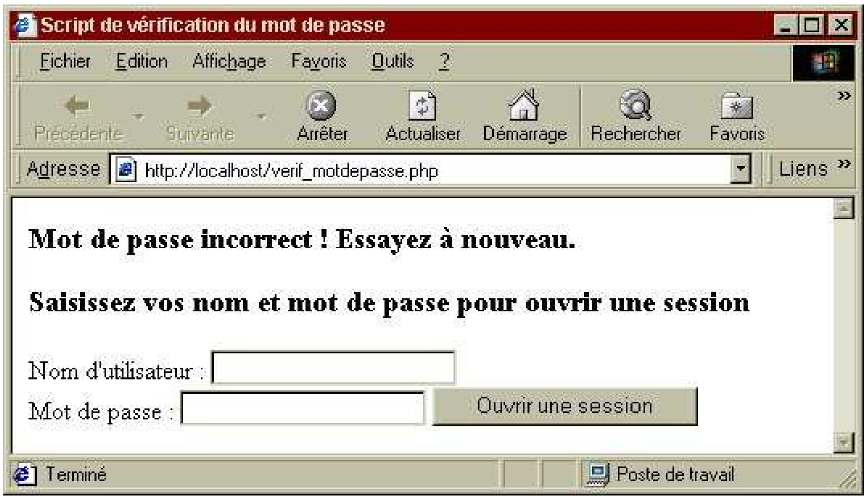
<u>Etape 1</u> : Chargement initial de verif_motdepasse.php.



<u>Etape 2</u> : Le nom d'utilisateur saisi est incorrect



<u>Etape 3</u> : Le mot de passe saisi est incorrect.



<u>Etape 4</u> : Connexion correcte.

Question :

Etudiez et écrivez le code du script *verif_motdepasse.php*.

Réponses

1. La modularisation

Il n’est rien de plus simple, grâce à l’utilisation des modèles, que de créer plusieurs pages de structure et de disposition identiques, tout en variant leur contenu. Lorsqu'on construit un site Web à partir d'un modèle, il faut modulariser les parties similaires du site.

La puissance de la modularité permet de simplifier la tâche du webmaster notamment lors des opérations de maintenance. Il suffit pour cela de construire un site dynamique, en intégrant des pages qui contiennent des composants modulaires.

Ces composants sont de petits objets "auto-contenants" qui composent un objet de plus grande taille. Une page Web représente l'objet principal. Le site Web quant à lui, obéit à une charte graphique. Il est composé de différents éléments tels des barres de navigation, des icônes ou autres images, et leur disposition est généralement identiques sur toutes les pages du site.

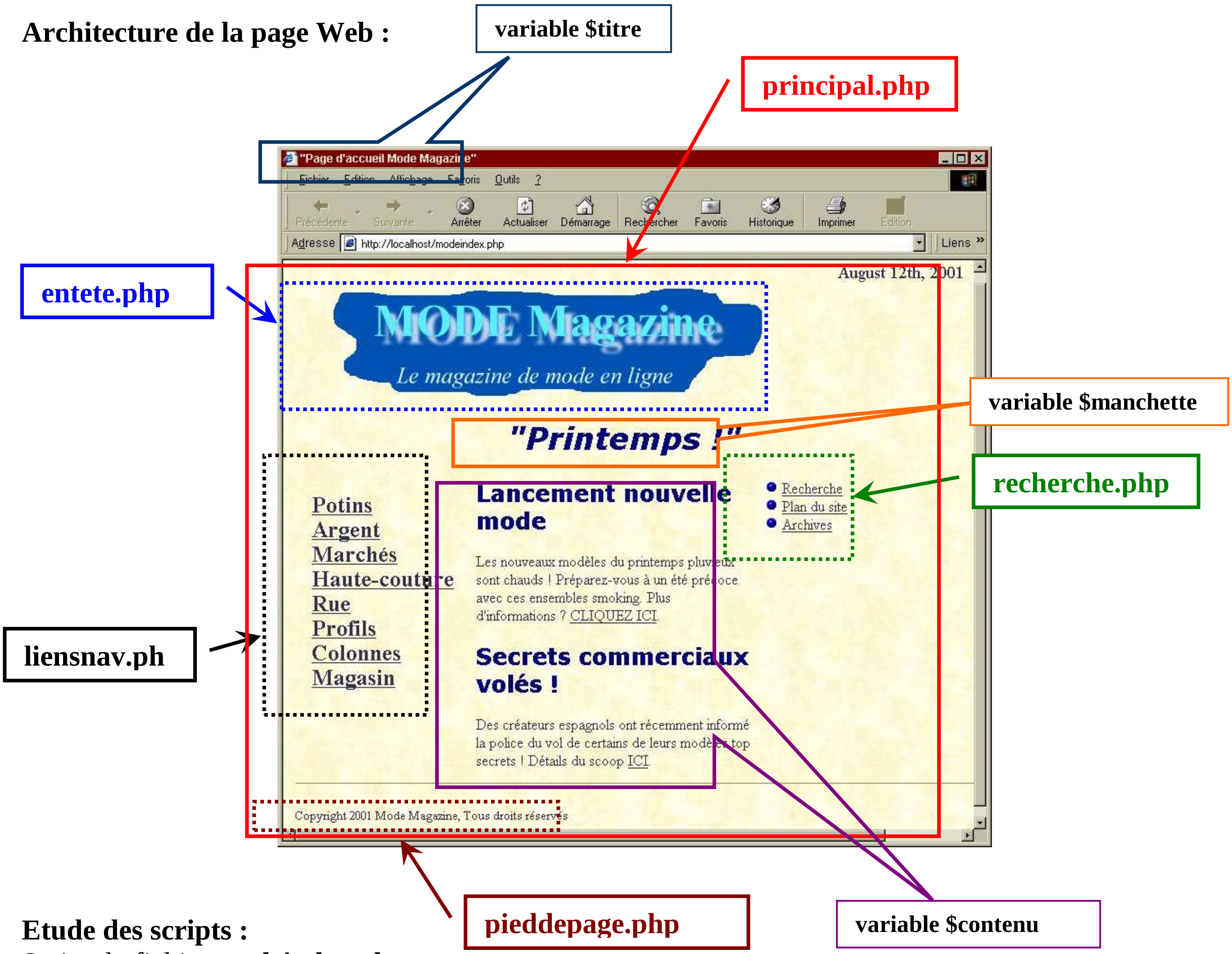
Il est possible, en outre, de prévoir un graphisme récurrent en entête de chaque page ou en pied de page, pour y accueillir les informations de copyright par exemple. Tous ces éléments peuvent être transformés en un module unique puis référencés sur toutes les autres pages.

Grâce à la modularisation, on obtient une cohérence visuelle de l'ensemble du site. Pour effectuer des changements, il suffit de modifier un fichier, l'ensemble des changements est ensuite reporté dans toutes les pages lors de la génération de la page Web par le serveur PHP.

2. Exercice : modulariser le site Web de la société "Mode"

Cet exercice initie aux concepts d'un site modulaire et en donne un exemple.

Architecture de la page Web :



Etude des scripts :

Script du fichier **modeindex.php**

```
<?
// Spécifier $titre, $manchette et $contenu
$titre = "Page d'accueil de MODE Magazine";
$manchette = "Printemps !";
$contenu = "
<h2>Lancement nouvelle mode</h2>
<p>Les nouveaux modèles du printemps pluvieux sont chauds.
Préparez-vous à un été précoce avec ces ensembles .smoking.
Plus d'informations ? <a href=\"#\">CLIQUEZ ICI</a>.
<h2>Secrets commerciaux volés !</h2>
<p>Des créateurs espagnols ont récemment informé la police
du vol de certains de leurs modèles top secrets.
Détails du scoop <a href=\"modevols.php\">ICI</a>.
";
include("principal.php");
?>
```

Script du fichier **principal.php**

```
<?
include("entete.php');
head("$titre");
'?>
<table width="800">
<tr><td>
<? headline($manchette); ?>
<!-- Liens de navigation -->
 <? include("liensnav.php") ?>
<table>
<tr><td width='300">
<!-- Insérer le contenu ici -->
<? print $contenu ?>
</td>
<td valign-"TOP">
<!-- Liens Recherche, Plan du site, Archives -->
<? include("recherche.php") ?>
</tr></td>
</table>
</td></tr>
<tr><td>
<!-- Insérer le pied de page ici -->
<? include("pieddepage.php") ?>
</tr></td>
</table>
</body>
</html>
```

Script du fichier **entete.php**

```
<?
function head($titre) {
?>
<html>
<head>
<title><? print $titre ?></title>
<!-- Insérer style ici -->
<? include("style.php") ?>
</head>
<body background="mode_ap.gif">
<div align="RIGHT"><? print date("F jS, Y") ?></div>
<img src="modelogo.gif" width=516 height=122>
<p>
<?
}
function headline($manchette)
?>
<hl><i><? print $manchette ?></i></hl>
<?
}
?>
```

Script du fichier : **liensnav.php**

```
<table cellpadding="15" align="LEFT">
<tr><td>
<a  href="#"  class="large">Potins</a><br>
<a  href="#"  class='large'>Argent</a><br>
<a  href="#'  class="large">Marchés</a><br>
<a  href="#"  class="large'>Haute couture</a><br>
<a  href="0"  class='large">Rue</a><br>
<a  href="#"  class="large">Profils</a><br>
<a  href="#"  class="large">Colonnes</a><br>
<a  href="#"  class="large">Magasin</a><br>
</tr></td>
</table>
```

Script du fichier : **recherche.php**

```
<img src="puce.gif" width=5 height=5> <a href="#"> Recherche</a><br>
<img src="puce.gif" width=5 height=5> <a href="#"> Plan du site</a><br>
<img src="puce.gif" width=5 height=5> <a href="#"> Archives</a>
```

Script du fichier : **pieddepage.php**

```
<hr size="1" noshade>
<p class="small"> Copyright 2001 Magazine MODE, Tous droits réservés</P>
```

Script de la feuille de style : **style.php**

```
<style><!--
P {
font-size : small;}
H1 {
font-family : Verdana, Arial;
text-align : center ;
color : Navy;}
H2 {
font-family : Verdana, Arial;
color : Navy;}
H3 {
font-family : Verdana, Arial;}
A {
color : #302e52;}
A:Visited {
color : #302e52;}
A:Active {
color : #302e52;}
BODY {
font-size : medium;
font-family : Georgia, Times;
color : #302e52;}
A.large {
font-size : large;
color : #302e52;
font-weight : bold;}
A:Visited.large {
font-size : large;
color : #302e52;
font-weight : bold;}
A:Active.large    {
font-size : large;
color : #302e52;
font-weight : bold;}
P.small {
font-size : x-small;}
</style>
```

Script du fichier : **modevols.php**

```
<?
// Spécifiez $titre, $manchette et $contenu
$titre = "Modèles volés !";
$manchette = "Modèles volés !";
$contenu = "
<p><b>Madrid, Espagne</b> - Une nouvelle inquiétante a été annoncée aujourd'hui en Espagne.
Certains des plus grands concepteurs de mode ont récemment été victimes de ce que l'on nomme &quot;
l'espionnage industriel.&quot;
<p>Selon l'illustre concepteur Gonzales Fererro &quot; C'était encore ce chinois. Il vole nos idées tous les ans. C'est celui des publicités pour
des T-shirts.&quot;
<p>Lorsque nous lui avons demandé, quelle société chinoise parmi les centaines qui font des publicités sur les T-shirts a volé ces modèles,
Fererro a répondu, &quot;Vous savez de qui je parle. C'est un crime !&quot;
Il a alors quitté l'interview en fulminant.
<p>La police ne dispose pour l'instant d'aucune piste sur le voleur.
";
include("principal.php");
?>
```

Principe de fonctionnent :

Etape 1.

L'une des pages *content*, comme *modeindex.php*, est appelée par le serveur Web. La page de contenu se développe en trois points :

- le titre de la page (`$titre`) ;
- la ligne d'entête de la page (`$manchette`) ;
- le contenu de la page (`$contenu`).

Etape 2.
Une fois ces points définis, le script appelle le modèle de mise en forme principal du site : *principal.php*. Il détermine où ont affichés les autres fichiers *include* de la page. Le fichier *principal.php* héberge la plupart des fichiers de l'ensemble des fichiers. Il contient :

- *entete.php* ;
- *liensnav.php* ;
- *recherche.php* ;
- *pieddepage.php*.

En outre, le fichier *principal.php* affiche le contenu de la variable $contenu de l'étape 1. Les deux autres variables de l'étape 1, $titre et $manchette sont envoyées à *entete.php* pour être exploitées dans la fonction head( ). Le fichier *entete.php* doit appeler *style.php* avant de pouvoir retourner le résultat dans principal.php.

Etape 3.
Les fichiers *recherche.php*, *liensnav.php* et *pieddepage.php* sont affichés directement à partir de ce fichier. Aucun de ces fichiers ne doit inclure d'autres fichiers à inclure ici. Le fichier *entete.php* retourne l'entête HTML (y compris $titre) ainsi que la date encours, le logo qui s'affiche dans la partie supérieure de la page, l'image d'arrière-plan et la variable $manchette. Le fichier *principal.php* appelle *style.php* pour afficher la feuille de style.

En conclusion

La création d'un site modulaire se traduit par un gain de temps. Pourquoi actualiser constamment un grand nombre de fichiers alors que l'on peut actualiser tout le site en quelques secondes ? En outre, un site modulaire reste cohérent au fil des pages. Si vous modifiez les liens de navigation, ils le sont pour toutes les pages sur lesquelles ils sont présents.

Quelques conseils

Saisissez les scripts et essayez-les. Modifiez le fichier *style.php* et agissez sur l'apparence du site. Modifiez également le fichier *principal.php* pour diversifier la disposition du site. Et pour finir, essayez de créer plusieurs fichiers conteneurs (modeindex.php, modevols. php) pour voir comment le contenu s'adapte au reste de la page.

3. Authentification de l'utilisateur

Le Script suivant sert à vérifier le mot de passe d'un utilisateur. Si le mot de passe est correct, le script retourne un message indiquant qu'il est accepté. Dans le cas contraire, le script retourne un message signalant que le mot de passe est erroné et invite à nouveau l'utilisateur à saisir son adresse électronique et son mot de passe. Les exemples sont illustrés par les copies d'écran de l'énoncé.

Script du fichier : **verif_motdepasse.php**

```
<html>
<head>
<title>Script de vérification du mot de passe</title>
</head>
<body>
<?php
function print_form( ) {
	?>
	<form action="verif_motdepasse.php" method="POST">
	<h3>Saisissez vos nom et mot de passe pour ouvrir une session</h3>
	Nom d’utilisateur : <input type="text" name="nom_utilisateur">
	<br>Mot de passe : <input type="password" name="motdepasse">
	<input type="submit" name="soumettre" value="Ouvrir une session">
	</form>
	<?
}
if(isset($soumettre)):
	if(!$bd = mysql_connect("localhost","root")):
		print("<hl>Connexion à la base de données impossible !</hl>\n");
	else:
		mysql_select_db("php', $bd);
	endif;
	$sql = "select * from utilisateurs where nom_utilisateur = '$nom_utilisateur'";
	$resultat = mysql_query($sql);
	$compte_lignes = mysql_num_rows($resultat);
	if($compte_lignes == 0):
		?>
		<h3>Nom d'utilisateur erroné ! Essayez à nouveau.</h3>
		<?
		print_form( );
	else:
		$ligne = mysql_fetch_array($resultat);
		if($motdepasse != $ligne["motdepasse_utilisateur"]):
			?>
			<h3>Mot de passe incorrect ! Essayez à nouveau.</h3>
			<?
			print_form( );
		else:
			?>
			<h3>Mot de passe accepté !</h3>
			<?
		endif
	endif;
else:
	print_form( );
endif;
?>
</body>
</html>
```

Comment fonctionne le script :

- ✓ Les lignes 1 à 5 sont en langage HTML standard.
- ✓ La ligne 6 accueille la balise de début PHP. A partir de là, le serveur Web lit la page comme du code PHP au lieu du langage HTML.
- ✓ Les lignes 7 à 16 contiennent une fonction qui imprime le formulaire de connexion de la page. Ce code est imbriqué dans une fonction dans la mesure où il peut être appelé depuis différentes parties du script.
- ✓ La ligne 18 vérifie si l'utilisateur a appuyé sur le bouton "Soumettre" du formulaire. Si c'est le cas, on peut lire les données saisies dans le formulaire. Dans le cas contraire, le script passe directement à la ligne 45.
- ✓ Si l'utilisateur a appuyé sur le bouton "Soumettre", les lignes 19 à 23 tentent d'établir la connexion avec le serveur MYSQL. En cas de problème de connexion à ce serveur, un message d'erreur s'affiche. S'il n'y a pas d'erreur lors de la connexion à la base de données, le script accède à la base de données spécifiée.
- ✓ Les lignes 24 à 26 interrogent la base de données et comptent le nombre de lignes du jeu de résultats.
- ✓ Les lignes 27 à 31 vérifient si la requête SQL a retourné une ligne. Si aucune ligne n'est retournée, le script suppose que le nom de l'utilisateur n'est pas valide dans la mesure où le script se sert du nom de l'utilisateur saisi dans le formulaire comme base de recherche de la ligne appropriée. Dans ce cas, le script affiche un message d'erreur indiquant à l'utilisateur que le nom saisi est incorrect avant d'afficher à nouveau le formulaire.
- ✓ Dans les lignes 32 à 38, le mot de passe est comparé au résultat de la requête pour vérifier la présence d'une correspondance entre la base de données et le nom d'utilisateur. Si le mot de passe saisi dans le formulaire ne correspond pas au mot de passe de la base de données, un message d'erreur et le formulaire s'affichent à l'écran.
- ✓ Dans les lignes 39 à 43, si le mot de passe est correct, l'utilisateur en est averti et l'accès lui est autorisé. C'est à partir de ce point que vous pouvez insérer les informations protégées par mot de passe.
- ✓ La ligne 44 contient l'instruction "endif" de l'instruction "if /then/else" commencée à la ligne 27.
- ✓ Les lignes 45 à 47 sont exécutées si l'utilisateur n'a pas appuyé sur le bouton "Soumettre". En fait, elles supposent que c'est la première fois que l'utilisateur affiche la page au cours de la session et qu'il doit saisir les informations d'authentification.
- ✓ La ligne 48 accueille la balise PHP de fin. Le serveur Web arrête de lire la page en PHP et la lit à nouveau au format HTML.
- ✓ Les lignes 49 à 50 contiennent le code HTML standard de fermeture de page.