

## Complément du cours : Chapitre 5

**Tableau 1 : Variables prédéfinies**

<code>\$GLOBALS</code>	Contient le nom et la valeur de toutes les variables globales du script. Les noms des variables sont les clés de ce tableau. <code>\$GLOBALS["mavar"]</code> récupère la valeur de la variable <code>\$mavar</code> en dehors de sa zone de visibilité (dans les fonctions, par exemple).
<code>\$_COOKIE</code>	Contient le nom et la valeur des cookies enregistrés sur le poste client. Les noms des cookies sont les clés de ce tableau Avant PHP 4.1, cette variable se nommait <code>\$HTTP_COOKIE_VARS</code> .
<code>\$_ENV</code>	Contient le nom et la valeur des variables d'environnement qui sont changeantes selon les serveurs. Avant PHP 4.1, cette variable se nommait <code>\$HTTP_ENV_VARS</code> .
<code>\$_FILES</code>	Contient le nom des fichiers téléchargés à partir du poste client. Avant PHP 4.1, cette variable se nommait <code>\$HTTP_FILES_VARS</code> .
<code>\$_GET</code>	Contient le nom et la valeur des données issues d'un formulaire envoyé par la méthode GET. Les noms des champs du formulaire sont les clés de ce tableau Avant PHP 4.1, cette variable se nommait <code>\$HTTP_GET_VARS</code> .
<code>\$_POST</code>	Contient le nom et la valeur des données issues d'un formulaire envoyé par la méthode POST. Les noms des champs du formulaire sont les clés de ce tableau. Avant PHP 4.1, cette variable se nommait <code>\$HTTP_POST_VARS</code> .
<code>\$_REQUEST</code>	Contient l'ensemble des variables superglobales <code>\$_GET</code> , <code>\$_POST</code> , <code>\$_COOKIE</code> et <code>\$_FILES</code> . Avant PHP 4.1, cette variable n'existait pas.
<code>\$_SERVER</code>	Contient les informations liées au serveur Web, tel le contenu des en-têtes HTTP ou le nom du script en cours d'exécution. Retenons les variables suivantes : <code>\$_SERVER["HTTP_ACCEPT_LANGUAGE"]</code> , qui contient le code de langue du navigateur client. <code>\$_SERVER["HTTP_COOKIE"]</code> , qui contient le nom et la valeur des cookies lus sur le poste client. <code>\$_SERVER["HTTP_HOST"]</code> , qui donne le nom de domaine. <code>\$_SERVER["SERVER_ADDR"]</code> , qui indique l'adresse IP du serveur. <code>\$_SERVER["PHP_SELF"]</code> , qui contient le nom du script en cours. Nous l'utiliserons souvent dans les formulaires. <code>\$_SERVER["QUERY_STRING"]</code> , qui contient la chaîne de la requête utilisée pour accéder au script.
<code>\$_SESSION</code>	Contient l'ensemble des noms des variables de session et leurs valeurs.

**Tableau 2 : Fonctions mathématiques**

<code>double/integer abs (double/integer X)</code>	Valeur absolue de X : <code>echo abs(-543); //affiche 543.</code>
<code>double acos (double X)</code>	Arc cosinus de X, qui doit être compris entre - 1 et + 1. Le résultat est en radians : <code>echo acos(0.5); // affiche 1.0471975511966.</code>
<code>double acosh (double X)</code>	Arc cosinus hyperbolique de X. Ne fonctionne pas sous Windows.
<code>double asin (double X)</code>	Arc sinus de X, qui doit être compris entre - 1 et + 1. Le résultat est en radians : <code>echo asin(0.5); // affiche 0.5235987755983.</code>
<code>double asinh (double X)</code>	Arc sinus hyperbolique de X. Ne fonctionne pas sous Windows.
<code>double atan (double X)</code>	Arc tangente de X. Le résultat est en radians : <code>echo atan(5);// affiche 0.46364760900081.</code>
<code>double atan2 (double Y, double X )</code>	Arc tangente du rapport Y/X. Le résultat est en radians. Il faut que Y soit différent de 0.
<code>double atanh (double X)</code>	Arc tangente hyperbolique de X.
<code>string base_convert (string N, integer B1, integer B2)</code>	Convertit le nombre N contenu dans une chaîne de la base B1 dans la base B2.
<code>integer bindec (string X)</code>	Convertit un nombre binaire X contenu dans une chaîne en base 10.
<code>double ceil (double X)</code>	Retourne l'entier immédiatement supérieur à X.
<code>double cos (double X)</code>	Cosinus de X qui doit être exprimé en radians.
<code>double cosh (double X)</code>	Cosinus hyperbolique de X.
<code>string decbin (integer X)</code>	Convertit X de la base 10 en binaire.
<code>string dechex (integer X)</code>	Convertit X de la base 10 en hexadécimal.
<code>string decoct (integer X)</code>	Convertit X de la base 10 en octal.
<code>double deg2rad (double X)</code>	Convertit X de degrés en radians.
<code>double exp (double X)</code>	Exponentielle de X, soit $e^x$ .
<code>double expm1 (double X)</code>	Retourne l'exponentielle de $X - 1$ , soit $e^x - 1$ .
<code>double floor (double X)</code>	Retourne la partie entière de X, soit l'entier immédiatement inférieur à X.
<code>double fmod (double X, double Y)</code>	Retourne le reste de la division de Y par X pour des opérandes de type double.
<code>integer getrandmax (void)</code>	Indique la valeur maximale retournée par la fonction <code>rand()</code> .
<code>integer hexdec (string CH)</code>	Convertit la chaîne hexadécimale CH en décimal.
<code>double hypot (double X, double Y)</code>	Retourne la valeur de l'hypoténuse d'un triangle rectangle dont les côtés de l'angle droit sont X et Y, donc la valeur de la racine carrée de $(X^2 + Y^2)$ .

<code>boolean is_finite ( double X )</code>	Retourne TRUE si la valeur X est finie, c'est-à-dire dans l'intervalle des valeurs admises pour un double, et FALSE dans le cas contraire.
<code>boolean is_infinite ( double X )</code>	Retourne TRUE si la valeur X est supérieure à la valeur maximale admise pour un double, et FALSE dans le cas contraire.
<code>boolean is_nan (double X)</code>	Retourne TRUE si la valeur X n'est pas un nombre, et FALSE dans le cas contraire.
<code>double log_value (void)</code>	Retourne un nombre aléatoire compris entre 0 et 1.
<code>double log (double X, double B)</code>	Logarithme népérien (de base e) du nombre X.
<code>double log10 (double X)</code>	Logarithme décimal (de base 10) de X.
<code>double loglp (double X)</code>	Logarithme népérien de (1 + X).
<code>double/integer max (double/integer X, double/integer Y)</code>	Retourne la valeur maximale de X et de Y.
<code>double/integer min (double/integer X, double/integer Y)</code>	Retourne la valeur minimale de X et de Y.
<code>integer mt_getrandmax (void)</code>	Retourne la plus grande valeur aléatoire que peut retourner la fonction <code>mt_rand()</code> .
<code>integer mt_rand ( integer Min, integer Max)</code>	Génère un résultat compris entre Min et Max ou entre 0 et la constante <code>RAND_MAX</code> si vous omettez les paramètres.
<code>void mt_srand ( integer N)</code>	Initialise le générateur de nombres aléatoires pour la fonction <code>mt_rand()</code> . Le paramètre <i>N</i> est un entier quelconque.
<code>integer octdec (string CH)</code>	Convertit un nombre octal contenu dans la chaîne CH en base 10.
<code>double pi (void)</code>	Retourne la valeur de $\pi$ .
<code>double pi (void)</code>	Retourne la valeur de $\pi$ .
<code>double/integer pow (double/integer X, double/integer Y)</code>	Calcule X à la puissance Y. Les paramètres peuvent être entiers ou décimaux.
<code>double rad2deg (double X)</code>	Convertit X de radians en degrés.
<code>integer rand (integer Min, integer Max)</code>	Retourne un nombre aléatoire compris entre Min et Max y compris les bornes.
<code>double round (double X, integer N)</code>	Arrondit X avec N décimales.
<code>double sin (double X)</code>	Sinus de X exprimé en radians.
<code>double sinh (double X)</code>	Sinus hyperbolique de X.
<code>double sqrt (double X)</code>	Racine carrée de X (qui doit être positif).
<code>void srand (integer N)</code>	Initialise le générateur de nombres aléatoires de la fonction <code>rand()</code> . Le paramètre <i>N</i> est un entier quelconque.
<code>double tan (double X)</code>	Tangente de X qui doit être en radians.
<code>double tanh (double X)</code>	Tangente hyperbolique de X.

**Tableau 3 : Opérateurs de comparaison****Les opérateurs de comparaison**

Opérateur	Description
==	Teste l'égalité de deux valeurs. L'expression <code>\$a == \$b</code> vaut TRUE si la valeur de <code>\$a</code> est égale à celle de <code>\$b</code> et FALSE dans le cas contraire : <code>\$a = 345;</code> <code>\$b = "345";</code> <code>\$c = (\$a==\$b);</code> <code>\$c</code> est un booléen qui vaut TRUE car dans un contexte de comparaison numérique, la chaîne "345" est évaluée comme le nombre 345. Si <code>\$b="345 éléphants"</code> nous obtenons le même résultat.
!= ou <>	Teste l'inégalité de deux valeurs. L'expression <code>\$a != \$b</code> vaut TRUE si la valeur de <code>\$a</code> est différente de celle de <code>\$b</code> et FALSE dans le cas contraire.
===	Teste l'identité des valeurs et des types de deux expressions. L'expression <code>\$a === \$b</code> vaut TRUE si la valeur de <code>\$a</code> est égale à celle de <code>\$b</code> et que <code>\$a</code> et <code>\$b</code> sont du même type. Elle vaut FALSE dans le cas contraire : <code>\$a = 345;</code> <code>\$b = "345";</code> <code>\$c = (\$a=== \$b);</code> <code>\$c</code> est un booléen qui vaut FALSE car si les valeurs sont égales, les types sont différents (integer et string).
!==	Teste la non-identité de deux expressions. L'expression <code>\$a !== \$b</code> vaut TRUE si la valeur de <code>\$a</code> est différente de celle de <code>\$b</code> ou si <code>\$a</code> et <code>\$b</code> sont d'un type différent. Dans le cas contraire, elle vaut FALSE : <code>\$a = 345;</code> <code>\$b = "345";</code> <code>\$c = (\$a!== \$b);</code> <code>\$c</code> est un booléen qui vaut TRUE car si les valeurs sont égales, les types sont différents (integer et string).
<	Teste si le premier opérande est strictement inférieur au second.
<=	Teste si le premier opérande est inférieur ou égal au second.
>	Teste si le premier opérande est strictement supérieur au second.
>=	Teste si le premier opérande est supérieur ou égal au second.

**Tableau 4 : Opérateurs logiques**

**Les opérateurs logiques**

Opérateur	Description
OR	Teste si l'un au moins des opérandes a la valeur TRUE : \$a = true; \$b = false; \$c = false; \$d = (\$a OR \$b); // \$d vaut TRUE. \$e = (\$b OR \$c); // \$e vaut FALSE.
	Équivaut à l'opérateur OR mais n'a pas la même priorité.
XOR	Teste si un et un seul des opérandes a la valeur TRUE : \$a = true; \$b = true; \$c = false; \$d = (\$a XOR \$b); // \$d vaut FALSE. \$e = (\$b XOR \$c); // \$e vaut TRUE.
AND	Teste si les deux opérandes valent TRUE en même temps : \$a = true; \$b = true; \$c = false; \$d = (\$a AND \$b); // \$d vaut TRUE. \$e = (\$b AND \$c); // \$e vaut FALSE.
&&	Équivaut à l'opérateur AND mais n'a pas la même priorité.
!	Opérateur unaire de négation, qui inverse la valeur de l'opérande : \$a = TRUE; \$b = FALSE; \$d = !\$a; // \$d vaut FALSE. \$e = !\$b; // \$e vaut TRUE.

**Tableau 5 : Les caractères de formatage du type de donnée**

Caractère	Signification
%b	Interprète la chaîne \$ch comme un entier et l'affiche en binaire : \$ch='89'; printf ('En binaire \$ch s'écrit %b  ', \$ch) ; //Affiche :En binaire 89 s'écrit 1011001
%c	Interprète la chaîne \$ch comme un entier et affiche le caractère dont le code ASCII correspond à ce nombre : \$ch='115'; printf (' Le caractère de code \$ch est %c  ', \$ch); //Affiche: Le caractère de code 115 est s
%d	Interprète la chaîne \$ch comme un entier signé et l'affiche comme un nombre en base 10 : \$ch = '-25'; printf ('La valeur de \\\\$ch est %d', \$ch); //Affiche -25
%f	Interprète la chaîne \$ch comme un nombre de type double et l'affiche avec sa partie décimale à 6 chiffres. Les caractères non numériques de \$ch ne sont pas pris en compte : \$ch = '25.52 mètres'; printf ('La longueur est de %f m', \$ch); // Affiche: La longueur est de 25.520000 m
%o	Interprète le chaîne \$ch comme un entier et l'affiche en octal : \$ch = 252; printf ('En octal le nombre \$ch s'écrit %o', \$ch); // Affiche: En octal le nombre 252 s'écrit 374
%s	Interprète \$ch comme une chaîne et l'affiche telle quelle : \$ch1 = 'Monsieur ' ; \$ch2 = ' Schtroumpf' ; sprintf ('Bonjour,%s %s brave !', \$ch1 \$ch2); Équivaut à : echo 'Bonjour \$ch1 \$ch2, brave !';
%x ou %X	Interprète la chaîne \$ch comme un entier et l'affiche en hexadécimal en minuscules (%x) ou (%X) : \$ch = '252756'; printf ('En hexadécimal \$ch s'écrit %x ', \$ch) ; // Affiche: En hexadécimal 252756 s'écrit 3db54

**Tableau 6 : Principales fonctions de mise en forme d'une chaîne de caractères**

<code>string strtolower(string \$chaine)</code>	Retourne la chaîne avec tous les caractères en <b>minuscules</b> .
<code>string strtoupper(string \$chaine)</code>	Retourne la chaîne avec tous les caractères en <b>majuscules</b> .
<code>string ucwords(string \$chaine)</code>	Retourne la chaîne avec toutes les initiales des mots qui la composent en majuscules.
<code>string ucfirst(string \$chaine)</code>	Retourne la chaîne avec uniquement la première lettre en majuscule.
<code>string ltrim(string \$chaine [,string liste])</code>	Renvoie la chaîne nettoyée des espaces situés en début de chaîne.
<code>string rtrim(string \$chaine [,string liste])</code>	Renvoie la chaîne nettoyée des espaces situés en fin de chaîne.
<code>string trim(string \$chaine [,string liste])</code>	Supprime les espaces situés en début et en fin de chaîne.
<code>string wordwrap(string \$chaine [, int N [,string car [,boolean coupe]]])</code>	Permet d'afficher un texte long avec une largeur maximale déterminée. Le paramètre N définit cette largeur. Le paramètre car contient la chaîne à insérer dans \$chaine tous les N caractères. Le paramètre booléen coupe permet, s'il vaut TRUE, d'effectuer une césure des mots dont la longueur dépasse N caractères.

**Tableau 7 : Exemples d'utilisation des fonctions de mise en forme d'une chaîne de caractères**

<b>Exemple 1</b>	<pre> &lt;?php \$nom = "ENgels" ; \$prenom = "jEan " ; \$adresse = "21, rue compoinT" ; \$ville = "75018 pÂris" ; \$mail = "ENGELS@funPHP.Com" ; \$prenom = ucfirst (strtolower (\$prenom)) ; \$nom = strtoupper (\$nom) ; \$adresse = ucwords(strtolower (\$adresse)) ; \$ville = strtoupper (\$ville); \$mail = strtolower (\$mail); echo "Mes coordonnées &lt;br /&gt;"; echo \$prenom, \$nom, "&lt;br /&gt;"; echo \$adresse, "&lt;br /&gt;" ; echo \$ville, "&lt;br /&gt;" ; ?&gt; </pre>	<p>Mes coordonnées  Jean ENGELS  21, rue Compoint  75018 PARIS</p>
<b>Exemple 2</b>	<pre> &lt;?php \$a = "...Jean " ; \$b = "Dupont__"; echo \$a,\$b,"&lt;br /&gt;"; echo trim(\$a,' '),trim(\$b,' _'); ?&gt; </pre>	<p>...Jean Dupont__  Jean Dupont</p>
<b>Exemple 3</b>	<pre> echo wordwrap(\$ch,30,"&lt;br /&gt;",1); </pre>	<p>Le contenu de la chaîne \$ch s'affiche dans une colonne de 30 caractères de largeur. Dans le code source HTML, un élément &lt;br/&gt; est trouvé tous les 30 caractères.</p>



**Tableau 8 : Principales fonctions de recherche de sous-chaînes**

string strstr(string \$chaine1, string \$chaine2)	Recherche si la chaîne \$chaine2 est contenue dans la chaîne \$chaine1 et retourne tous les caractères allant de la première occurrence de \$chaine2 jusqu'à la fin de \$chaine1. Si la chaîne \$chaine2 ne figure pas dans la chaîne \$chaine1, la fonction retourne FALSE. Cette recherche est sensible à la casse.
string stristr(string \$chaine1, string \$chaine2)	Même fonctionnement que la fonction strstr sauf que la recherche ici est insensible à la casse.
string strrchr(string \$chaine1, string \$chaine2)	Recherche si la chaîne \$chaine2 est contenue dans la chaîne \$chaine1 et retourne tous les caractères allant de la dernière occurrence de \$chaine2 jusqu'à la fin de \$chaine1. Si la chaîne \$chaine2 ne figure pas dans la chaîne \$chaine1, la fonction retourne FALSE.
string substr(string \$chaine, integer indice [, integer N])	Retourne la chaîne contenant N caractères de \$chaine extraits à partir de l'indice indice inclus. Si le paramètre N n'est pas spécifié, elle retourne la sous-chaîne comprise entre l'indice indice et la fin de \$chaine.
int substr_count(string \$chaine1, string \$chaine2)	Retourne le nombre d'occurrence d'une sous-chaîne \$chaine2 dans une chaîne \$chaine1.
string str_replace(string \$chaine1, string \$chaine2, string \$chaine3 [, string \$var])	Retourne la chaîne \$chaine3 dont toutes les occurrences de \$chaine1 ont été remplacées par la chaîne \$chaine2. Le paramètre \$var correspond au nom d'une variable à laquelle est passé par référence le nombre de remplacement effectué.
string strpos(string \$chaine1, string \$chaine2)	Retourne la position du premier caractère de la première occurrence d'une chaîne \$chaine2 dans une chaîne \$chaine1 ou FALSE si \$chaine2 ne figure pas dans \$chaine1. Cette fonction est sensible à la casse.
string stripos(string \$chaine1, string \$chaine2)	Même fonctionnement que la fonction strpos sauf que cette fonction est insensible à la casse.
string strrpos(string \$chaine1, string \$chaine2)	Retourne la position du premier caractère de la dernière occurrence d'une

	chaîne \$chaine2 dans une chaîne \$chaine1 ou FALSE si \$chaine2 ne figure pas dans \$chaine1. Cette fonction est sensible à la casse.
string stripos(string \$chaine1, string \$chaine2)	Même fonctionnement que la fonction strrpos sauf que cette fonction est insensible à la casse.
string sscanf(string \$chaine, string "format" [, \$var1, \$var2...])	Extrait des sous-chaînes et affecte leurs valeurs à des variables.

**Tableau 9 : Exemples d'utilisation des fonctions de recherche de sous-chaînes**

Exemple 1
<pre>\$ch = "Perette et le pot au lait" \$ssch = strstr (\$ch, "pot") echo \$ssch ;</pre>
Affiche la sous-chaîne "pot au lait"
Exemple 2
<pre>\$ch = "Perette et le pot au lait. C'est pas de pot !" ; \$ssch = strrchr(\$ch, "pot") echo \$ssch ;</pre>
Affiche la sous-chaîne "pot !"
Exemple 3
<pre>&lt;?php \$ch = "Perette et le pot au lait. C'est pas de pot!" ; \$ssch = substr (\$ch, 8, 9) ; echo \$ssch,"&lt;br /&gt;" ;  \$ssch = substr(\$ch,8); echo \$ssch ,"&lt;br /&gt;"; \$ch2="pot"; \$nb=substr_count(\$ch,\$ch2); echo "Le mot \$ch2 est présent \$nb fois dans \$ch &lt;br /&gt;"; \$ch3=str_replace('pot','bol',\$ch); echo \$ch3,"&lt;br /&gt;" ; ?&gt;</pre>
<p>et le pot  et le pot au lait. C'est pas de pot!  Le mot pot est présent 2 fois dans Perette et le pot au lait. C'est pas de pot!  Perette et le bol au lait. C'est pas de bol!  2 remplacements</p>
Exemple 4
<pre>&lt;?php \$ch = "Perette et le pot au lait. C'est pas de pot ! La Fontaine" ; echo "\\$ch = \$ch &lt;br /&gt;"; \$ch2 = "pot" ; //recherche sensible à la casse \$n=strpos (\$ch, \$ch2); echo "Le mot \$ch2 commence à la position \$n dans \\$ch &lt;br /&gt;" ; //recherche insensible à la casse \$ch3 = "POT" ; \$n2=stripos(\$ch, \$ch3); echo "Le mot \$ch3 commence à la position \$n2 dans \\$ch &lt;br /&gt;" ; //recherche de la dernière occurrence sensible à la casse \$n3=strrpos(\$ch, \$ch2); echo "La dernière occurrence du mot \$ch2 commence à la position \$n3 dans \\$ch&lt;br /&gt;"  //recherche sensible à la casse de l'existence d'un mot \$ch4="fontaine";</pre>

```
if(!strpos ($ch,$ch4))  
{ echo "Le mot $ch4 n'est pas dans \"$ch\";}  
?>
```

```
$ch = Perette et le pot au lait. C'est pas de pot ! La Fontaine  
Le mot pot commence à la position 14 dans $ch  
Le mot POT commence à la position 14 dans $ch  
La dernière occurrence du mot pot commence à la position 40 dans $ch  
Le mot fontaine n'est pas dans $ch
```

### Exemple 5

```
<?php  
$personne = "1685-1750 Jean-Sébastien Bach";  
$format="%d-%d %s %s";  
$nb = sscanf($personne,$format,$ne,$mort,$prenom,$nom);  
echo "$prenom $nom né en $ne, mort en $mort <br />";  
echo "Nous lisons $nb informations";  
?>
```

```
Jean-Sébastien Bach né en 1685, mort en 1750  
Nous lisons 4 informations
```

**Tableau 10 : Opérateurs de comparaison de chaînes de caractères**

Opérateurs de comparaison	exemples
<p>Opérateur d'égalité (==) :</p> <ul style="list-style-type: none"> <li>- Si les deux opérandes sont des chaînes de caractères, elles doivent avoir exactement les mêmes caractères pour que l'égalité soit vérifiée.</li> <li>- Si la comparaison se fait entre une chaîne commençant par des chiffres et un nombre, elle a lieu comme s'il s'agissait de deux nombres, seuls les caractères numériques du début de la chaîne étant pris en compte.</li> </ul>	<pre>\$ch1="PHP 5"; \$ch2="PHP 4"; if(\$ch2==\$ch1) echo "TRUE &lt;br /&gt;"; else echo "FALSE &lt;br /&gt;"; // affiche FALSE</pre> <pre>\$nb=59; \$ch="59scripts"; if(\$ch==\$nb) echo "TRUE &lt;br /&gt;"; else echo "FALSE &lt;br /&gt;"; // affiche TRUE</pre>
<p>Opérateur d'identité (===) :</p> <ul style="list-style-type: none"> <li>- La valeur TRUE n'est obtenue que si les deux opérandes ont la même valeur et le même type.</li> </ul>	<pre>\$nb=59; \$ch="59scripts"; if(\$ch==\$nb) echo "TRUE &lt;br /&gt;"; else echo "FALSE &lt;br /&gt;"; // affiche FALSE</pre>
<p>Opérateurs &lt;, &gt;, &lt;= et &gt;= :</p> <ul style="list-style-type: none"> <li>- Si les opérandes sont des chaînes, la comparaison est effectuée en fonction de l'ordre ASCII des caractères des chaînes.</li> </ul>	<pre>\$ch1="Blanc"; \$ch2="Bleu"; \$ch3="blanc"; if(\$ch1&gt;\$ch2) echo "TRUE &lt;br /&gt;"; else echo "FALSE &lt;br /&gt;"; //Affiche FALSE if(\$ch1&lt;\$ch3) echo "TRUE &lt;br /&gt;"; else echo "FALSE &lt;br /&gt;"; //Affiche TRUE</pre>

**Tableau 11 : Fonctions de comparaison de chaînes de caractères**

Fonctions	Exemples
int strcmp (string \$ch1, string \$ch2) - Permet de comparer des chaînes et obtenir leur ordre alphabétique, au sens ASCII du terme. Retourne -1 si \$ch1 est inférieure à \$ch2, 1 dans le cas contraire et 0 en cas d'égalité. Cette fonction est <b>sensible</b> à la casse.	<pre> \$ch1="Blanc"; \$ch2="Bleu"; \$ch3="blanc"; echo strcmp (\$ch1,\$ch2); //Affiche -1 echo strcasecmp (\$ch1,\$ch3); //Affiche 0 echo strncasecmp ( \$ch1, \$ch2,2); //Affiche 0           </pre>
int strcasecmp (string \$ch1, string \$ch2) - Même fonctionnement que strcmp sauf que cette fonction est <b>insensible</b> à la casse.	
int strncmp(string \$ch1, string \$ch2, int N) - Réalise la même action que la fonction strcmp mais en limitant la comparaison aux N premiers caractères.	
int strncasecmp ( string \$ch1, string \$ch2, int N) - Réalise la même action que la fonction strcasecmp mais en limitant la comparaison aux N premiers caractères.	
int strnatcmp (string \$ch1, string \$ch2) - Effectue une comparaison entre deux chaînes dans l'ordre « naturel ». Cette fonction est sensible à la casse.	<pre> \$ch4="page2"; \$ch5="page12"; echo strcmp ( \$ch4,\$ch5); //Affiche 1 echo strnatcmp(\$ch4,\$ch5); //Affiche -1           </pre>
int strnatcasecmp (string \$ch1, string \$ch2) - Même fonctionnement que la fonction strnatcmp sauf que cette fonction est insensible à la casse.	
int similar_text ( string \$ch1, string \$ch2 [, \$pourcent]) - Retourne le nombre de caractères communs à deux chaînes. Lorsque le troisième paramètre est utilisé, le pourcentage de similarité est retourné dans la variable \$pourcent. Cette fonction est sensible à la casse.	<pre> \$ch4="MySQL"; \$ch5="PgSQL"; echo similar_text(\$ch4,\$ch5,\$pourc), " caractères communs"; echo "Similarité : ", \$pourc, "%";  // affiche : 3 caractères communs // Similarité : 60%           </pre>

**Tableau 12: Fonctions permettant la transformation de chaînes en tableaux**

Fonctions	Exemples
<p>array <b>explode</b> ( string <b>sep</b>, string \$ch [, int N])</p> <ul style="list-style-type: none"> <li>- Retourne un tableau des mots de \$ch, le critère de séparation étant donné par la chaîne sep (souvent une espace). Si le dernier paramètre est fourni, le tableau ne contient que N éléments au maximum, le dernier élément contenant toute la fin de la chaîne initiale.</li> </ul>	<pre>&lt;?php //Passage chaîne -&gt; tableau \$ch1="L'avenir est à PHP5 et MySQL"; \$tab1=explode(" ",\$ch1); echo \$ch1,"&lt;br /&gt;"; print_r(\$tab1); echo "&lt;hr /&gt;"; \$ch2="C:\wampserver\www\php5\chaines\string2.php"; \$tab2=explode("\\",\$ch2); echo \$ch2,"&lt;br /&gt;"; print_r(\$tab2); echo "&lt;hr /&gt;"; //Passage tableau -&gt; chaîne \$tab3[0]="Bonjour"; \$tab3[1]="monsieur"; \$tab3[2]="Rasmus"; \$tab3[3]="Merci!"; \$ch3=implode(" ",\$tab3); echo \$ch3,"&lt;br /&gt;"; ?&gt;</pre>
<p>string <b>implode</b> ( string sep, array \$tab)</p> <ul style="list-style-type: none"> <li>- Il s'agit de la fonction réciproque de explode(). Elle retourne une chaîne composée des éléments d'un tableau séparés par un caractère donné.</li> </ul>	<pre>// résultat affiché : /* L'avenir est à PHP5 et MySQL Array ( [0] =&gt; L'avenir [1] =&gt; est [2] =&gt; à [3] =&gt; PHP5 [4] =&gt; et [5] =&gt; MySQL ) C:\wampserver\www\php5\chaines\string2.php Array ( [0] =&gt; C: [1] =&gt; wampserver [2] =&gt; www [3] =&gt; php5 [4] =&gt; chaines [5] =&gt; string2.php ) Bonjour monsieur Rasmus Merci! */</pre>

**Tableau 13: Exemple de Lecture et affichage d'un tableau avec la fonction `each()`**

Script PHP :
<pre>&lt;?php //*****Lecture d'un tableau indicé***** \$montab=array("Paris","London","Brüssel");//indices 0,1,2 //Ajout d'un élément au tableau \$montab[9]="Berlin"; //Lecture des éléments reset(\$montab); //Pour que le pointeur interne du tableau soit positionné au début du tableau while(\$selement=each(\$montab)) {     echo "L'élément d'indice \$selement[0] a la valeur \$selement[1]&lt;br /&gt;"; } echo "&lt;br/&gt;"; //*****Lecture d'un tableau associatif***** \$montab=array("France"=&gt;"Paris","Great Britain"=&gt;"London","België"=&gt;"Brüssel"); //Ajout d'un élément au tableau \$montab["Deutschland"]="Berlin"; //Lecture des éléments reset(\$montab); while(\$selement=each(\$montab)) {     echo "L'élément de clé {\$selement['key']} a la valeur {\$selement['value']}&lt;br /&gt;"; } ?&gt;</pre>
Résultat affiché :
<p>L'élément d'indice 0 a la valeur Paris L'élément d'indice 1 a la valeur London L'élément d'indice 2 a la valeur Brüssel L'élément d'indice 9 a la valeur Berlin</p> <p>L'élément de clé France a la valeur Paris L'élément de clé Great Britain a la valeur London L'élément de clé België a la valeur Brüssel L'élément de clé Deutschland a la valeur Berlin</p>



**Tableau 14: Exemple de lecture et affichage d'un tableau avec l'instruction foreach :**

Script PHP
<pre>&lt;?php //***** //Lecture de tableau indicé sans récupération des indices //***** \$tab=array("Paris","London","Brüssel"); echo "&lt;H3&gt;Lecture des valeurs des éléments &lt;/H3&gt;"; foreach(\$tab as \$ville) { echo "&lt;b&gt;\$ville&lt;/b&gt; &lt;br&gt;"; } echo"&lt;hr&gt;"; //***** //Lecture de tableau indicé avec récupération des indices //***** echo "&lt;h3&gt;lecture des indices et des valeurs des éléments &lt;/h3&gt;"; foreach(\$tab as \$indice=&gt;\$ville) { echo "L'élément d'indice &lt;b&gt;\$indice&lt;/b&gt; a la valeur &lt;b&gt;\$ville&lt;/b&gt;&lt;br&gt;"; } echo"&lt;hr&gt;"; //***** //Lecture de tableau associatif avec récupération des clés //***** \$tab2=array("France"=&gt;"Paris","Great Britain"=&gt;"London","België"=&gt;"Brüssel"); echo "&lt;h3&gt;lecture des clés et des valeurs des éléments&lt;/h3&gt;"; foreach(\$tab2 as \$cle=&gt;\$ville) { echo "L'élément de clé &lt;b&gt;\$cle&lt;/b&gt; a la valeur &lt;b&gt;\$ville&lt;/b&gt; &lt;br&gt;"; } echo"&lt;hr&gt;"; ?&gt;</pre>

**Tableau 15: Exemple d'ajout et de suppression d'éléments dans un tableau :**

Script PHP:
<pre>&lt;?php \$tab= array(800,1492, 1515, 1789); print_r(\$tab); //Ajout au début du tableau \$poitiers=732; \$nb=array_unshift(\$tab,500,\$poitiers); echo "Le tableau \$tab a maintenant \$nb éléments &lt;br&gt;"; print_r(\$tab); //Ajout à la fin du tableau \$armit=1918; \$newnb=array_push(\$tab,1870,1914,\$armit); echo "Le tableau \$tab a maintenant \$newnb éléments &lt;br&gt;"; print_r(\$tab); //Suppression du dernier élément \$suppr= array_pop(\$tab); echo "Le tableau \$tab a perdu l'élément \$suppr &lt;br&gt;"; print_r(\$tab); //Suppression du premier élément \$suppr= array_shift(\$tab); echo "Le tableau \$tab a perdu l'élément \$suppr &lt;br&gt;"; print_r(\$tab); //Suppression de l'élément d'indice 4 unset(\$tab[4]); print_r(\$tab); ?&gt;</pre>
Résultat affiché :
<pre>Array ( [0] =&gt; 800 [1] =&gt; 1492 [2] =&gt; 1515 [3] =&gt; 1789 ) Le tableau \$tab a maintenant 6 éléments Array ( [0] =&gt; 500 [1] =&gt; 732 [2] =&gt; 800 [3] =&gt; 1492 [4] =&gt; 1515 [5] =&gt; 1789 ) Le tableau \$tab a maintenant 9 éléments Array ( [0] =&gt; 500 [1] =&gt; 732 [2] =&gt; 800 [3] =&gt; 1492 [4] =&gt; 1515 [5] =&gt; 1789 [6] =&gt; 1870 [7] =&gt; 1914 [8] =&gt; 1918 ) Le tableau \$tab a perdu l'élément 1918 Array ( [0] =&gt; 500 [1] =&gt; 732 [2] =&gt; 800 [3] =&gt; 1492 [4] =&gt; 1515 [5] =&gt; 1789 [6] =&gt; 1870 [7] =&gt; 1914 ) Le tableau \$tab a perdu l'élément 500 Array ( [0] =&gt; 732 [1] =&gt; 800 [2] =&gt; 1492 [3] =&gt; 1515 [4] =&gt; 1789 [5] =&gt; 1870 [6] =&gt; 1914 ) L'élément d'indice 4 a été supprimé Array ( [0] =&gt; 732 [1] =&gt; 800 [2] =&gt; 1492 [3] =&gt; 1515 [5] =&gt; 1870 [6] =&gt; 1914 )</pre>

**Tableau 16: Exemple d'intersection et différence de deux tableaux:**

Script PHP:
<pre>&lt;?php \$tab1=array("Blanc","Jaune","Rouge","Vert","Bleu","Noir"); \$tab2=array("Bleu","Rouge","Violet","Noir","Jaune","Orange"); echo"Le tableau 1 contient les éléments:&lt;br /&gt;"; print_r(\$tab1); echo "&lt;hr /&gt;"; echo"Le tableau 2 contient les éléments:&lt;br /&gt;"; print_r(\$tab2); echo "&lt;hr /&gt;"; echo "Intersection de \\$tab1 et \\$tab2 : "; \$tab3=array_intersect(\$tab1,\$tab2); print_r(\$tab3); echo"&lt;br /&gt;"; echo "Intersection de \\$tab2 et \\$tab1 : "; \$tab4= array_intersect(\$tab2,\$tab1); print_r(\$tab4); echo"&lt;hr /&gt;"; \$tab5= array_diff(\$tab1,\$tab2); echo "Différence de \\$tab1 et \\$tab2 : "; print_r(\$tab5); echo"&lt;br /&gt;"; \$tab6= array_diff(\$tab2,\$tab1); echo "Différence de \\$tab2 et \\$tab1 : "; print_r(\$tab6); echo"&lt;br /&gt;"; ?&gt;</pre>
Résultat affiché :
<p>Le tableau 1 contient les éléments: Array ( [0] =&gt; Blanc [1] =&gt; Jaune [2] =&gt; Rouge [3] =&gt; Vert [4] =&gt; Bleu [5] =&gt; Noir )</p> <p>Le tableau 2 contient les éléments: Array ( [0] =&gt; Bleu [1] =&gt; Rouge [2] =&gt; Violet [3] =&gt; Noir [4] =&gt; Jaune [5] =&gt; Orange )</p> <p>Intersection de \$tab1 et \$tab2 : Array ( [1] =&gt; Jaune [2] =&gt; Rouge [4] =&gt; Bleu [5] =&gt;Noir )</p> <p>Intersection de \$tab2 et \$tab1 : Array ( [0] =&gt; Bleu [1] =&gt; Rouge [3] =&gt; Noir [4] =&gt;Jaune )</p> <p>Différence de \$tab1 et \$tab2 : Array ( [0] =&gt; Blanc [3] =&gt; Vert )</p> <p>Différence de \$tab2 et \$tab1 : Array ( [2] =&gt; Violet [5] =&gt; Orange )</p>

**Tableau 17: Exemple de tri d'un tableau indicé selon l'ordre ASCII :**

Script PHP :
<pre>&lt;?php //***** //TABLEAU INDICE //***** //Définition du tableau \$tabind=array("Blanc2","Jaune","rouge","Vert","Bleu","Noir","Blanc10"); \$copie= \$tabind; echo "&lt;b&gt;Tableau indicé d'origine&lt;/b&gt;&lt;br /&gt;"; print_r(\$tabind); //Fonction sort() echo "&lt;hr /&gt;Tri en ordre ASCII sans sauvegarde des indices&lt;br /&gt;"; \$tabind=\$copie; sort(\$tabind); print_r(\$tabind); //Fonction rsort() echo "&lt;hr /&gt; Tri en ordre ASCII inverse sans sauvegarde des indices&lt;br /&gt;"; \$tabind=\$copie; rsort(\$tabind); print_r(\$tabind); //Fonction array_reverse() echo "&lt;hr /&gt;Inversion de l'ordre des éléments&lt;br /&gt;"; \$tabind=\$copie; \$tabrev=array_reverse(\$tabind); print_r(\$tabrev); ?&gt;</pre>
Résultat affiché :
<p>Tableau indicé d'origine Array ( [0] =&gt; Blanc2 [1] =&gt; Jaune [2] =&gt; rouge [3] =&gt; Vert [4] =&gt; Bleu [5] =&gt; Noir [6] =&gt; Blanc10 )</p> <p>Tri en ordre ASCII sans sauvegarde des indices Array ( [0] =&gt; Blanc10 [1] =&gt; Blanc2 [2] =&gt; Bleu [3] =&gt; Jaune [4] =&gt; Noir [5] =&gt; Vert [6] =&gt; rouge )</p> <p>Tri en ordre ASCII inverse sans sauvegarde des indices Array ( [0] =&gt; rouge [1] =&gt; Vert [2] =&gt; Noir [3] =&gt; Jaune [4] =&gt; Bleu [5] =&gt; Blanc2 [6] =&gt; Blanc10 )</p> <p>Inversion de l'ordre des éléments Array ( [0] =&gt; Blanc10 [1] =&gt; Noir [2] =&gt; Bleu [3] =&gt; Vert [4] =&gt; rouge [5] =&gt; Jaune [6] =&gt; Blanc2 )</p>

**Tableau 18: Exemple de tri d'un tableau indicé selon l'ordre naturel :**

Script PHP :
<pre>&lt;?php //***** //TABLEAU INDICE //***** //Définition du tableau \$tabind=array("Blanc2","Jaune","rouge","Vert","Bleu","Noir","Blanc10","1ZZ","2AA"); \$copie= \$tabind; echo "&lt;b&gt;Tableau indicé d'origine&lt;/b&gt;&lt;br /&gt;"; print_r(\$tabind); //***** echo "&lt;hr /&gt;Tri en ordre naturel avec sauvegarde des indices&lt;br /&gt;"; \$tabind=\$copie; natsort(\$tabind); print_r(\$tabind); //***** echo "&lt;hr /&gt;Tri en ordre naturel insensible à la casse avec sauvegarde des indices&lt;br /&gt;"; \$tabind=\$copie; natcasesort(\$tabind); print_r(\$tabind); foreach (\$tabind as \$cle=&gt;\$val) { echo "\$cle =&gt; \$val &lt;br /&gt;"; } ?&gt;</pre>
Résultat affiché :
<p>Tableau indicé d'origine Array ( [0] =&gt; Blanc2 [1] =&gt; Jaune [2] =&gt; rouge [3] =&gt; Vert [4] =&gt; Bleu [5] =&gt; Noir [6] =&gt; Blanc10 [7] =&gt; 1ZZ [8] =&gt; 2AA ) Tri en ordre naturel avec sauvegarde des indices Array ( [7] =&gt; 1ZZ [8] =&gt; 2AA [0] =&gt; Blanc2 [6] =&gt; Blanc10 [4] =&gt; Bleu [1] =&gt; Jaune [5] =&gt; Noir [3] =&gt; Vert [2] =&gt; rouge ) Tri en ordre naturel insensible à la casse avec sauvegarde des indices Array ( [7] =&gt; 1ZZ [8] =&gt; 2AA [0] =&gt; Blanc2 [6] =&gt; Blanc10 [4] =&gt; Bleu [1] =&gt; Jaune [5] =&gt; Noir [2] =&gt; rouge [3] =&gt; Vert )</p>

**Tableau 19: Exemple de tri d'un tableau associatif :**

Script PHP :
<pre>&lt;?php //***** //TABLEAU ASSOCIATIF \$tabass= array("white2"=&gt;"Blanc2","yellow"=&gt;"Jaune","red"=&gt;"rouge""green"=&gt;"Vert", "blue"=&gt;"Bleu","black"=&gt;"Noir","white10"=&gt;"Blanc10"); \$copieass=\$tabass; echo "&lt;h4&gt;Tableau associatif d'origine&lt;/h4&gt;"; print_r(\$tabass); echo "&lt;h4&gt;Tri en ordre ASCII des clés&lt;/h4&gt;"; \$tabass=\$copieass; ksort(\$tabass); print_r(\$tabass); echo "&lt;h4&gt;Tri en ordre ASCII inverse des clés&lt;/h4&gt;"; \$tabass=\$copieass; krsort(\$tabass); print_r(\$tabass); ?&gt;</pre>
Résultat affiché :
<p>Tableau associatif d'origine Array ( [white2] =&gt; Blanc2 [yellow] =&gt; Jaune [red] =&gt; rouge [green] =&gt; Vert [ blue] =&gt; Bleu [black] =&gt; Noir [white10] =&gt; Blanc10 ) Tri en ordre ASCII des clés Array ( [ blue] =&gt; Bleu [black] =&gt; Noir [green] =&gt; Vert [red] =&gt; rouge [white10] =&gt; Blanc10 [white2] =&gt; Blanc2 [yellow] =&gt; Jaune ) Tri en ordre ASCII inverse des clés Array ( [yellow] =&gt; Jaune [white2] =&gt; Blanc2 [white10] =&gt; Blanc10 [red] =&gt; rouge [green] =&gt; Vert [black] =&gt; Noir [ blue] =&gt; Bleu )</p>

**Tableau 20: Exemple de sélection d'éléments dans un tableau :**

Script PHP :
<pre>&lt;?php //Définition du tableau \$villes=array("Paris","Perpignan","Marseille","pau","Nantes","Lille"); //Fonction de sélection function init(\$ville) { if(\$ville[o]=="P"    \$ville[o]=="p" ) { return \$ville; } } //Utilisation de array_filter() \$select=array_filter(\$villes,"init"); print_r(\$select); ?&gt;</pre>
Résultat affiché :
<pre>Array ( [0] =&gt; Paris [1] =&gt; Perpignan [3] =&gt; pau )</pre>

## Tableau 21: Exemple d'application d'une fonction aux éléments d'un tableau:

### Script PHP :

```
<?php
$tabx= range(1,10);
function tabval($val,$ind)
{
echo "<tr><td><b>", $ind+1,"</b></td><td>".cos(m_pi/$val)."</td></tr>";
}
echo"<table border=\"3\"><tr>
<td> <h2>tableau de valeurs</h2>
<table border=\"1\" >
<thead><th> x </th><th> cos(pi/x)</th></thead>";
array_walk($tabx,"tabval");
echo"</table></td> ";
?>
```

### Résultat affiché :

Application d'une fonction aux éléments d'un tableau - Mozilla Firefox

Fichier Édition Affichage Historique Marque-pages Outils ?

http://www.funhtml.com/php5/C5tableaux/tableau25.html

google earth


### Tableau de valeurs d'une fonction

x	cos(pi/x)
1	-1
2	6.1230317691119E-17
3	0.5
4	0.70710678118655
5	0.80901699437495
6	0.86602540378444
7	0.90096886790242
8	0.92387953251129
9	0.93969262078591
10	0.95105651629515

Terminé



**Tableau 22-1 : Exemple 1 de récupération des données d'un formulaire:**

Script PHP :
<pre>&lt;html&gt; &lt;head&gt; &lt;meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" /&gt; &lt;title&gt;Formulaire traité par PHP&lt;/title&gt; &lt;/head&gt; &lt;body&gt; &lt;form action="&lt;?=\$_SERVER["PHP_SELF"]?&gt;" method="post" enctype="application/x-www-form-urlencoded"&gt; &lt;fieldset&gt; &lt;legend&gt;&lt;b&gt;Infos&lt;/b&gt;&lt;/legend&gt; &lt;div&gt; Nom : &lt;input type="text" name="nom" size="40" /&gt; &lt;br /&gt; Débutant : &lt;input type="radio" name="niveau" value="débutant" /&gt; Initié : &lt;input type="radio" name="niveau" value="initié" /&gt;&lt;br /&gt; &lt;input type="reset" value="Effacer" /&gt; &lt;input type="submit" value="Envoyer" /&gt; &lt;/div&gt; &lt;/fieldset&gt; &lt;/form&gt; &lt;?php if(isset(\$_POST["nom"]) &amp;&amp; isset(\$_POST["niveau"])) { echo "&lt;h2&gt; Bonjour ". stripslashes(\$_POST["nom"]). " vous êtes ".\$_POST["niveau"]." en PHP&lt;/h2&gt;"; } ?&gt; &lt;/body&gt; &lt;/html&gt;</pre>
Résultat affiché :


**N.B :**

- `$_SERVER["PHP_SELF"]` : désigne le script lui-même comme cible pour le traitement de données transmises.
- `application/x-www-form-urlencoded` : la valeur par défaut associée à l'attribut **enctype** qui détermine le type d'encodage des données transmises au serveur. Cette valeur est utilisée dans la plupart des cas, à l'exception du :
  - cas du transfert de fichiers du poste client vers le serveur : elle doit être égale à « `multipart/form-data` ».
  - cas où l'attribut action dans l'élément `<form>` a la valeur « `mailto` » : elle prend les valeurs « `text/plain` » ou « `text/html` » selon que le contenu est envoyé à une adresse e-mail au format texte ou HTML.
- La fonction `stripslashes()` est utilisée pour supprimer les caractères d'échappement ajoutés automatiquement devant les caractères spéciaux éventuellement utilisés dans les données saisies avant de les afficher dans la page.
- Lorsque le script contenant le formulaire est chargé du traitement des données, l'ensemble de la page est réaffiché après traitement, de même que l'ensemble du formulaire qui se retrouve ainsi dans son état initial.

**Tableau 22-2 : Exemple 2 de récupération des données d'un formulaire (Maintien de l'état du formulaire) :**

Script PHP :
<pre> &lt;html&gt; &lt;head&gt; &lt;meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" /&gt; &lt;title&gt;Formulaire traité par PHP&lt;/title&gt; &lt;/head&gt; &lt;body&gt; &lt;form action="&lt;?=\$_SERVER["PHP_SELF"]?&gt;" method="post" enctype="application/x-www- form-urlencoded"&gt; &lt;fieldset&gt; &lt;legend&gt;&lt;b&gt;Infos&lt;/b&gt;&lt;/legend&gt; Nom : &lt;input type="text" name="nom" size="40" value="&lt;?php if(isset(\$_POST["nom"])) echo \$_POST["nom"]?&gt;" /&gt; &lt;br /&gt; Débutant : &lt;input type="radio" name="niveau" value="débutant" &lt;?php if(isset(\$_POST["niveau"]) &amp;&amp; \$_POST["niveau"]=="débutant") echo "checked=\\"checked\\"?&gt; /&gt; Initié : &lt;input type="radio" name="niveau" value="initié" &lt;?php if(isset(\$_POST["niveau"]) &amp;&amp; \$_POST["niveau"]=="initié") echo "checked=\\"checked\\"?&gt; /&gt;&lt;br /&gt; &lt;input type="reset" value="Effacer" /&gt; &lt;input type="submit" value="Envoyer" /&gt; &lt;/fieldset&gt; &lt;/form&gt; &lt;?php if(isset(\$_POST["nom"]) &amp;&amp; isset(\$_POST["niveau"])) { echo "&lt;h2&gt; Bonjour ". stripslashes(\$_POST["nom"]). " vous êtes ".\$_POST["niveau"]. " en PHP&lt;/h2&gt;"; } ?&gt; &lt;/body&gt; &lt;/html&gt; </pre>
Résultat affiché :


**Tableau 22-3 : Exemple 3 de récupération des données d'un formulaire:**

Fichier <b>form5.html</b>
<pre> &lt;html&gt; &lt;head&gt; &lt;meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" /&gt; &lt;title&gt;Listes à choix multiples&lt;/title&gt; &lt;/head&gt; &lt;body&gt; &lt;form method="post" action="form5.php" &gt; &lt;fieldset&gt; &lt;legend&gt;Recherche d'emploi: complétez la fiche &lt;/legend&gt; &lt;div&gt; &lt;span&gt;Nom&lt;input type="text" name="ident[]" /&gt; Prénom&lt;input type="text" name="ident[]" /&gt; Age&lt;input type="text" name="ident[]" /&gt; &lt;br /&gt;&lt;br /&gt; Langues pratiquées&lt;br /&gt; &lt;select name="lang[]" multiple="multiple"&gt; &lt;option value="français"&gt; français&lt;/option&gt; &lt;option value="anglais"&gt; anglais&lt;/option&gt; &lt;option value="allemand"&gt; allemand&lt;/option&gt; &lt;option value="espagnol"&gt; espagnol&lt;/option&gt; &lt;/select&gt;&lt;br /&gt;&lt;br /&gt; Compétences informatiques&lt;br /&gt; XHTML&lt;input type="checkbox" name="competent[]" value="XHTML" /&gt; PHP&lt;input type="checkbox" name="competent[]" value="PHP" /&gt; MySQL&lt;input type="checkbox" name="competent[]" value="MySQL" /&gt; ASP.Net&lt;input type="checkbox" name="competent[]" value="ASP.Net" /&gt; &lt;/span&gt;&lt;br /&gt;&lt;br /&gt; &lt;input type="reset" value="EFFACER"/&gt; &lt;input type="submit" value="ENVOI"/&gt; &lt;/div&gt; &lt;/fieldset&gt; &lt;/form&gt; &lt;/body&gt; &lt;/html&gt; </pre>
Fichier <b>form5.php</b>
<pre> &lt;html&gt; &lt;head&gt; &lt;meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" /&gt; &lt;title&gt;Compétences Info&lt;/title&gt; &lt;/head&gt; &lt;body&gt; &lt;?php if(isset(\$_POST["ident"]) &amp;&amp; isset(\$_POST["lang"]) &amp;&amp; isset(\$_POST["competent"])) { </pre>

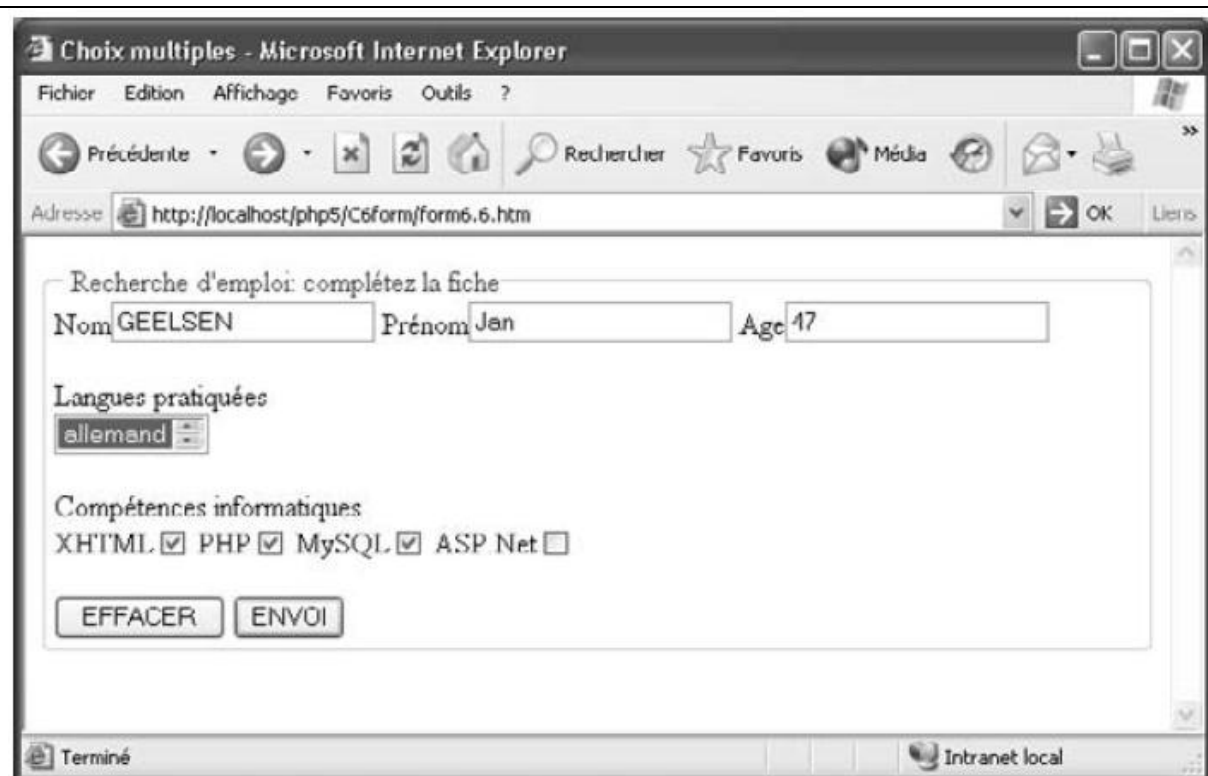
```

echo "<table border='1'><tr><th> Récapitulatif de votre fiche d'information
personnelle</th></tr><tr><td>";
$nom=$_POST["ident"][0];
$prenom=$_POST["ident"][1];
$age=$_POST["ident"][2];
$lang = $_POST["lang"];
$competent=$_POST["competent"];
echo"Vous êtes :<b> $prenom ", stripslashes($nom) ,"</b><br />Vous avez <b>$age ans
</b> ";
echo "<br />Vous parlez :";
echo "<ul>";
foreach($lang as $valeur)
{
echo " <li> $valeur </li>";
}
echo "</ul>";
echo "Vous avez des compétences informatiques en :";
echo "<ul>";
foreach($competent as $valeur)
{
echo "<li> $valeur </li> ";
}
echo "</ul> </td></tr>";
}
else
{
echo"<script type='text/javascript'>" ;
echo "alert('Cochez au moins une compétence!!');";
echo "window.history.back();";
echo "</script>";
}
?>
</body>
</html>

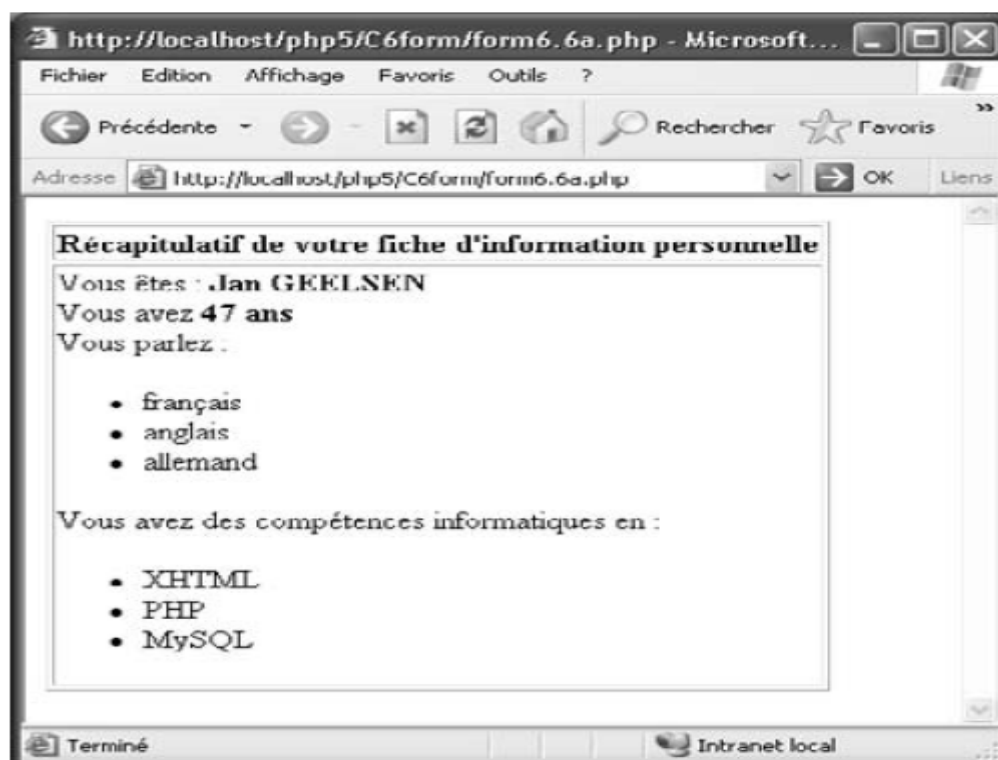
```

Résultat affiché :

Avant de cliquer sur le bouton Envoi :



Après avoir cliqué sur le bouton Envoi :



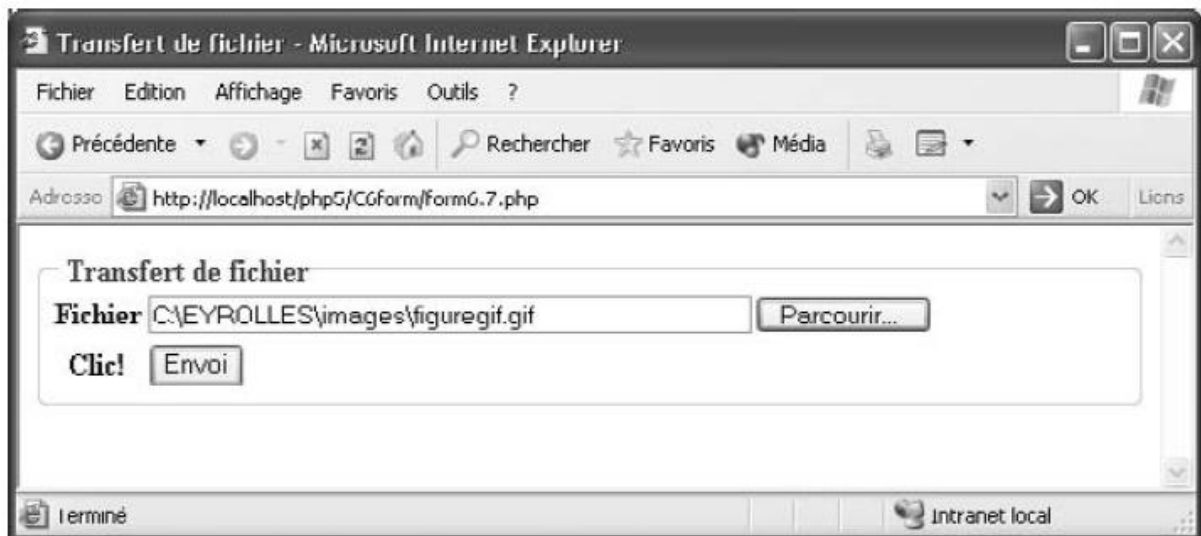
**Tableau 23 : Exemple de transfert de fichiers vers le serveur via un formulaire:**

Fichier form6.php
<pre>&lt;html&gt; &lt;head&gt; &lt;meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" /&gt; &lt;title&gt;Transfert de fichier&lt;/title&gt; &lt;/head&gt; &lt;body&gt; &lt;form action="form6.php" method="post" enctype="multipart/form-data"&gt; &lt;fieldset&gt; &lt;input type="hidden" name="MAX_FILE_SIZE" value="100000" /&gt; &lt;legend&gt;&lt;b&gt;Transfert de fichier&lt;/b&gt;&lt;/legend&gt; &lt;table&gt; &lt;tbody&gt; &lt;tr&gt; &lt;th&gt;Fichier&lt;/th&gt; &lt;td&gt; &lt;input type="file" name="fich" accept="image/gif" size="50"/&gt;&lt;/td&gt; &lt;/tr&gt; &lt;tr&gt; &lt;th&gt;Clic!&lt;/th&gt; &lt;td&gt; &lt;input type="submit" value="Envoi" /&gt;&lt;/td&gt; &lt;/tr&gt; &lt;/tbody&gt; &lt;/table&gt; &lt;/fieldset&gt; &lt;/form&gt; &lt;!--Code PHP --&gt; &lt;?php if(isset(\$_FILES['fich'])) { echo "Taille maximale autorisée :",\$_POST["MAX_FILE_SIZE"]," octets&lt;br /&gt;"; echo "&lt;b&gt;Clés et valeurs du tableau \$_FILES &lt;/b&gt;&lt;br /&gt;"; foreach(\$_FILES["fich"] as \$cle =&gt; \$valeur) { echo "clé : \$cle valeur : \$valeur &lt;br /&gt;"; } //Enregistrement et renommage du fichier \$result=move_uploaded_file(\$_FILES["fich"]["tmp_name"],"imagephp.gif"); if(\$result==TRUE) {echo "&lt;br /&gt;&lt;big&gt;Le transfert est réalisé !&lt;/big&gt;";} else {echo "&lt;br /&gt; Erreur de transfert n°",\$_FILES["fich"]["error"];} } ?&gt; &lt;/body&gt;</pre>

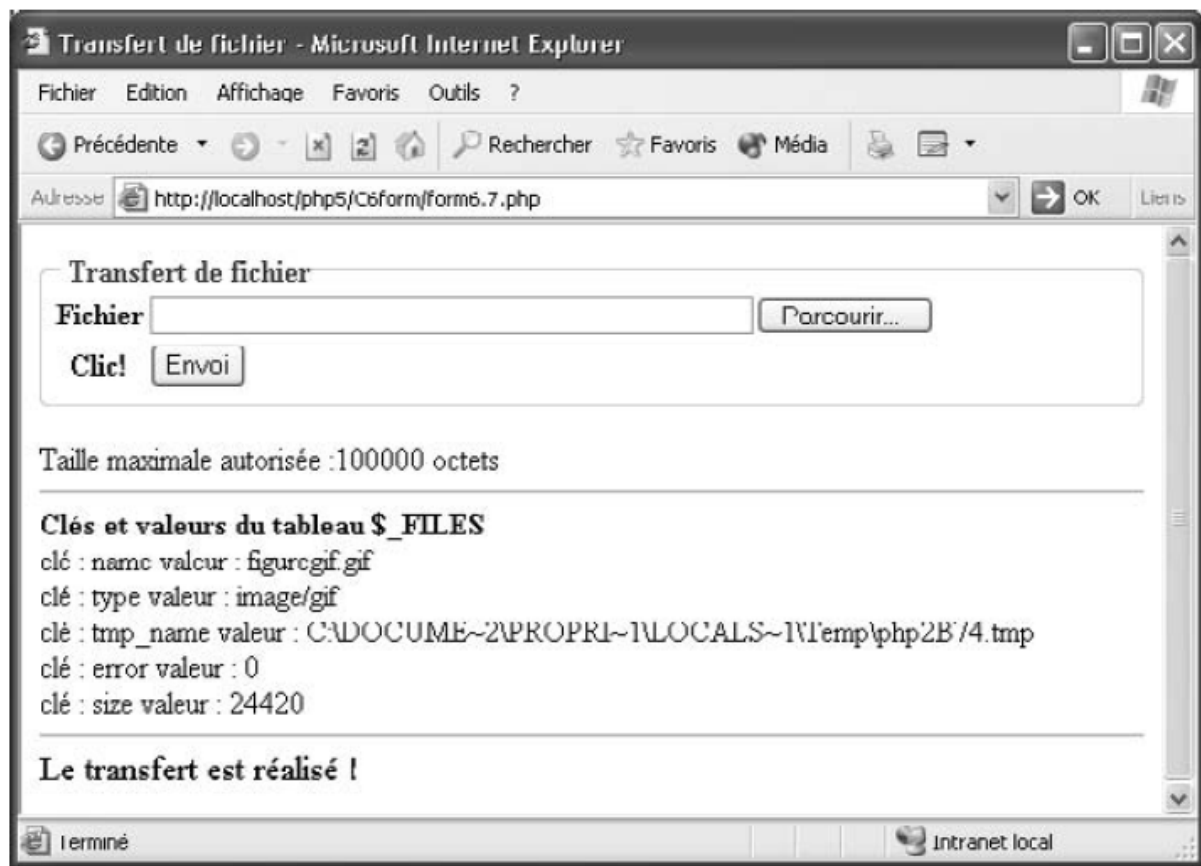
</html>

### Résultat affiché :

Avant de cliquer sur le bouton Envoi :



Après avoir cliqué sur le bouton Envoi :





**Tableau 24 : Exemple de gestion de boutons d'envoi multiples (calculatrice en ligne) :**

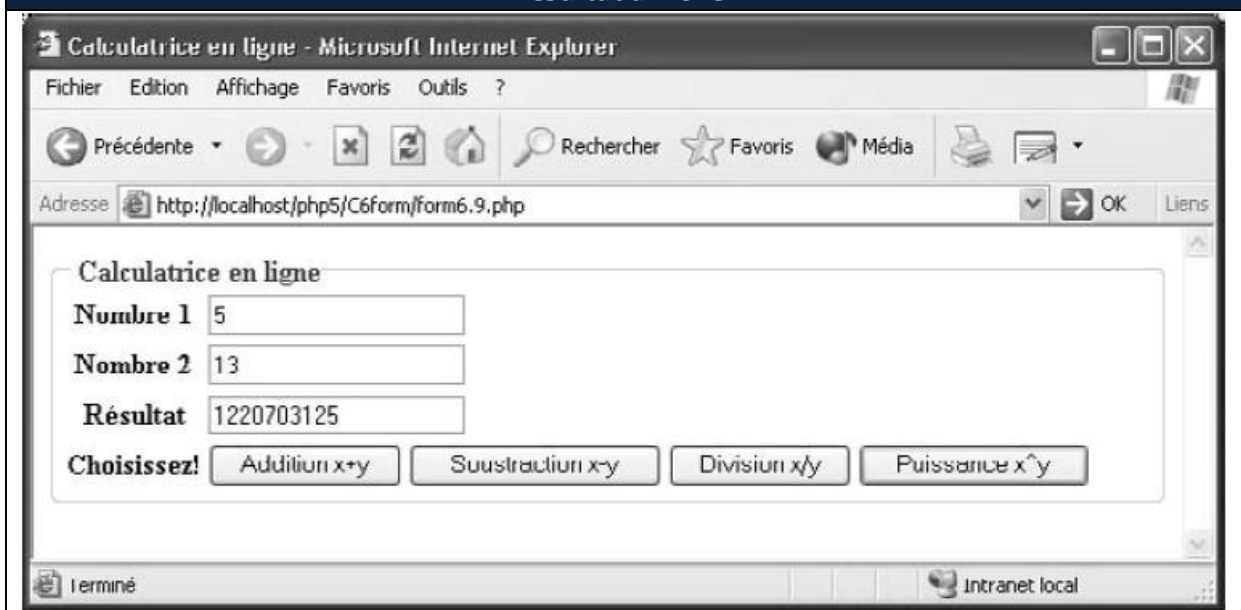
Script PHP
<pre>&lt;html&gt; &lt;head&gt; &lt;title&gt;Calculatrice en ligne&lt;/title&gt; &lt;meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1" /&gt; &lt;/head&gt; &lt;body&gt; &lt;!--Code PHP --&gt; &lt;?php if(isset(\$_POST["calcul"])&amp;&amp;isset(\$_POST["nb1"])&amp;&amp;isset(\$_POST["nb2"])) { switch(\$_POST["calcul"]) { case "Addition x+y": \$resultat= \$_POST["nb1"]+\$_POST["nb2"]; break; case "Soustraction x-y": \$resultat= \$_POST["nb1"]-\$_POST["nb2"]; break; case "Division x/y": \$resultat= \$_POST["nb1"]/\$_POST["nb2"]; break; case "Puissance x^y": \$resultat= pow(\$_POST["nb1"],\$_POST["nb2"]); break; } } else { echo "&lt;h3&gt;Entrez deux nombres : &lt;/h3&gt;"; } ?&gt; &lt;!-- Code HTML du formulaire --&gt; &lt;form action="&lt;?=\$_SERVER['PHP_SELF']?&gt;" method="post" &gt; &lt;fieldset&gt; &lt;legend&gt;&lt;b&gt;Calculatrice en ligne&lt;/b&gt;&lt;/legend&gt; &lt;table&gt; &lt;tbody&gt; &lt;tr&gt; &lt;th&gt;Nombre X&lt;/th&gt; &lt;td&gt; &lt;input type="text" name="nb1" value="&lt;?php if(isset(\$_POST["nb1"])) echo \$_POST['nb1'];else echo" ?&gt;" /&gt; &lt;/td&gt; &lt;/tr&gt; &lt;/tbody&gt; &lt;/table&gt;</pre>

```

<tr>
<th>Nombre Y</th>
<td> <input type="text" name="nb2" value="<?php if(isset($_POST["nb2"]))
echo $_POST['nb2'];else echo" ?>"/>
</td>
</tr>
<tr>
<th>Résultat </th>
<td> <input type="text" name="result" value="<?php if(isset($resultat))
echo $resultat;else echo" ?>"/>
</td>
</tr>
<tr>
<th>Choisissez!</th>
<td>
<input type="submit" name="calcul" value="Addition x+y" />
<input type="submit" name="calcul" value="Soustraction x-y" />
<input type="submit" name="calcul" value="Division x/y" />
<input type="submit" name="calcul" value="Puissance x^y" />
</td>
</tr>
</tbody>
</table>
</fieldset>
</form>
</body>
</html>

```

#### Résultat affiché :

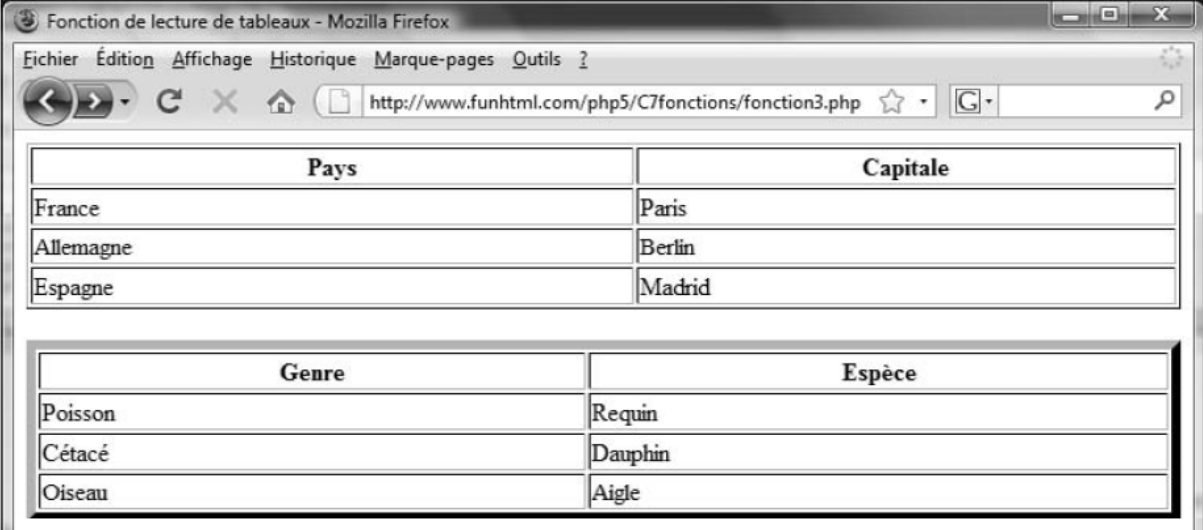


**Tableau 25 : Exemple d'appel d'une fonction dans un script PHP :**

**Script PHP**

```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<title>Fonction de lecture de tableaux</title>
</head>
<body>
<div>
<?php
//Définition de la fonction
function tabuni($tab,$bord,$lib1,$lib2)
{
echo "<table border=\"\$bord\" width=\"100%\"><tbody><tr><th>\$lib1</th><th>\$lib2
</th></tr>";
foreach($tab as $cle=>$valeur)
{
echo "<tr><td>\$cle</td> <td>\$valeur </td></tr>";
}
echo "</tbody> </table><br />";
}
//Définition des tableaux
$tab1 = array("France"=>"Paris","Allemagne"=>"Berlin","Espagne"=>"Madrid");
$tab2 = array("Poisson"=>"Requin","Cétacé"=>"Dauphin","Oiseau"=>"Aigle");
//Appels de la fonction
tabuni($tab1,1,"Pays","Capitale");
tabuni($tab2,6,"Genre","Espèce");
?>
</div>
</body>
</html>
```

**Résultat affiché :**



Pays	Capitale
France	Paris
Allemagne	Berlin
Espagne	Madrid

Genre	Espèce
Poisson	Requin
Cétacé	Dauphin
Oiseau	Aigle

**Tableau 26 : Exemple de fonctions retournant plusieurs valeurs :**

Script PHP
<pre>&lt;html&gt; &lt;head&gt; &lt;meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" /&gt; &lt;title&gt;Nombres complexes&lt;/title&gt; &lt;/head&gt; &lt;body&gt; &lt;div&gt; &lt;?php function modarg(\$reel,\$imag) {     \$mod =sqrt(\$reel*\$reel + \$imag*\$imag);     \$arg = atan2(\$imag,\$reel);     return array("module"=&gt;\$mod,"argument"=&gt;\$arg); } //Appels de la fonction \$a= 5; \$b= 8; \$complex= modarg(\$a,\$b); echo "&lt;b&gt;Nombre complexe \$a + \$b i:&lt;/b&gt;&lt;br /&gt; module = ", \$complex["module"], "&lt;br /&gt;argument = ", \$complex["argument"]," radians&lt;br /&gt;"; ?&gt; &lt;/div&gt; &lt;/body&gt; &lt;/html&gt;</pre>
Résultat affiché :
<p>Nombre complexe 5 + 8 i: module = 9.4339811320566 argument = 1.0121970114513 radians</p>

**Tableau 27 : Exemple de fonctions avec paramètres par défaut :**

Script PHP
<pre>&lt;html&gt; &lt;head&gt; &lt;meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" /&gt; &lt;title&gt;Fonction avec une valeur de paramètre par défaut&lt;/title&gt; &lt;/head&gt; &lt;body&gt; &lt;div&gt; &lt;?php function ht(\$prix,\$tax=19.6) { return "Prix Hors Taxes :". round(\$prix*(1-\$tax/100),2); } \$prix=154; echo "Prix TTC= \$prix &amp;euro; ",ht(\$prix)," &amp;euro;&lt;br /&gt;"; echo "Prix TTC= \$prix &amp;euro; ",ht(\$prix,19.6)," &amp;euro;&lt;br /&gt;"; echo "Prix TTC= \$prix &amp;euro; ",ht(\$prix,5.5)," &amp;euro;&lt;br /&gt;"; ?&gt; &lt;/div&gt; &lt;/body&gt; &lt;/html&gt;</pre>

**Tableau 28-1 : Exemple 1 de fonctions avec un nombre variable de paramètres (utilisation de type array):**

Script PHP
<pre>&lt;html&gt; &lt;head&gt; &lt;meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" /&gt; &lt;title&gt;Nombre de paramètres variable&lt;/title&gt; &lt;/head&gt; &lt;body&gt; &lt;div&gt; &lt;?php function prod(\$tab) {     \$n=count(\$tab);     echo "Il y a \$n paramètres :";     \$prod = 1;     foreach(\$tab as \$val)     {         echo "\$val, ";         \$prod *=\$val;     }     echo " le produit vaut ";     return \$prod; } \$tab1= range(1,10); echo "Produit des nombres de 1 à 10 : ", prod(\$tab1),"&lt;br /&gt;"; \$tab2 = array(7,12,15,3,21); echo "Produit des éléments :", prod(\$tab2),"&lt;br /&gt;"; ?&gt; &lt;/div&gt; &lt;/body&gt; &lt;/html&gt;</pre>
Résultat affiché
<p>Produit des nombres de 1 à 10 : Il y a 10 paramètres :1, 2, 3, 4, 5, 6, 7, 8, 9, 10, le produit vaut 3628800</p> <p>Produit des éléments :Il y a 5 paramètres :7, 12, 15, 3, 21, le produit vaut 79380</p>

**Tableau 28-2 : Exemple 2 de fonctions avec un nombre variable de paramètres (utilisation des fonctions particulières):**

Script PHP
<pre> &lt;html&gt; &lt;head&gt; &lt;meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" /&gt; &lt;title&gt;Produit d'un nombre indéterminé d'arguments&lt;/title&gt; &lt;/head&gt; &lt;body&gt; &lt;div&gt; &lt;?php //Utilisation de func_num_arg() et func_get_arg() function prod1() { \$prod = 1; //détermine le nombre d'arguments \$n=func_num_args(); for(\$i=0;\$i&lt;\$n;\$i++) { echo func_get_arg(\$i); (\$i==(\$n-1)?print(" = "):print(" * ")); \$prod *=func_get_arg(\$i); } return \$prod; } //Appels de la fonction prod1() echo "Produit des nombres :", prod1(5,6,7,8,11,15),"&lt;br /&gt;"; \$a=55;\$b=22;\$c=5;\$d=9; echo "Produit de ",prod1(\$a,\$b,\$c,\$d),"&lt;hr /&gt;"; //***** //Utilisation de func_get_args() seule //***** function prod2() { \$prod = 1; //Récupération des paramètres dans un tableau \$tabparam = func_get_args(); foreach(\$tabparam as \$cle=&gt;\$val) { //Présentation echo \$val; (\$cle==count(\$tabparam)-1)?print(" = "):print(" * "); //Calcul du produit \$prod *=\$val; } return \$prod; </pre>

```
}  
echo "Produit des nombres :", prod2(5,6,7,8,11,15),"<br />";  
$a=55;$b=22;$c=5;$d=9;  
echo "Produit de ",prod2($a,$b,$c,$d),"<hr />";  
?>  
</div>  
</body>  
</html>
```

#### Résultat affiché

Produit des nombres :  $5 * 6 * 7 * 8 * 11 * 15 = 277200$   
Produit de  $55 * 22 * 5 * 9 = 54450$

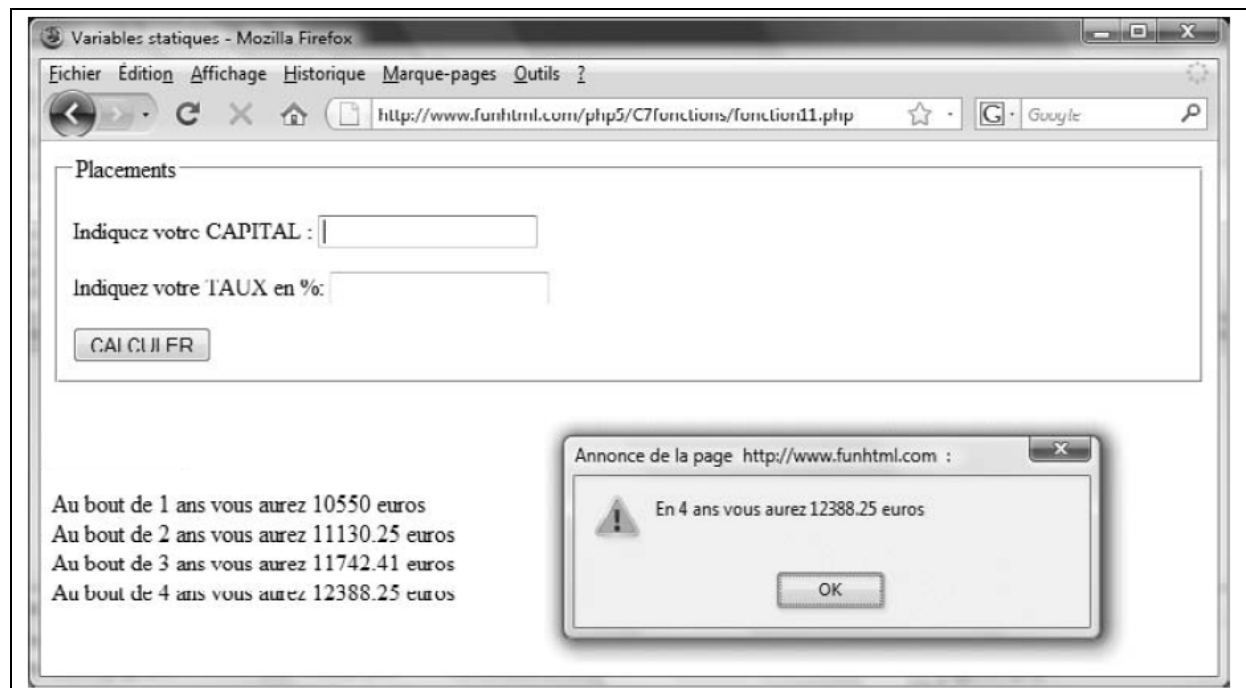


**Tableau 29 : Exemple d'utilisation des variables locales et globales dans un script PHP :**

Script PHP
<pre>&lt;html&gt; &lt;head&gt; &lt;meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" /&gt; &lt;title&gt;Variables locales et variables globales&lt;/title&gt; &lt;/head&gt; &lt;body&gt; &lt;div&gt; &lt;?php \$externe="dehors"; \$interne ="dedans"; function waytu(\$interne) { \$interne = "Si tu me cherches, je suis ".\$interne ." &lt;br /&gt;"; \$externe = "n'importe quoi!"; return \$interne; } echo waytu(\$interne);//affiche "Si tu me cherches, je suis dedans" echo \$interne," &lt;br /&gt;";//affiche "dedans" echo waytu(\$externe);//affiche "Si tu me cherches, je suis dehors" echo \$externe," &lt;br /&gt;";//affiche "dehors"  /*****Utilisation de variables globales dans une fonction*****/ function message(\$machin) { global \$truc; global \$intro; \$machin =" \$intro je suis \$truc \$machin &lt;br /&gt;"; \$truc = "zzzzzzzzzzzzzzzzzzzzzz!"; return \$machin; } \$intro= "Ne me cherches pas,"; \$truc = " parti "; echo message(" à Londres");// affiche "Ne me cherches pas, je suis partie à Londres" \$intro= "Si tu me cherches,"; \$truc=" revenu "; echo message(" de Nantes");// affiche "Si tu me cherches, je suis revenue de Nantes" echo \$truc; // affiche "zzzzzzzzzzzzzzzzzzzzzz!" ?&gt; &lt;/div&gt; &lt;/body&gt; &lt;/html&gt;</pre>

**Tableau 30 : Exemple d'utilisation des variables statiques :**

Script PHP
<pre>&lt;html&gt; &lt;head&gt; &lt;meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" /&gt; &lt;title&gt;Variables statiques&lt;/title&gt; &lt;/head&gt; &lt;body&gt; &lt;form method="post" action="fonction11.php" &gt; &lt;fieldset&gt; &lt;legend&gt;Placements&lt;/legend&gt; &lt;p&gt;Indiquez votre CAPITAL : &lt;input type="text" name="capital" value=""/&gt;&lt;/p&gt; &lt;p&gt;Indiquez votre TAUX en %: &lt;input type="text" name="taux" value=""/&gt;&lt;/p&gt; &lt;input type="submit" name="calcul" value="CALCULER"/&gt;&lt;br /&gt; &lt;/fieldset&gt; &lt;/form&gt; &lt;/body&gt; &lt;/html&gt; &lt;?php //Définition de la fonction function acquis(\$capital,\$taux) { static \$acquis=1; static \$an=1; \$coeff = 1+\$taux/100; \$acquis *= \$coeff; echo "&lt;script type='\"text/javascript\"' &gt; alert('En \$an ans vous aurez ". round(\$capital*\$acquis,2) ." euros') &lt;/script&gt;"; \$an++; return round(\$capital*\$acquis,2); } //Utilisation de la fonction if(isset(\$_POST["taux"])&amp;&amp; isset(\$_POST["capital"])&amp;&amp;is_numeric(\$_POST["capital"]) &amp;&amp; is_numeric(\$_POST["taux"])) { for(\$i=1;\$i&lt;5;\$i++) { echo "Au bout de \$i ans vous aurez ". acquis(\$_POST["capital"],\$_POST["taux"]). " euros &lt;br /&gt;"; } } ?&gt;</pre>
Résultat affiché



**Tableau 31 : Exemple de passage de paramètres par référence :**

Script PHP
<pre>&lt;html&gt; &lt;head&gt; &lt;meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" /&gt; &lt;title&gt;Passage par référence&lt;/title&gt; &lt;/head&gt; &lt;body&gt; &lt;div&gt; &lt;?php /***** passage par référence systématique *****/ //Définition de la fonction function prod(&amp;\$tab,\$coeff) { foreach(\$tab as \$cle=&gt;\$val) { if(is_numeric(\$val)) {\$tab[\$cle]*=\$coeff;} else { echo "Erreur : Le tableau est non numérique &lt;br /&gt;"; return FALSE; } } return \$tab; } echo "Tableau numérique &lt;br /&gt;"; \$tabnum = range(1,7); echo "Tableau avant l'appel &lt;br /&gt;",print_r( \$tabnum),"&lt;br /&gt;"; //Passage du tableau à la fonction \$result= prod(\$tabnum,3.5); echo "Tableau résultat &lt;br /&gt;",print_r( \$result),"&lt;br /&gt;"; echo "Tableau initial après l'appel &lt;br /&gt;",print_r( \$tabnum),"&lt;br /&gt;";  /***** passage par référence occasionnel *****/ //Définition de la fonction function &amp;hausse(\$prix,\$pourcent) { \$prix *=(1+\$pourcent/100); return round(\$prix,2); } //Utilisation de la fonction \$prix = 1500; echo hausse(\$prix,12),"&lt;br /&gt;"; // passage par valeur echo \$prix,"&lt;br /&gt;"; echo hausse(&amp;\$prix,12),"&lt;br /&gt;"; // passage par référence</pre>

```
echo $prix,"<br />";  
?>  
</div>  
</body>  
</html>
```

### Résultat affiché

Tableau numérique  
Tableau avant l'appel  
Array ( [0] => 1 [1] => 2 [2] => 3 [3] => 4 [4] => 5 [5] => 6 [6] => 7 ) 1  
Tableau résultat  
Array ( [0] => 3.5 [1] => 7 [2] => 10.5 [3] => 14 [4] => 17.5 [5] => 21 [6] => 24.5 ) 1  
Tableau initial après l'appel  
Array ( [0] => 3.5 [1] => 7 [2] => 10.5 [3] => 14 [4] => 17.5 [5] => 21 [6] => 24.5 ) 1  
1680  
1500  
1680  
1680

**Tableau 32 : Exemple d'utilisation des fonctions dynamiques :**

Script PHP
<pre>&lt;html&gt; &lt;head&gt; &lt;meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" /&gt; &lt;title&gt;Fonctions dynamiques&lt;/title&gt; &lt;/head&gt; &lt;body&gt; &lt;form method="post" action="fonction14.php" &gt; &lt;fieldset&gt; &lt;legend&gt;Choisissez votre fonction et son paramètre&lt;/legend&gt; &lt;input type="text" name="fonction" value="&lt;?= isset(\$_POST["fonction"]) ? \$_POST["fonction"] : "" ?&gt;" /&gt; &lt;input type="text" name="param" value="&lt;?= isset(\$_POST["param"]) ? \$_POST["param"] : "" ?&gt;" /&gt; &lt;input type="submit" value="Calculer" /&gt; &lt;/fieldset&gt; &lt;/form&gt; &lt;!-- Code PHP : gestion du formulaire--&gt; &lt;?php if((isset(\$_POST["fonction"])&amp;&amp; \$_POST["fonction"]!="") &amp;&amp; \$_POST["param"]!="") { \$fonction = \$_POST["fonction"]; \$params = \$_POST["param"]; if(function_exists(\$fonction)) { echo "Résultat : \$fonction(\$params) = ",\$fonction(\$params); } else echo "ERREUR DE FONCTION!"; } ?&gt; &lt;/body&gt; &lt;/html&gt;</pre>
Résultat affiché


**Tableau 33 : Exemple d'utilisation des fonctions conditionnelles :**

Script PHP
<pre>&lt;?php test(); //salut(); Cet appel provoquerait une erreur fatale \$heure=date("H"); //Définition d'une fonction conditionnelle if(\$heure&lt;18) { function salut() { echo "Bonjour : Fonction accessible seulement avant 18h00 &lt;br /&gt;"; } } else { function salut() { echo "Bonsoir : Fonction accessible seulement après 18h00 &lt;br /&gt;"; } } //Définition d'une fonction ordinaire function test() { echo "Fonction accessible partout &lt;br /&gt;"; return TRUE; } salut(); ?&gt;</pre>

**Tableau 34 : Exemple d'utilisation des fonctions récursives (Fonction qui retourne la factorielle d'un nombre entier n) :**

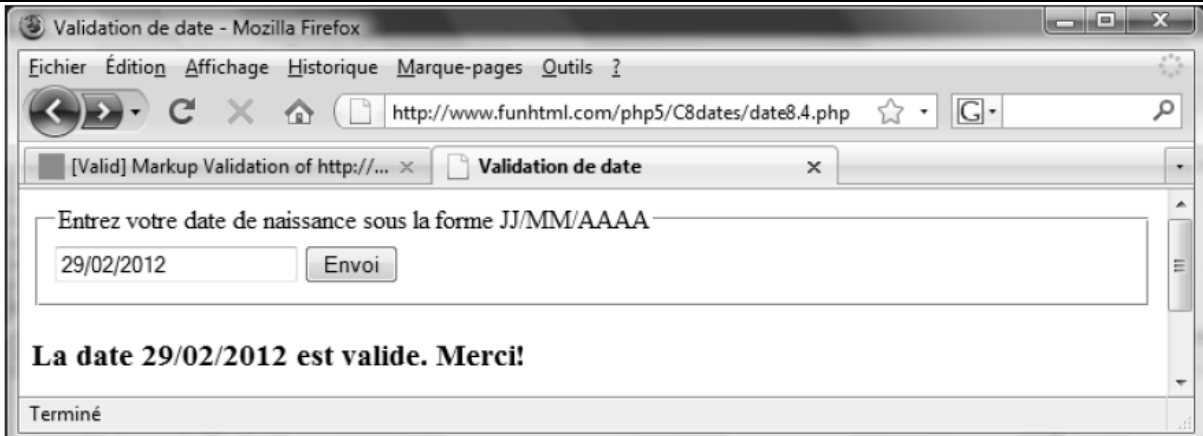
Script PHP
<pre>&lt;?php function facto(\$n) { if (\$n==0) return 1; else {return \$n*facto(\$n-1);} } echo "factorielle = ",facto(150); ?&gt;</pre>



**Tableau 35 : Exemple de définition d'une date :**

Script PHP
<pre>&lt;html&gt; &lt;head&gt; &lt;meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" /&gt; &lt;title&gt;Dates et durées&lt;/title&gt; &lt;/head&gt; &lt;body&gt; &lt;div&gt; &lt;?php //la fonction mktime() \$timepasse= mktime(12,5,30,5,30,1969); \$timeaujour = time(); \$duree = \$timeaujour-\$timepasse; echo "Entre le 30/05/1969 à 12h05m30s et maintenant il s'est écoulé",\$duree," secondes &lt;br /&gt;"; echo "Soit ",round(\$duree/3600), " heures &lt;br /&gt;"; echo "Ou encore ",round(\$duree/3600/24)," jours &lt;br /&gt;"; \$timefutur = mktime(12,5,30,12,25,2008); \$noel = \$timefutur-\$timeaujour; echo "Plus que ",\$noel, "secondes entre maintenant et Noël, soit ", round(\$noel/3600/24)," jours, Patience! &lt;br /&gt;"; //la fonction gmmktime() \$timepassegmt = gmmktime(12,5,30,5,30,1969); echo "Timestamp serveur pour le 30/5/1969= ",\$timepasse,"&lt;br /&gt;"; echo "Timestamp GMT pour le 30/5/1969= ",\$timepassegmt,"&lt;br /&gt;"; ?&gt; &lt;/div&gt; &lt;/body&gt; &lt;/html&gt;</pre>
Résultat affiché
<p>Entre le 30/05/1969 à 12h05m30s et maintenant il s'est écoulé 1243146454 secondes Soit 345318 heures Ou encore 14388 jours Plus que 5679146secondes entre maintenant et Noël, soit 66 jours, Patience! Timestamp serveur pour le 30/5/1969= -18622470 Timestamp GMT pour le 30/5/1969= -18618870</p>

**Tableau 36 : Exemple de vérification d'une date :**

Script PHP
<pre>&lt;html&gt; &lt;head&gt; &lt;meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" /&gt; &lt;title&gt;Validation de date&lt;/title&gt; &lt;/head&gt; &lt;body&gt; &lt;form method="post" action="&lt;?= \$_SERVER["PHP_SELF"] ?&gt;" &gt; &lt;fieldset&gt; &lt;legend&gt;Entrez votre date de naissance sous la forme JJ/MM/AAAA&lt;/legend&gt; &lt;input type="text" name="date" /&gt; &lt;input type="submit" value="Envoi"/&gt; &lt;/fieldset&gt; &lt;/form&gt; &lt;?php //checkdate if(isset(\$_POST["date"])) { \$date=\$_POST["date"]; \$tabdate=explode("/", \$date); if(!checkdate(\$tabdate[1], \$tabdate[0], \$tabdate[2])) {echo "&lt;hr /&gt;La date \$date n'est pas valide. Recommencez! &lt;hr /&gt;";} else {echo "&lt;h3&gt; La date \$date est valide. Merci!&lt;/h3&gt;";} } ?&gt; &lt;/body&gt; &lt;/html&gt;</pre>
Résultat affiché


## Tableau 37 :

**Tableau : Caractères de définition du format d'affichage**

Caractère de définition	Définition et résultat affiché
y	L'année en deux chiffres (05 pour 2005)
Y	L'année en quatre chiffres (2005)
L	Affiche 1 si l'année est bissextile et 0 sinon.
m	Le mois en deux chiffres de 01 à 12
n	Le mois en un ou deux chiffres de 1 à 12
M	Le mois en trois lettres (en anglais)
F	Le mois en toutes lettres (en anglais)
t	Le nombre de jours du mois de 28 à 31
d	Le jour du mois en deux chiffres de 01 à 31
j	Le jour du mois en un chiffre de 1 à 31
D	Le jour de la semaine en trois lettres (en anglais)
l (petit L)	Le jour de la semaine en toutes lettres (en anglais)
w	Le jour de la semaine codé de 0 pour dimanche à 6 pour samedi
S	Affiche le suffixe anglais « th » ou « nd » après les chiffres du jour.
z	Le jour de l'année de 0 à 366
g	Les heures de 1 à 12 (avec AM et PM)
h	Les heures de 01 à 12 (avec AM et PM)
G	Les heures de 0 à 23
H	Les heures sur deux chiffres de 00 à 23
a	Ajoute « am » pour le matin ou « pm » pour l'après-midi.
A	Ajoute « AM » pour le matin ou « PM » pour l'après-midi.
i	Les minutes en deux chiffres de 00 à 59
s	Les secondes en deux chiffres de 00 à 59
U	Affiche le timestamp UNIX.
Z	Donne le décalage horaire par rapport au temps GMT ou UTC en seconde, de - 43 200 à 43 200.
T	Affiche la ville significative du fuseau horaire, par exemple « Paris, Madrid ».
I	Affiche 0 pendant l'heure d'hiver et 1 pendant l'heure d'été.
r	Affiche la date complète au format RFC-822, par exemple: « Sun, 13 Apr 2003 22:34:46 +0200 ».
B	Heure Internet Swatch : invention de la société Swatch selon laquelle 24 heures sont divisées en 1 000 éléments nommés « beats ». Par exemple, midi vaut 500 beats.

### **Tableau 38 :**

**Tableau – Clés du tableau retourné par la fonction *getdate()***

Clé	Description
wday	Le jour de la semaine sous forme d'entier de 0 (dimanche) à 6 (samedi)
weekday	Le jour de la semaine sous forme de chaîne (en anglais)
mday	Le jour du mois sous forme d'entier de 0 à 31
mon	Le mois sous forme d'entier de 1 à 12
month	Le mois sous forme de chaîne (en anglais)
year	L'année en entier sur 4 chiffres
hours	L'heure de 0 à 23
minutes	Les minutes de 0 à 59
seconds	Les secondes de 0 à 59
yday	Le jour de l'année de 1 à 366
0	Le timestamp correspondant à la date

## Tableau 39 :

**Tableau – Caractères de formatage de la fonction `strftime()`**

Caractère	Description
%a	Jour de la semaine abrégé
%A	Jour de la semaine en entier
%b	Mois abrégé
%B	Mois en entier
%c	Affiche la date et l'heure au format local (exemple 24/ 04/2005 15:32:52 si la langue est le français).
%C	Numéro du siècle
%d	Jour du mois numérique de 01 à 31
%D	Équivalent de l'ensemble '%m%d%y'
%e	Jour du mois de 1 à 31 précédé d'une espace
%h	Équivalent de %b
%H	Nombre d'heures de 00 à 23
%I	Nombre d'heures de 00 à 12 (voir %p pour afficher am ou pm)
%j	Numéro du jour de l'année de 1 à 366
%m	Numéro du mois de 1 à 12
%H	Nombre de minutes de 0 à 59
%n	Saut de ligne
%p	Affiche « am » ou « pm » selon l'heure
%S	Nombre de secondes
%t	Équivalent de la tabulation /t
%T	Équivalent de l'ensemble '%H:%M:%S'
%u	Le jour de la semaine de 1 pour lundi à 7 pour dimanche (attention cette notation est différente de celle des fonctions <code>date</code> et <code>getdate</code> ).
%U	Numéro de la semaine (la première semaine commençant avec le premier dimanche de l'année, les jours qui précèdent ne comptent pas).
%V	Numéro de la semaine ISO de 01 à 53. La première semaine est celle qui a plus de 4 jours ; le lundi est le premier jour de la semaine.
%w	Le jour de la semaine de 0 pour dimanche à 6 pour samedi
%W	Numéro de la semaine (la première semaine commençant avec le premier lundi de l'année, les jours qui précèdent ne comptent pas).
%x	Affiche la date au format local défini par <code>setlocale()</code> . Exemple JJ/MM/AAAA.
%X	Affiche l'heure au format local défini par <code>setlocale()</code> . Exemple HH:MM:SS.
%y	L'année sur deux chiffres de 00 à 99
%Y	L'année sur quatre chiffres
%Z	Les villes correspondant au fuseau horaire
%%	Affiche le caractère « % » seul.

**Tableau 46-1 : Exemple 1 : Compteur de visites à l'aide d'un fichier :**

Script PHP
<pre>&lt;html&gt; &lt;head&gt; &lt;meta http-equiv="content-type" content="text/html; charset=iso-8859-1"&gt; &lt;title&gt;titre&lt;/title&gt; &lt;/head&gt; &lt;body&gt; &lt;?php if(file_exists("compteur.txt")) { if(\$id_file=fopen("compteur.txt","r")) { flock(\$id_file,2); \$nb=fread(\$id_file,10); \$nb++; flock(\$id_file,3); fclose(\$id_file); \$id_file=fopen("compteur.txt","w"); flock(\$id_file,2); fwrite(\$id_file,\$nb); flock(\$id_file,3); fclose(\$id_file); } else {echo "Echec d'ouverture du fichier !"; } } else { \$nb=10000; \$id_file=fopen("compteur.txt","w"); flock(\$id_file,2); fwrite(\$id_file,\$nb); flock(\$id_file,3); fclose(\$id_file); } echo "&lt;table border=\"1\" style=\"font-size:2em;\"&gt; &lt;tr&gt; &lt;td style=\"background-color:blue;color:white;\"&gt;Voici déjà &lt;/td&gt; &lt;td style=\"font-size:1.2em;background-color:white;\"&gt; \$nb &lt;/td&gt; &lt;td style=\"background-color:red;\"&gt; visites sur le site &lt;/td&gt;&lt;/tr&gt;&lt;/table&gt; "; ?&gt; &lt;/body&gt; &lt;/html&gt;</pre>
Résultat affiché sur le navigateur:

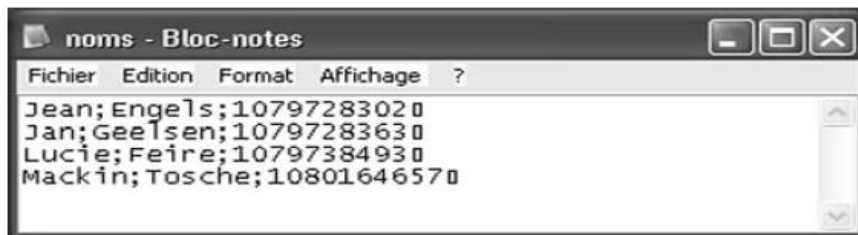
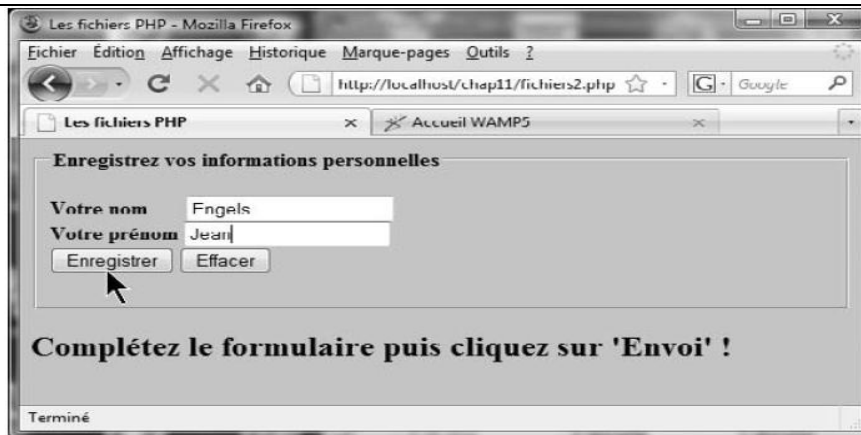


### Tableau 46-2 : Exemple 2 : Saisie et enregistrement à partir d'un formulaire :

[illegible]

**Résultat affiché sur le navigateur:**





**Tableau 47 : Exemple d'utilisation des fonctions de lecture d'un fichier dans un script PHP :**

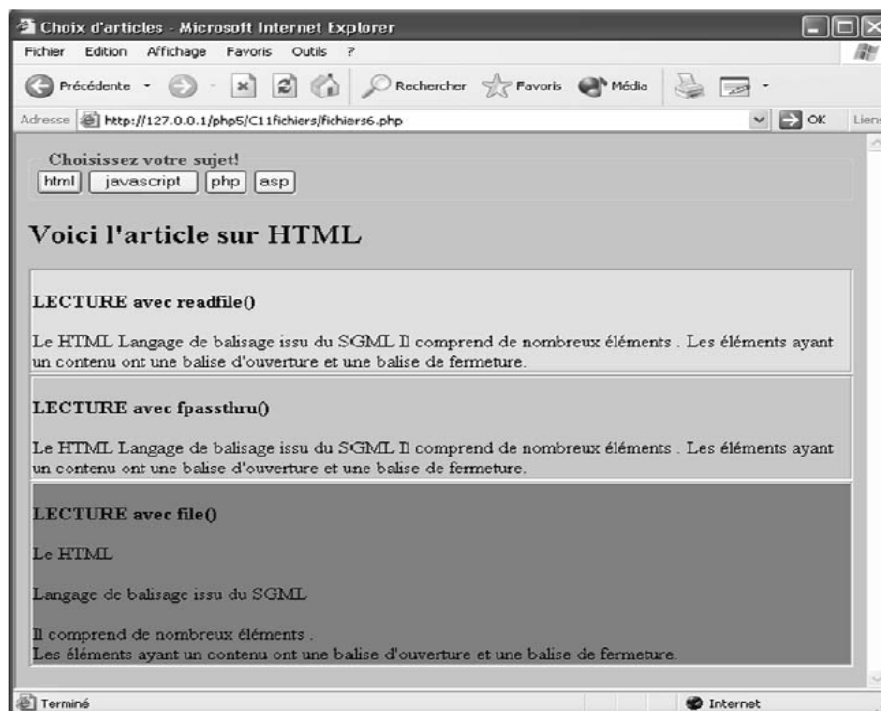
Script PHP
<pre> &lt;html&gt; &lt;head&gt; &lt;meta http-equiv="content-type" content="text/html; charset=iso-8859-1"&gt; &lt;title&gt;Choix d'articles &lt;/title&gt; &lt;/head&gt; &lt;body style="background-color: #ffcc00;"&gt; &lt;form action="&lt;?php echo \$_SERVER['PHP_SELF'] ?&gt;" method="post" &gt; &lt;fieldset&gt; &lt;legend&gt;&lt;b&gt;Choisissez votre sujet! &lt;/b&gt;&lt;/legend&gt; &lt;input type="submit" name="sujet" value="html" /&gt; &lt;input type="submit" name="sujet" value="javascript" /&gt; &lt;input type="submit" name="sujet" value="php" /&gt; &lt;input type="submit" name="sujet" value="asp" /&gt; &lt;/fieldset&gt; &lt;/form&gt; &lt;?php //AFFICHAGE if(isset(\$_POST['sujet'])) { \$sujet=\$_POST['sujet']; echo "&lt;h2&gt;Voici l'article sur ",strtoupper(\$sujet), "&lt;/h2&gt; "; //***** //Lecture du fichier avec readfile() //***** echo "&lt;div style=\" background-color:#FFCCFF ; border-width:3px ; border-style:groove; \" &gt;"; echo " &lt;h4&gt;LECTURE avec readfile()&lt;/h4&gt;"; readfile(\$sujet.".txt"); echo "&lt;/div&gt;"; //***** //Lecture du fichier avec fpassthru() //***** echo "&lt;div style=\" background-color:#FFAACC ; border-width:3px ; border-style:groove; \" &gt;"; echo " &lt;h4&gt;LECTURE avec fpassthru()&lt;/h4&gt;"; \$id_file=fopen(\$sujet.".txt","r"); fpassthru(\$id_file); echo "&lt;/div&gt;"; //***** //Lecture du fichier avec file() //***** echo "&lt;div style=\" background-color:#00AAFF ; border-width:3px ; border-style:groove; \" &gt;"; </pre>

```

echo " <h4>LECTURE avec file()</h4>";
$tab = file($sujet.".txt");
for($i=0;$i< count($tab); $i++)
{
echo $tab[$i],"<br>";
}
echo "</div>";
}
?>
</body>
</html>

```

### Résultat affiché sur le navigateur:



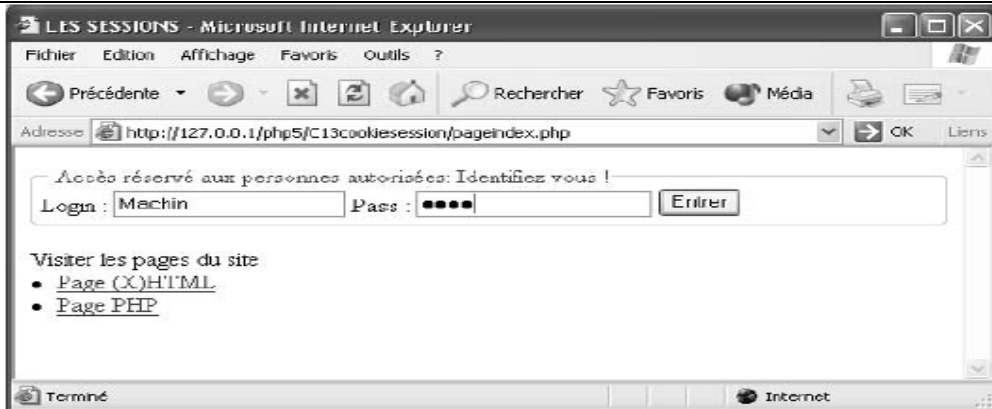
**Tableau 48 : Exemple 1 : Pages à accès réservé par une authentification :**

Script de la page d'accueil « <b>pageindex.php</b> » :
<pre>&lt;?php <b>session_start();</b> if(\$_POST['login']=="Machin" &amp;&amp; \$_POST['pass']=="4567") { \$_SESSION['acces']="oui"; <b>\$_SESSION['nom']=\$_POST['login'];</b> \$_SESSION['html']=0 ; \$_SESSION['php']=0 ; } ?&gt; &lt;html &gt; &lt;head&gt; &lt;meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" /&gt; &lt;title&gt;LES SESSIONS&lt;/title&gt; &lt;/head&gt; &lt;body&gt; &lt;div&gt; &lt;form method="post" action="&lt;?php echo \$_SERVER['PHP_SELF'] ?&gt;"&gt; &lt;fieldset&gt; &lt;legend&gt;Accès réservé aux personnes autorisées: Identifiez vous !&lt;/legend&gt; &lt;label&gt;Login : &lt;/label&gt;&lt;input type="text" name="login" /&gt; &lt;label&gt;Pass : &amp;nbsp;  &lt;/label&gt;&lt;input type="password" name="pass" /&gt; &lt;input type="submit" name="envoi" value="Entrer"/&gt; &lt;/fieldset&gt; &lt;/form&gt; Visiter les pages du site &lt;br /&gt; //***** //***** Session avec Cookie ***** //***** &lt;ul&gt; &lt;li&gt;&lt;a href="pagehtml.php"&gt;Page XHTML &lt;/a&gt;&lt;? if(isset(\$_SESSION['html'])) echo " vue ". \$_SESSION['html']. " fois"; ?&gt; &lt;/li&gt; &lt;li&gt;&lt;a href="pagephp.php"&gt;Page PHP 5&lt;/a&gt;&lt;? if(isset(\$_SESSION['php'])) echo " vue ". \$_SESSION['php']. " fois"; ?&gt; &lt;/li&gt; &lt;/ul&gt; //***** //***** Session sans Cookie ***** //***** /* &lt;ul&gt; &lt;li&gt;&lt;a href="pagehtml2.php?&lt;?php echo SID ?&gt;"&gt;Page XHTML &lt;/a&gt;&lt;?php if(isset(\$_SESSION['html'])) echo " vue ". \$_SESSION['html']. " fois"; ?&gt; &lt;/li&gt; &lt;li&gt;&lt;a href="pagephp2.php?&lt;?php echo SID ?&gt;"&gt;Page PHP 5&lt;/a&gt;&lt;?php if(isset(\$_SESSION['php'])) echo " vue ". \$_SESSION['php']. " fois"; ?&gt; &lt;/li&gt;</pre>

```

</ul>
*/
</div>
</body>
</html>

```

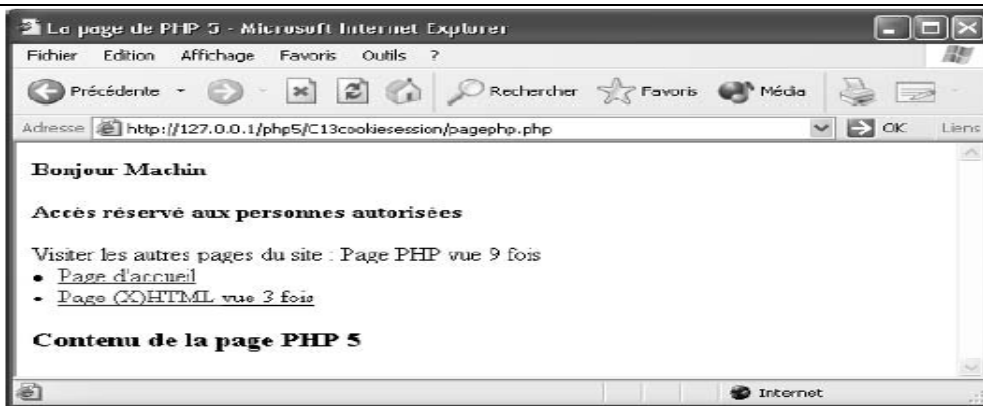


Script de la page « **pagephp.php** » :

```

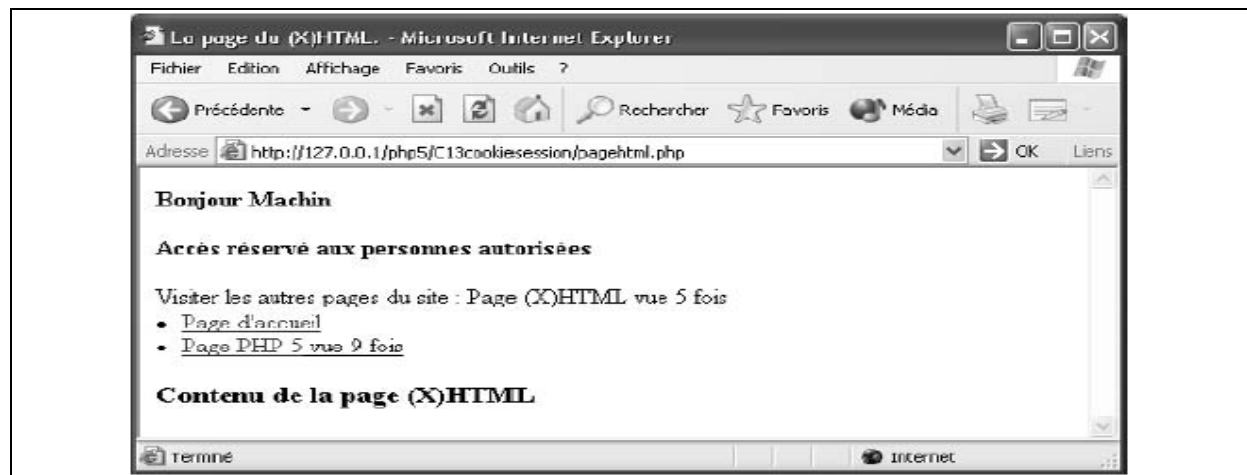
<?php
session_start();
if($_SESSION['acces']!="oui")
{
    header("Location:pageindex.php");
}
else
{
    echo "<h4>Bonjour ". $_SESSION['nom']."</h4>";
    $_SESSION['php'] ++;
}
?>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<title>La page de PHP 5</title>
</head>
<body>
<h4>Accès réservé aux personnes autorisées</h4>
<p> Visiter les autres pages du site :
<?php echo "Page PHP vue ". $_SESSION['php']. " fois"; ?>
<ul>
<li><a href="pageindex.php">Page d'accueil </a></li>
<li><a href="pagehtml.php">Page XHTML </a>
<? if(isset($_SESSION['html']))echo " vue ". $_SESSION['html']. " fois"; ?></li>
</ul>
</p>
<h3>Contenu de la page PHP 5</h3>
</body>
</html>

```



#### Script de la page « **pagehtml.php** » :

```
<?php
session_start();
if($_SESSION['acces']!="oui")
{
header("Location:pageindex.php");
}
else
{
echo "<h4>Bonjour ". $_SESSION['nom']."</h4>";
$_SESSION['html'] ++;
}
?>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<title>La page du (X)HTML.</title>
</head>
<body>
<h4> Accès réservé aux personnes autorisées</h4>
<p> Visiter les autres pages du site :
<?php echo "Page XHTML vue ". $_SESSION['html']. " fois"; ?>
<ul>
<li><a href="pageindex.php">Page d'accueil </a> </li>
<li><a href="pagephp.php">Page PHP 5</a>
<? if(isset($_SESSION['php']))echo " vue ". $_SESSION['php']. " fois"; ?> </li>
</ul>
<h3>Contenu de la page XHTML</h3>
</body>
</html>
```



**Tableau 49 : Exemple 2 : Gestion de panier et commande en ligne :**

```
Script PHP

<?php
session_start();
//AJOUTER
if($_POST["envoi"]=="AJOUTER" && $_POST["code"]!="" && $_POST["article"]!="" &&
$_POST["prix"]!="")
{
    $code=$_POST["code"];
    $article=$_POST["article"];
    $prix=$_POST["prix"];
    $_SESSION['code']=$_SESSION['code']."/".$code;
    $_SESSION['article']=$_SESSION['article']."/".$article;
    $_SESSION['prix']=$_SESSION['prix']."/".$prix;
}
//VERIFIER
if($_POST["envoi"]=="VERIFIER")
{
    echo "<table border='1' >";
    echo "<tr><td colspan='3'><b>Récapitulatif de votre commande</b></td>";
    echo "<tr><th>&nbsp;code&nbsp;</th><th>&nbsp;article&nbsp;</th><th>&nbsp;prix&nbsp;</th>";
    $prixtotal=0;
    $tab_code=explode("/*",$_SESSION['code']);
    $tab_article=explode("/*",$_SESSION['article']);
    $tab_prix=explode("/*",$_SESSION['prix']);
    for($i=1;$i<count($tab_code);$i++)
    {
        echo "<tr><td>{$tab_code[$i]}</td><td>{$tab_article[$i]}</td><td>".sprintf("%01.2f",
        $tab_prix[$i])."</td>";
        $prixtotal+=$tab_prix[$i];
    }
    echo "<tr><td colspan=2> PRIX TOTAL </td><td>".sprintf("%01.2f", $prixtotal)."</td>";
    echo "</table>";
}
//ENREGISTRER
if($_POST["envoi"]=="ENREGISTRER")
{
    $idfile=fopen("commande.txt","w");
    //
    $tab_code=explode("/*",$_SESSION['code']);
    $tab_article=explode("/*",$_SESSION['article']);
    $tab_prix=explode("/*",$_SESSION['prix']);
    for($i=0;$i<count($tab_code);$i++)
    {
        fwrite($idfile,$tab_code[$i]."; ".$tab_article[$i]."; ".$tab_prix[$i].";\n");
    }
}
```



```
}
fclose($idfile);
}
//LOGOUT
if($_POST["envoi"]=="LOGOUT")
{
    session_unset();
    session_destroy();
    echo "<h3>La session est terminée</h3>";
}
$_POST["envoi"]="";
?>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
<title>Gestion de panier</title>
</head>
<body>
<form action="<?php $_SERVER['PHP_SELF'] ?>" method="post" enctype="application/x-www-form-urlencoded">
<fieldset>
<legend><b>Saisies d'articles</b></legend>
<table>
<tbody>
<tr>
<th>code : </th>
<td><input type="text" name="code" /></td>
</tr>
<tr>
<th>article : </th>
<td><input type="text" name="article" /></td>
</tr>
<tr>
<th>prix : </th>
<td><input type="text" name="prix" /></td>
</tr>
<tr>
<td colspan="3">
<input type="submit" name="envoi" value="AJOUTER" />
<input type="submit" name="envoi" value="VERIFIER" />
<input type="submit" name="envoi" value="ENREGISTRER" />
<input type="submit" name="envoi" value="LOGOUT" />
</td>
</tr>
</tbody>
</table>
</fieldset>
```

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

### Page de saisie de commandes :

Gestion de panier - Mozilla Firefox

Fichier Édition Affichage Historique Marque-pages Outils ?

http://localhost/chap12/cooksess5.php

**Saisies d'articles**

code : 5431A

article : PHP 5

prix : 29.90

AJOUTER VERIFIER ENREGISTRER LOGOUT

Terminé

### Affichage du contenu du panier :

Récapitulatif de votre commande		
code	article	prix
5431A	PHP 5	29.90
7299D	MySQL	19.50
2531H	XHTML et CSS	29.90
PRIX TOTAL		79.30

**Tableau 50 : Exemple 1 : Envoi d'un e-mail de confirmation au format texte :**

Script PHP
<pre> &lt;html&gt; &lt;head&gt; &lt;meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" /&gt; &lt;style type="text/css"&gt; td {background-color:yellow;color:blue;font-family: arial, helvetica, sans-serif;font-size: 12pt;font-weight: bold;} &lt;/style&gt; &lt;title&gt;Votre commande&lt;/title&gt; &lt;/head&gt; &lt;body&gt; &lt;div&gt;&lt;h3&gt;Articles &lt;/h3&gt; "XHTML et CSS" : 29.90 ¤&lt;br /&gt;"PHP 5" : 29.50 ¤&lt;br /&gt;"MySQL" : 19.75 ¤&lt;br /&gt;&lt;br /&gt;&lt;/div&gt; &lt;form action="&lt;?=\$_SERVER['PHP_SELF'] ?&gt; " method="post" enctype="application/x- www-form-urlencoded" &gt; &lt;fieldset&gt; &lt;legend&gt;Passez votre commande&lt;/legend&gt; &lt;table border="0" &gt; &lt;tr&gt; &lt;td&gt;Article&lt;/td&gt; &lt;td&gt;&lt;input type="text" name="article" size="40" maxlength="256" /&gt;&lt;/td&gt; &lt;/tr&gt; &lt;tr&gt; &lt;td&gt;Quantité&lt;/td&gt; &lt;td&gt;&lt;input type="text" name="quantite" size="40" /&gt;&lt;/td&gt; &lt;/tr&gt; &lt;tr&gt; &lt;td&gt;Nom&lt;/td&gt; &lt;td&gt;&lt;input type="text" name="nom" size="40" maxlength="256" /&gt;&lt;/td&gt; &lt;/tr&gt; &lt;tr&gt; &lt;td&gt;Adresse&lt;/td&gt; &lt;td&gt;&lt;input type="text" name="adresse" size="40" maxlength="256" /&gt;&lt;/td&gt; &lt;/tr&gt; &lt;tr&gt; &lt;td&gt;Mail&lt;/td&gt; &lt;td&gt;&lt;input type="text" name="mail" size="40" maxlength="256" /&gt;&lt;/td&gt; &lt;/tr&gt; &lt;tr &gt; &lt;td colspan="2"&gt;&amp;nbsp;&amp;nbsp;&amp;nbsp;&lt;input type="submit" name="envoi" value="Commander" /&gt;&lt;/td&gt; &lt;/tr&gt; &lt;/table&gt; &lt;/fieldset&gt; &lt;/form&gt; </pre>

```

<!-- SCRIPT PHP -->
<?php
//Création du tarif des livres
$tarif= array("XHTML et CSS"=>29.90,"PHP 5"=>29.50,"MySQL"=>19.75);
//Gestion de la commande
if(isset($_POST['article'])&&isset($_POST['quantite'])&&isset($_POST['nom'])&&isset($_POST['adresse']) && isset($_POST['mail'])) )
{
    $article=$_POST['article'];
    $prix= $tarif[$article];
    $objet="Confirmation de commande";
    //Contenu du mail
    $text.= "Nous avons bien reçu votre commande de : \n";
    $text.="{"$_POST['quantite']} livres ";
    $text.= $_POST['article'] . " au prix unitaire de : ". $prix ." euros \n";
    $text.= "Soit un prix total de : ". $prix * $_POST['quantite'] ." euros \n";
    $text.= "Adresse de livraison : \n". $_POST['nom']. " \n";
    $text.= $_POST['adresse']. " \n";
    $text .=" Cordialement";
    if(mail($_POST['mail'],$objet,$text))
    {
        echo "<h1>Vous allez recevoir un mail de confirmation </ h1>";
    }
    else
    {
        echo "<h1>Le mail n'a pas été envoyé: recommencez! </h1>";
    }
}
?>
</body>
</html>

```

### Résultat affiché dans le navigateur



The screenshot shows a Microsoft Internet Explorer window titled "Votre commande - Microsoft Internet Explorer". The address bar displays "http://www.xxxphp.com/php5/C13cookiesession/mail1.php". The main content area shows a form titled "Passez votre commande" with the following fields:

<b>Article</b>	PHP5
<b>Quantité</b>	3
<b>Nom</b>	Anatole RICOD
<b>Adresse</b>	21 rue Compoint 75018 PARIS
<b>Mail</b>	aricod@tiscali.fr

Below the form is a button labeled "Commander". The status bar at the bottom indicates "Terminé" and "Internet".

**De :** webmaster@comp... **À :** ...s@...  
**Objet :** Confirmation de commande

Nous avons bien reçu votre commande de :  
3 livres PHP5 au prix unitaire de : 29 euros  
Soit un prix total de : 87 euros  
Adresse de livraison :  
Anatole RICOD  
21 rue Compont 75018 PARIS  
Cordialement

**L'e-mail reçu par le client**

**Tableau 51 : Exemple 2 : Envoi d'e-mail d'annonce d'un grand événement au format HTML:**

Script PHP

```

<?php
$dest = "abonne@machin.com";
$objet = "Test mail en HTML";
//Contenu HTML du mail
$texte = "<html><head><title>Envoi de mail HTML</title></head>
<body><h1>La bonne nouvelle du mois</h1>
<b>Sortie de PHP 5 version finale!</b>
/**** N.B : lors de l'insertion d'une image dans un e-mail au format HTML vous devez
précisez son chemin absolu ***/
<img src=http://static.php.net/www.php.net/images/php.gif alt=\"Logo PHP\" />
<a href=http://www.php.net.\">Plus d'infos ici</A>
<p>Télécharger un installateur pour une utilisation en local<br />
<a href=http://www.phpathome.fr.tc\">Le site PHP@Home</A>
</body></html>";
//En têtes indispensables pour un mail en HTML
$entete="MIME-Version: 1.0";
$entete .= "Content-Type:text/html;charset=iso-8859-1\n";
$entete .= "Content-Transfer-Encoding: 8bit\n";
$entete .= "From: engels@funhtml.com \n";
//Envoi du mail
if (mail($dest,$objet,$texte,$entete))
{
print "Le mail a été envoyé<br> <br />";
}
else
{
print "le mail n'a pas été envoyé<br />";
}
?>

```

Résultat affiché sur le navigateur

De : webmaster@funphp.com À : @funphp.com  
Objet : Test mail en HTML

## La bonne nouvelle du mois

**Sortie de PHP 5 version finale!**

[Plus d'infos ici](#)

Télécharger un installateur pour une utilisation en local

[Le site PHP@Home](#)

**Tableau 52 : Les types de données numériques en MySQL :**

Type	Définition et caractéristiques
TINYINT	Un très petit entier prenant des valeurs de $-128 (-2^7)$ à $127 (2^7 - 1)$ . S'il est suivi de l'option UNSIGNED, les valeurs sont positives et varient de 0 à 255 ( $2^8 - 1$ ). Chaque valeur occupe 1 octet.
SMALLINT	Un petit entier prenant des valeurs de $-32768 (-2^{15})$ à $32767 (2^{15} - 1)$ . S'il est suivi de l'option UNSIGNED, les valeurs sont positives et varient de 0 à 65535 ( $2^{16} - 1$ ). Chaque valeur occupe 2 octets.
MEDIUMINT	Entier moyen prenant des valeurs de $-8388608 (-2^{23})$ à $8388607 (2^{23} - 1)$ . S'il est suivi de l'option UNSIGNED, les valeurs sont positives et varient de 0 à 16777215 ( $2^{24} - 1$ ). Chaque valeur occupe 3 octets.
INT ou INTEGER	Entier prenant des valeurs de $-2147483648 (-2^{31})$ à $2147483647 (2^{31} - 1)$ . S'il est suivi de l'option UNSIGNED, les valeurs sont positives et varient de 0 à 4294967295 soit $2^{32} - 1$ . Chaque valeur occupe 4 octets.
BIGINT	Grand entier prenant des valeurs de $-9223372036854775808 (-2^{63})$ à $9223372036854775807 (2^{63} - 1)$ . S'il est suivi de l'option UNSIGNED, les valeurs sont positives et varient de 0 à 18446744073709551615, soit $2^{64} - 1$ . Chaque valeur occupe 8 octets.
FLOAT	Nombre à virgule flottante en simple précision prenant des valeurs de $-3.402823466E+38$ à $-1.75494351E-38$ pour les nombres négatifs et de $1.75494351E-38$ à $3.402823466E+38$ pour les positifs. S'il est suivi de l'option UNSIGNED, les valeurs sont uniquement positives. Avec les options FLOAT (M,D), l'affichage s'effectue avec M chiffres dont D décimales. Chaque valeur occupe 4 octets.
DOUBLE	Nombre à virgule flottante en double précision prenant des valeurs de $-1.7976931348623157E+308$ à $-2.2250738585072014E-308$ pour les nombres négatifs et de $2.2250738585072014E-308$ à $1.7976931348623157E+308$ pour les positifs, auxquelles s'ajoute la valeur exacte de 0. S'il est suivi de l'option UNSIGNED, les valeurs sont uniquement positives. Avec les options DOUBLE (M,D) l'affichage se fait avec M chiffres dont D décimales. Chaque valeur occupe huit octets.
DECIMAL	Nombre à virgule flottante qui doit être signé. La valeur est stockée comme une chaîne de caractères dont chaque caractère est un chiffre. Les valeurs sont les mêmes que pour le type DOUBLE. S'il est suivi de l'option UNSIGNED, les valeurs sont uniquement positives. Avec les options DECIMAL (M), l'affichage s'effectue avec M chiffres (par défaut 10 chiffres), ce qui limite l'intervalle de valeurs. Avec les options DECIMAL (M,D), l'affichage s'effectue avec M chiffres dont D décimales. Chaque valeur occupe autant d'octets qu'il y a de caractères dans le nombre.

**Tableau 53 : Les types chaînes de caractères en MySQL :**

Type	Définition et caractéristiques
CHAR( <i>M</i> )	Chaîne de caractères de longueur fixe de <i>M</i> caractères complétée par des espaces si la donnée stockée est plus petite. Les espaces sont supprimées lors de la lecture. La longueur indiquée varie de 0 à 255 caractères. L'option CHAR( <i>M</i> ) BINARY rend la chaîne sensible à la casse lors des recherches. Une colonne de type CHAR(0) n'occupe qu'un octet et peut contenir les valeurs NULL et '' (chaîne vide), ce qui permet de simuler une valeur booléenne. La chaîne stockée occupe toujours <i>M</i> octets, même si elle ne contient qu'un seul caractère significatif.
VARCHAR( <i>M</i> )	Chaîne de caractères de longueur variable comprise entre 1 et <i>M</i> caractères. La valeur de <i>M</i> varie de 1 à 255 caractères. L'option VARCHAR( <i>M</i> ) BINARY rend la chaîne sensible à la casse lors des recherches. La chaîne stockée occupe <i>N</i> + 1 octets quand elle comprend <i>N</i> caractères.
TINYTEXT TINYBLOB	Texte d'une longueur comprise entre 1 et 255 caractères. Le type TINYBLOB est sensible à la casse. La chaîne stockée occupe <i>N</i> + 1 octets quand elle comprend <i>N</i> caractères.
TEXT BLOB	Texte d'une longueur comprise entre 1 et 65 535 caractères. Le type BLOB est sensible à la casse. La chaîne stockée occupe <i>N</i> + 2 octets quand elle comprend <i>N</i> caractères.
MEDIUMTEXT MEDIUMBLOB	Texte d'une longueur comprise entre 1 et 16 777 215 caractères. Le type MEDIUMBLOB est sensible à la casse. La chaîne stockée occupe <i>N</i> + 3 octets quand elle comprend <i>N</i> caractères.
LONGTEXT LONGBLOB	Texte d'une longueur comprise entre 1 et 4 294 967 295 caractères. Le type LONGBLOB est sensible à la casse. La chaîne stockée occupe <i>N</i> + 4 octets quand elle comprend <i>N</i> caractères.
ENUM('chaîne1', ..., 'chaîneN')	Permet le choix d'une seule valeur parmi l'énumération des <i>N</i> chaînes de caractères définies dans le type. La valeur NULL est toujours admise, même si elle ne figure pas dans la liste. Le type peut définir jusqu'à 65 535 valeurs. À chaque chaîne correspond une valeur numérique de 1 à 65 535, correspondant à son ordre d'apparition dans la définition du type. La valeur 0 correspond à une chaîne vide. La définition de ENUM('bleu', 'blanc', 'rouge') pour une colonne ne permet de stocker qu'une valeur parmi les trois couleurs de la liste ou la valeur NULL. Pour le HTML, ce type correspond à une liste de sélection <select> de N+1 <option> (les <i>N</i> valeurs proposées plus le choix NULL par défaut) à choix unique.
SET('chaîne1', ..., 'chaîneN')	Permet le choix de une ou plusieurs valeurs simultanément parmi l'ensemble des <i>N</i> chaînes de caractères définies dans le type. L'ensemble peut contenir jusqu'à 64 valeurs. À chaque choix correspond une valeur numérique entière égale à $2^{n-1}$ si <i>n</i> est la position de la chaîne dans l'ensemble (soit 1 pour la première, 2 pour la deuxième, 4 pour la troisième, etc.) ou encore une valeur binaire dans laquelle chaque bit est à 1 si la valeur est choisie (soit 0001 pour la première, 0010 pour la deuxième, 0100 pour la troisième, etc.). Si plusieurs valeurs sont choisies, la valeur numérique correspondante est la somme des valeurs de chacune (par exemple 5 pour la première et la troisième valeur). En HTML, ce type correspond à une liste de sélection <select> de N <option> à choix multiple (avec l'attribut multiple).



**Tableau 54 : Les types de dates et d'heures en MySQL :**

Type	Définition et caractéristiques
DATE	Une date au format AAAA-MM-JJ dans l'intervalle de 1000-01-01 à 9999-12-31. Chaque enregistrement occupe 3 octets.
DATETIME	Contient la date et l'heure au format AAAA-MM-JJ HH:MM:SS dans l'intervalle de 1000-01-01 00:00:00 à 9999-12-31 23:59:59. Chaque enregistrement occupe 8 octets.
TIME	Stocke l'heure au format HH:MM:SS ou HHH:MM:SS pour un intervalle de valeurs allant de -838:59:59 à 838:59:59 permettant d'effectuer des calculs de durée excédant 24 heures. Chaque enregistrement occupe 3 octets.
YEAR	Représente les années au format YYYY pour un intervalle allant de 1901 à 2155. Si l'année est fournie avec deux chiffres, les valeurs de 00 à 69 correspondent aux années 2000 à 2069 et les valeurs 70 à 99 aux années 1970 à 1999. Chaque enregistrement occupe 1 octet.

## Tableau 62 : Lecture de la table « article » de la base de données

### « Magasin »:

Script PHP
<pre>&lt;html&gt; &lt;head&gt; &lt;meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" /&gt; &lt;title&gt;Lecture de la table article&lt;/title&gt; &lt;style type="text/css" &gt; table {border-style:double;border-width:3px;border-color:red;background-color:yellow;} &lt;/style&gt; &lt;/head&gt; &lt;body&gt; &lt;?php include("connex.php"); \$idcom=connex("magasin","myparam"); \$requete="SELECT * FROM article ORDER BY categorie"; \$result=mysql_query(\$requete,\$idcom); if(!\$result) { echo "Lecture impossible"; } else { \$nbcol=mysql_num_fields(\$result); \$nbart=mysql_num_rows(\$result); echo "&lt;h3&gt; Tous nos articles par catégorie&lt;/h3&gt;"; echo "&lt;h4&gt; Il y a \$nbart articles en magasin &lt;/h4&gt;"; echo "&lt;table border='1'&gt;&lt;tbody&gt;"; echo "&lt;tr&gt;&lt;th&gt;Code article&lt;/th&gt; &lt;th&gt;Description&lt;/th&gt; &lt;th&gt;Prix&lt;/th&gt;&lt;th&gt;Catégorie&lt;/th&gt;&lt;/tr&gt;"; while(\$ligne=mysql_fetch_array(\$result,MYSQL_NUM)) { echo "&lt;tr&gt;"; foreach(\$ligne as \$valeur) { echo "&lt;td&gt; \$valeur &lt;/td&gt;"; } echo "&lt;/tr&gt;"; } echo "&lt;/tbody&gt;&lt;/table&gt;"; } mysql_free_result(\$result); ?&gt; &lt;/body&gt; &lt;/html&gt;</pre>
Résultat affiché dans le navigateur

Lecture de la table article - Microsoft Internet Explorer

Fichier Edition Affichage Favoris Outils ?

Précédente Recherche Favoris Média

Adresse http://127.0.0.1/plp5/C15mysql/mysulx2.php OK

**Tous nos articles par catégorie**

Il y a 11 articles en magasin

Code article	Description	Prix	Catégorie
CS330	Caméscope Sony DCR-PC330	1629.00	vidéo
CP100	Caméscope Panasonic SV-AV 100	1490.00	vidéo
NIK55	Nikon F55+zoom 28/80	269.00	photo
NIK80	Nikon F80	479.00	photo
CA300	Canon EOS 3000V zoom 28/80	329.00	photo
SOXMP	PC Portable Sony Z1-XMP	2399.00	informatique
HP497	PC Bureau HP497 écran TFT	1500.00	informatique
DEL30	Portable Dell X300	1715.00	informatique
SAX15	Portable Samsung X15 XVM	1999.00	informatique
DVD75	DVD vierge par 3	17.50	divers
CAS07	Cassette DV60 par 5	26.90	divers

Terminé Internet

**Tableau 63 : Exemple illustrant une requête SQL sélectionnant tous les articles de marque Sony dans la table Article en définissant des alias pour les noms de colonnes :**

Script PHP
<pre> &lt;html&gt; &lt;head&gt; &lt;meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" /&gt; &lt;title&gt;Lecture de la table article&lt;/title&gt; &lt;style type="text/css" &gt; table {border-style:double;border-width:3px;border-color:red;background-color:yellow;} &lt;/style&gt; &lt;/head&gt; &lt;body&gt; &lt;?php include("connex.php"); \$idcom=connex("magasin","myparam"); \$requete="SELECT id_article AS 'Code article',designation AS 'Désignation',prix AS 'Prix Unitaire',categorie AS 'Catégorie' FROM article WHERE designation LIKE '%Sony%' ORDER BY categorie"; \$result=mysql_query(\$requete,\$idcom); if(!\$result) { echo "Lecture impossible"; } else { \$nbart=mysql_num_rows(\$result); \$ligne=mysql_fetch_array(\$result,MYSQL_ASSOC); echo "&lt;h3&gt; Tous nos articles de la marque Sony&lt;/h3&gt;"; echo "&lt;h4&gt; Il y a \$nbart articles en magasin &lt;/h4&gt;"; echo "&lt;table border='1'&gt; &lt;tr&gt;"; foreach(\$ligne as \$nomcol=&gt;\$valcol) { echo "&lt;th&gt; \$nomcol &lt;/th&gt;"; } echo "&lt;/tr&gt;"; //***** do { echo "&lt;tr&gt;"; foreach(\$ligne as \$valcol) { echo "&lt;td&gt; \$valcol &lt;/td&gt;"; } echo "&lt;/tr&gt;"; } while(\$ligne=mysql_fetch_array(\$result,MYSQL_NUM)); </pre>

```
echo "</table>";  
}  
?>  
</body>  
</html>
```

### Résultat affiché dans le navigateur

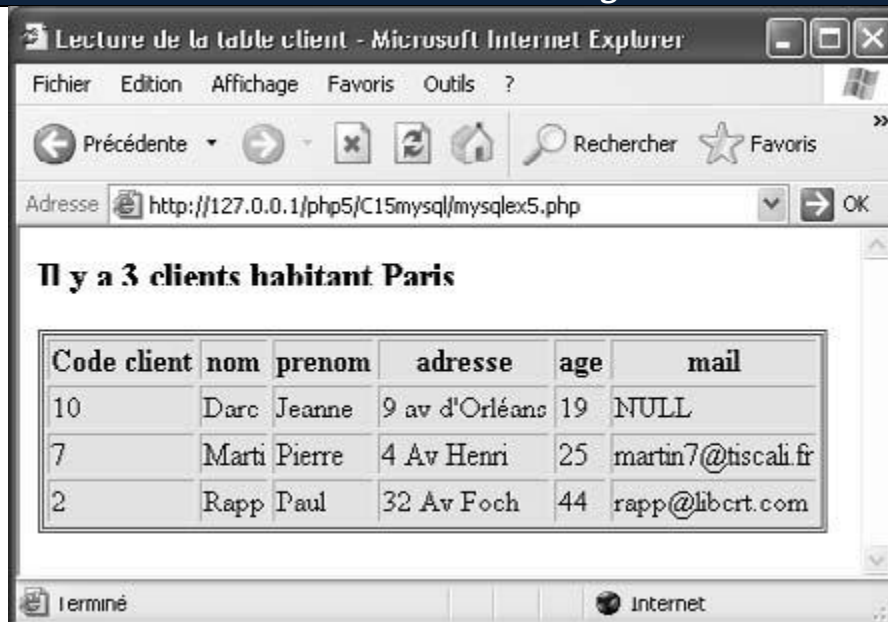


**Tableau 64 : Exemple illustrant une requête SQL sélectionnant les caractéristiques des clients habitant à Paris :**

Script PHP
<pre> &lt;html&gt; &lt;head&gt; &lt;meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" /&gt; &lt;title&gt;Lecture de la table client&lt;/title&gt; &lt;style type="text/css" &gt; table {border-style:double;border-width: 3px;border-color:red;background-color: yellow;} &lt;/style&gt; &lt;/head&gt; &lt;body&gt; &lt;?php include("connex.php"); \$idcom=connex("magasin","myparam"); \$requete="SELECT id_client AS 'Code client',nom,prenom,adresse,age FROM client WHERE ville ='Paris' ORDER BY nom"; \$result=mysql_query(\$requete,\$idcom); if(!\$result) { echo "Lecture impossible"; } else { \$nbcol=mysql_num_fields(\$result); \$nbart=mysql_num_rows(\$result); echo "&lt;h3&gt; Il y a \$nbart clients habitant Paris&lt;/h3&gt;"; //Affichage des titres du tableau echo "&lt;table border='1'&gt; &lt;tr&gt;"; for(\$i=0;\$i&lt;\$nbcol;\$i++) { echo "&lt;th&gt;", mysql_field_name(\$result,\$i)," &lt;/th&gt;"; } echo "&lt;/tr&gt;"; //Affichage des valeurs du tableau for(\$i=0;\$i&lt;\$nbart;\$i++) { \$ligne=mysql_fetch_row(\$result); echo "&lt;tr&gt;"; for(\$j=0;\$j&lt;\$nbcol;\$j++) { echo "&lt;td&gt;",\$ligne[\$j],"&lt;/td&gt;"; } echo "&lt;/tr&gt;"; } echo "&lt;/table&gt;"; mysql_free_result(\$result); } </pre>

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

### Résultat affiché dans le navigateur



**Tableau 65 : Exemple : Insertion à partir d'un formulaire permettant à un client d'enregistrer ses coordonnées lors d'une commande :**

Script PHP
<pre> &lt;html&gt; &lt;head&gt; &lt;meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" /&gt; &lt;title&gt;Saisissez vos coordonnées&lt;/title&gt; &lt;/head&gt; &lt;body&gt; &lt;form action= "&lt;?php echo \$_SERVER['PHP_SELF'];?&gt;" method="post" enctype="application/x-www-form-urlencoded"&gt; &lt;fieldset&gt; &lt;legend&gt;&lt;b&gt;Vos coordonnées&lt;/b&gt;&lt;/legend&gt; &lt;table&gt; &lt;tr&gt;&lt;td&gt;Nom : &lt;/td&gt;&lt;td&gt;&lt;input type="text" name="nom" size="40" maxlength="30"/&gt;&lt;/td&gt;&lt;/tr&gt; &lt;tr&gt;&lt;td&gt;Prénom : &lt;/td&gt;&lt;td&gt;&lt;input type="text" name="prenom" size="40" maxlength="30"/&gt;&lt;/td&gt;&lt;/tr&gt; &lt;tr&gt;&lt;td&gt;Age : &lt;/td&gt;&lt;td&gt;&lt;input type="text" name="age" size="40" maxlength="2"/&gt;&lt;/td&gt;&lt;/tr&gt; &lt;tr&gt;&lt;td&gt;Adresse : &lt;/td&gt;&lt;td&gt;&lt;input type="text" name="adresse" size="40" maxlength="60"/&gt;&lt;/td&gt;&lt;/tr&gt; &lt;tr&gt;&lt;td&gt;Ville : &lt;/td&gt;&lt;td&gt;&lt;input type="text" name="ville" size="40" maxlength="40"/&gt;&lt;/td&gt;&lt;/tr&gt; &lt;tr&gt;&lt;td&gt;Mail : &lt;/td&gt;&lt;td&gt;&lt;input type="text" name="mail" size="40" maxlength="50"/&gt;&lt;/td&gt;&lt;/tr&gt; &lt;tr&gt; &lt;td&gt;&lt;input type="reset" value=" Effacer "&gt;&lt;/td&gt; &lt;td&gt;&lt;input type="submit" value=" Envoyer "&gt;&lt;/td&gt; &lt;/tr&gt; &lt;/table&gt; &lt;/fieldset&gt; &lt;/form&gt; &lt;?php include('connex.php'); if(!empty(\$_POST['nom'])&amp;&amp; !empty(\$_POST['adresse'])&amp;&amp; !empty(\$_POST['ville'])) { \$id_client="\N"; /* La valeur "\N" représente la valeur NULL de la variable \$id_client. Elle permet que la colonne id_client de la table client soit incrémentée d'une unité à chaque nouvelle insertion car elle est déclarée avec l'option AUTO_INCREMENT. */ \$nom= \$_POST['nom']; \$prenom=\$_POST['prenom']; \$age=\$_POST['age']; \$adresse=\$_POST['adresse']; \$ville=\$_POST['ville']; \$mail=\$_POST['mail']; //Requête SQL \$requete="INSERT INTO client VALUES('\$id_client','\$nom','\$prenom','\$age','\$adresse','\$ville','\$mail')"; \$idcom=connex('magasin','myparam'); \$result=mysql_query(\$requete,\$idcom); mysql_close(\$idcom); </pre>



```

if(!$result)
{
echo "<h2>Erreur d'insertion \n n°",mysql_errno()," : ",mysql_error()."</h2>";
}
else
{
echo "<script type=\"text/javascript\">
alert('Vous êtes enregistré Votre numéro de client est : ".mysql_insert_id()."')</script>";
/* la fonction mysql_insert_id() retourne la dernière valeur insérée dans une colonne ayant
l'option AUTO_INCREMENT. Elle sert à donner son numéro au client pour qu'il puisse s'identifier
lors d'une autre connexion.*/
}
}
}
else {echo "Formulaire à compléter!";}
?>
</body>
</html>

```

### Résultat affiché dans le navigateur

Avant de cliquer sur le bouton « Envoyer » :

The screenshot shows a web browser window titled "Saisissez vos coordonnées - Microsoft Internet Explorer". The address bar displays "http://127.0.0.1/php5/C15mysql/mysqlx7.php". The form content is as follows:

Vos coordonnées	
Nom :	Mac Cartney
Prénom :	Paul
Age :	55
Adresse :	Abney Road
Ville :	London
Mail :	scarabe@paulmac.uk
<input type="button" value="Effacer"/> <input type="button" value="Envoyer"/>	

Après avoir cliqué sur le bouton « Envoyer » :

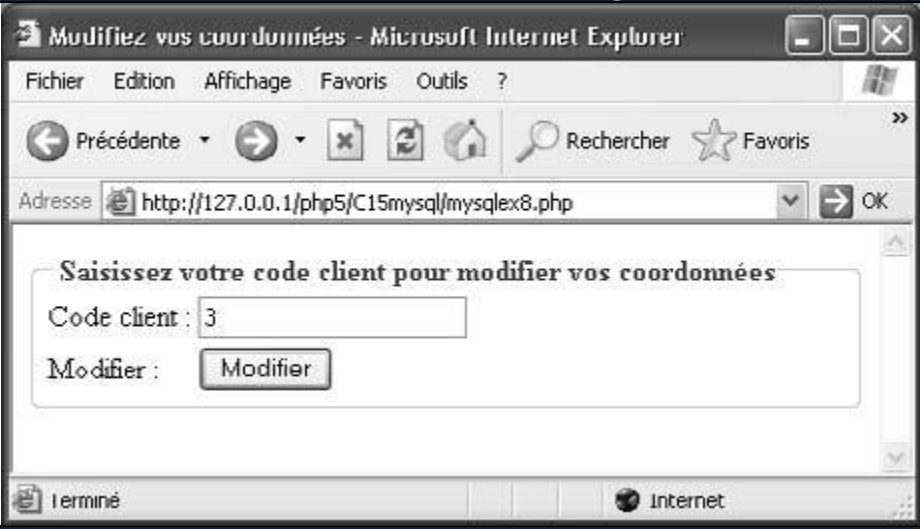
Microsoft Internet Explorer



vous êtes enregistré votre numéro de client est : 37

OK

**Tableau 66 : Exemple : modification des coordonnées d'un client déjà enregistré en cas de changement d'adresse ou de nom :**

Fichier : mysqlx8.php
<pre> &lt;html&gt; &lt;head&gt; &lt;meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" /&gt; &lt;title&gt;Modifiez vos coordonnées&lt;/title&gt; &lt;/head&gt; &lt;body&gt; &lt;form action="mysqlx9.php" method="post" enctype="application/x-www-form-urlencoded"&gt; &lt;fieldset&gt; &lt;legend&gt;&lt;b&gt;Saisissez votre code client pour modifier vos coordonnées&lt;/b&gt;&lt;/legend&gt; &lt;table&gt; &lt;tr&gt; &lt;td&gt;Code client : &lt;/td&gt; &lt;td&gt;&lt;input type="text" name="code" size="20" maxlength="10"/&gt; &lt;/td&gt;&lt;/tr&gt; &lt;tr&gt;&lt;td&gt;Modifier : &lt;/td&gt; &lt;td&gt;&lt;input type="submit" value="Modifier"/&gt;&lt;/td&gt;&lt;/tr&gt; &lt;/table&gt; &lt;/fieldset&gt; &lt;/form&gt; &lt;/body&gt; &lt;/html&gt; </pre>
Résultat affiché dans le navigateur

Fichier : mysqlx9.php
<pre> /* ceci doit figurer en tête du fichier car ce script PHP utilise la fonction header() pour effectuer la redirection */ &lt;?php if(empty(\$_POST['code'])) {header("Location:mysqlx8.php");} ?&gt; &lt;html&gt; &lt;head&gt; &lt;meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" /&gt; &lt;title&gt;Modifiez vos coordonnées&lt;/title&gt; &lt;/head&gt; &lt;body&gt; &lt;?php if(\$_POST['modif']!='Enregistrer') </pre>

```

{
//CREATION DU FORMULAIRE
include('connex.php');
$code=mysql_escape_string($_POST['code']); /* pour ajouter le caractère d'échappement pour
les caractères spéciaux des chaînes récupérées dans le tableau $_POST (N.B : inutile dans ce
cas)*/
//Requête SQL
$requete="SELECT * FROM client WHERE id_client='$code' ";
$idcom=connex('magasin','myparam'); /* une première connexion avec la base de données pour
lire les anciennes coordonnées du client */
$result=mysql_query($requete,$idcom);
$coord=mysql_fetch_row($result);
mysql_close($idcom);
//Création du formulaire
echo "<form action= \"\". $_SERVER['PHP_SELF'].\" \" method= \"post \" enctype= \"application/x-
www-form-urlencoded\">";
echo "<fieldset>";
echo "<legend><b>Modifiez vos coordonnées</b></legend>";
echo "<table>";
echo "<tr><td>Nom : </td><td><input type= \"text\" name= \"nom\" size= \"40\" maxlength= \"30\"
value= \"\$coord[1]\" /> </td></tr>"; /*coord[0] correspond à la valeur du champ id_client don't la
valeur est déjà contenu dans la variable $code */
echo "<tr><td>Prénom : </td><td><input type= \"text\" name= \"prenom\" size= \"40\"
maxlength= \"30\" value= \"\$coord[2]\" /></td></tr>";
echo "<tr><td>Age : </td><td><input type= \"text\" name= \"age\" size= \"40\" maxlength= \"2\"
value= \"\$coord[3]\" /></td></tr>";
echo "<tr><td>Adresse : </td><td><input type= \"text\" name= \"adresse\" size= \"40\"
maxlength= \"60\" value= \"\$coord[4]\" /></td></tr>";
echo "<tr><td>Ville : </td><td><input type= \"text\" name= \"ville\" size= \"40\" maxlength= \"40\"
value= \"\$coord[5]\" /></td></tr>";
echo "<tr><td>Mail : </td><td><input type= \"text\" name= \"mail\" size= \"40\" maxlength= \"50\"
value= \"\$coord[6]\" /></td></tr>";
echo "<tr><td><input type= \"reset\" value= \" Effacer \"></td><td><input type= \"submit\"
name= \"modif\" value= \"Enregistrer\"></td></tr></table>";
echo "</fieldset>";
echo "<input type= \"hidden\" name= \"code\" value= \"\$code\" />"; /* Le champ caché code du
formulaire permet de passer la valeur du code client à la partie du script chargée de
l'enregistrement des données modifiées */
echo "</form>";
}
elseif(isset($_POST['nom'])&& isset($_POST['adresse'])&&isset($_POST['ville']))
{
//ENREGISTREMENT
include('connex.php');
$nom=mysql_escape_string($_POST['nom']);
$adresse=mysql_escape_string($_POST['adresse']);
$ville=mysql_escape_string($_POST['ville']);
$mail=mysql_escape_string($_POST['mail']);
$code=mysql_escape_string($_POST['code']);
//Requête SQL
$requete="UPDATE client SET nom='$nom',adresse='$adresse',ville='$ville',mail='$mail'
WHERE id_client='$code'";

```

```

$idcom=connex('magasin','myparam'); /* une deuxième connexion avec la base de données pour
enregistrer les modifications apportées par le client */
$result=mysql_query($requete,$idcom);
mysql_close($idcom);
if(!$result)
{
echo "<script type='text/javascript'> alert('Erreur : ".mysql_error()."')</script>";
}
else
{
echo "<script type='text/javascript'> alert('Vos modifications sont enregistrées');
window.location='index.htm';</script>"; /* une fois l'opération de mise à jour se termine le
visiteur est redirigé vers la page d'accueil du site index.htm. */
}
}
else
{
echo "Modifier vos coordonnées!";
}
?>
</body>
</html>

```

#### Résultat affiché dans le navigateur :



**Tableau 67 : Exemple de recherche dans la base MySQL :**

Script PHP
<pre>&lt;html&gt; &lt;head&gt; &lt;meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" /&gt; &lt;title&gt;Rechercher un article dans le magasin&lt;/title&gt; &lt;/head&gt; &lt;body&gt; &lt;form action= "&lt;?php echo \$_SERVER['PHP_SELF'];&gt;" method="post" enctype="application/x-www-form-urlencoded"&gt; &lt;fieldset&gt; &lt;legend&gt;&lt;b&gt;Rechercher un article en magasin&lt;/b&gt;&lt;/legend&gt; &lt;table&gt; &lt;tr&gt; &lt;td&gt;Mot-clé: &lt;/td&gt; &lt;td&gt;&lt;input type="text" name="motcle" size="40" maxlength="40" value="&lt;?php \$_POST['motcle'];&gt;" /&gt;&lt;/td&gt; &lt;/tr&gt; &lt;tr&gt; &lt;td&gt;Dans la catégorie : &lt;/td&gt; &lt;td&gt; &lt;select name="categorie"&gt; &lt;option value="tous"&gt;Tous&lt;/option&gt; &lt;option value="vidéo"&gt;Vidéo&lt;/option&gt; &lt;option value="informatique"&gt;Informatique&lt;/option&gt; &lt;option value="photo"&gt;Photo&lt;/option&gt; &lt;option value="divers"&gt;Divers&lt;/option&gt; &lt;/select&gt; &lt;/td&gt; &lt;/tr&gt; &lt;tr&gt; &lt;td&gt;Trier par : &lt;/td&gt; &lt;td&gt; &lt;select name="tri"&gt; &lt;option value="prix"&gt;Prix&lt;/option&gt; &lt;option value="categorie"&gt;Catégorie&lt;/option&gt; &lt;option value="id_article"&gt;Code&lt;/option&gt; &lt;/select&gt; &lt;/td&gt; &lt;/tr&gt; &lt;tr&gt; &lt;td&gt;En ordre: &lt;/td&gt; &lt;td&gt;Croissant&lt;input type="radio" name="ordre" value="ASC" checked="checked" /&gt; Décroissant&lt;input type="radio" name="ordre" value="DESC" /&gt; &lt;/td&gt; &lt;/tr&gt; &lt;tr&gt;&lt;td&gt;Envoyer&lt;/td&gt;&lt;td&gt;&lt;input type="submit" name="" value="OK" /&gt; &lt;/td&gt; &lt;/tr&gt; &lt;/table&gt; &lt;/fieldset&gt; &lt;/form&gt; &lt;?php if(!empty(\$_POST['motcle'])) {</pre>

```

include('connex.php');
$motcle=mysql_escape_string($_POST['motcle']);
$categorie=mysql_escape_string($_POST['categorie']);
$ordre=mysql_escape_string($_POST['ordre']);
$tri=mysql_escape_string($_POST['tri']);
//Requête SQL
$reqcategorie=(($_POST['categorie']=="tous"))?"":"AND categorie='".$categorie."'"; /* Si la catégorie
choisie est "tous", la partie de la commande WHERE concernant cette catégorie est vide. Pour les
autres choix, elle est égale à "AND categorie=$categorie" */
$requete="SELECT id_article AS 'Code article',designation AS 'Description',prix,categorie AS
'Catégorie' FROM article WHERE lower(designation) LIKE'%$motcle%".$reqcategorie."ORDER BY
$tri $ordre";
$idcom=connex('magasin','myparam');
$result=mysql_query($requete,$idcom);
if(!$result)
{
echo "Lecture impossible";
}
else
{
$nbcol=mysql_num_fields($result);
$nbart=mysql_num_rows($result);
echo "<h3> Il y a $nbart articles correspondant au mot-clé : $motcle</h3>";
//Affichage des titres du tableau
echo "<table border='1'> <tr>";
for($i=0;$i<$nbcol;$i++)
{
echo "<th>", mysql_field_name($result,$i)," </th>";
}
echo "</tr>";
//Affichage des valeurs du tableau
for($i=0;$i<$nbart;$i++)
{
$ligne=mysql_fetch_row($result);
echo "<tr>";
for($j=0;$j<$nbcol;$j++)
{
echo "<td>",$ligne[$j],"</td>";
}
echo "</tr>";
}
echo "</table>";
mysql_free_result($result);
}
}
?>
</body>
</html>

```

Résultat affiché dans le navigateur

