

# PHP

## Sommaire :

<b>QU'EST CE QUE PHP ?</b> .....	2	<b>L'INSTRUCTION BREAK</b> .....	8
<b>COMMENTAIRES</b> .....	2	<b>L'INSTRUCTION DO... WHILE</b> .....	8
<b>LES INSTRUCTIONS</b> .....	2	<b>L'INSTRUCTION FOR</b> .....	8
<b>LES VARIABLES</b> .....	2	<b>L'INSTRUCTION EXIT</b> .....	8
<b>LES OPERATEURS</b> .....	3	<b>LES VARIABLES D'ENVIRONNEMENT</b> .....	8
<b>LES OPERATEURS ARITHMETIQUES</b> .....	3	<b>LES FONCTIONS INCLUDE ET REQUIRE</b> .....	8
<b>LES OPERATEURS LOGIQUES ET</b>		<b>LECTURE ET ECRITURE DE FICHIERS</b> .....	8
<b>RELATIONNELS</b> .....	3	<b>FONCTIONS SUR LES DOSSIERS</b> .....	9
<b>L'OPERATEUR DE CONCATENATION</b> .....	3	<b>FONCTIONS RELATIVES AU SYSTEME DE</b>	
<b>LES TABLEAUX</b> .....	3	<b>FICHIERS</b> .....	9
<b>LES TABLEAUX A UNE DIMENSION</b> .....	3	<b>FONCTIONS HTTP</b> .....	11
<b>INDEXATION AUTOMATIQUE</b> .....	3	<b>FONCTIONS DE CALENDRIER</b> .....	11
<b>INDEXATION DES TABLEAUX</b> .....	4	<b>FONCTIONS DE DATES ET HEURES</b> .....	12
<b>INITIALISATION DES TABLEAUX</b> .....	4	<b>IMAGES</b> .....	13
<b>LES TABLEAUX MULTIDIMENSIONNELS</b> .....	4	<b>OPTIONS PHP &amp; INFORMATIONS</b> .....	14
<b>LES FONCTIONS RELATIVES AUX</b>		<b>GESTION DES SESSIONS</b> .....	14
<b>TABLEAUX</b> .....	4	<b>FONCTIONS CONCERNANT LES VARIABLES</b>	
<b>ENVOYER DES DONNEES VERS LE</b>		<b>.....</b>	15
<b>NAVIGATEUR</b> .....	5	<b>FONCTIONS MATHEMATIQUES</b> .....	16
<b>RECUPERER DES DONNEES A PARTIR D'UN</b>		<b>FONCTIONS DIVERSES</b> .....	17
<b>FORMULAIRE</b> .....	6	<b>FONCTIONS MYSQL</b> .....	17
<b>EXPRESSIONS REGULIERES</b> .....	6	<b>FONCTIONS ODBC</b> .....	19
<b>LES INSTRUCTIONS DE CONTROLE</b> .....	7	<b>FONCTIONS DE RESEAU</b> .....	19
<b>L'INSTRUCTION IF</b> .....	7	<b>EXEMPLES DE SCRIPT</b> .....	20
<b>L'INSTRUCTION SWITCH</b> .....	7		
<b>L'INSTRUCTION WHILE</b> .....	7		

# Qu'est ce que PHP ?

PHP est un langage de script qui s'inclut dans le langage HTML et est traité par le serveur. La syntaxe du langage PHP provient du C. Grâce à des nombreuses extensions, PHP peut générer des fichiers PDF, s'interfacer avec des serveurs de messagerie, des serveurs LDAP ou encore générer des images et graphiques GIF à la volée

PHP peut s'interfacer à la quasi totalité des SGBD du marché, qu'ils soient commerciaux ou qu'ils viennent du monde du Free Software.

Pour signaler du code PHP dans une page on utilise les balises `<? Et ?>`

## Commentaires

Pour indiquer un commentaire on utilise

```
/* le texte du commentaire */
code PHP /* un autre commentaire */
code PHP // encore une autre forme de commentaire
```

## Les instructions

Les instructions doivent finir par le symbole `;`

## Les variables

Pour le langage PHP, il existe 3 types de données : les entiers, les nombres décimaux et les chaînes de texte.

Lorsque vous écrivez du code PHP, il est inutile de distinguer les différents types de variables, il suffit de leur assigner leur valeur.

Affectation d'une variable

```
$mavARIABLE = 50
```

ou

```
$autrevariable = "chaîne de texte"
```

Remarque : on peut aussi utiliser des guillemets simples

Une valeur ne peut être utilisée que dans la fonction courante, cependant, en utilisant la fonction `global` la portée de la variable est étendue à toute la page.

Exemple :

```
<?
function essai1()
{
    $variable="texte de essai1";
    echo $ variable ;
}
function essai2()
{
    $variable ="texte de essai2";
    echo $variable;
}
function essai3()
{
```

```

    global $variable;
    echo $variable;
}
$ variable ="texte de essai3";

Test1() ; // Retourne texte de essai1
Test2(); // Retourne texte de essai2
Test3(); // Retourne texte de essai3
?>

```

Les variables peuvent être définies comme constantes. Les constantes sont identiques aux variables mais elles ne peuvent être paramétrées qu'une seule fois.

## Les opérateurs

### Les opérateurs arithmétiques

Opérateur	Signification
+	Addition
-	Soustraction
*	Multiplication
/	Division
%	Modulo division
++	Incrémentatation
--	Décrémentatation

### Les opérateurs logiques et relationnels

Opérateur	Signification
<	Est inférieur à
>	Est supérieur à
<=	Est inférieur ou égal à
>=	Est supérieur ou égal à
==	Est égal à
!=	N'est pas égal à
AND &&	Et
OR	Ou
XOR	Ou exclusif
!	Sauf

### L'opérateur de concaténation

Pour concaténer deux chaînes, on utilise en PHP le point. Exemple : `expr1.expr2`

## Les tableaux

Les tableaux enregistrent plusieurs valeurs dans des listes. Vous pouvez accéder à chacun des éléments grâce à un index.

### Les tableaux à une dimension

```

<?
    $villes[0] = « Lille » ;
    $villes[1] = « Paris » ;
    $villes[2] = « Toulouse » ;
    echo « Je vis à $villes[0]<BR> » ;
?>

```

### Indexation automatique

```

<?
    $villes[] = « Lille » ;

```

```

$villes[] = « Paris » ;
$villes[] = « Toulouse » ;
for($i=0 ; $i<count($villes) ; $i++)
    echo "La ville $I correspond à $villes[$I]<BR>";

```

?>

### Indexation des tableaux

Celle-ci peut se faire grâce à des chiffres (confère ci-dessus) ou grâce à des chaînes.

<?

```

$UserInfo[« Nom »] = « J. Pierron » ;
$UserInfo[« Ville »] = « Lille » ;
$UserInfo[« Profession »] = « Webmaster » ;
for(
reset($UserInfo) ;
$key=key($UserInfo) ;
next($UserInfo))
{
$val = pos($UserInfo);
echo "$key correspond à $val.<BR>\n";
}

```

?>

### Initialisation des tableaux

Cette initialisation se fait grâce à la commande Array.

#### Exemples

<?

```

$day = Array(1 => "Lundi", "Mardi", "Mercredi", "Jeudi", "Vendredi",
"Samedi", "Dimanche");
echo "Jour 5 : $day[5]<BR>";

```

?>

<?

```

$day = Array(
    "Lun" => "Lundi",
    "Mar" => "Mardi",
    "Mer" => "Mercredi",
    "Jeu" => "Jeudi",
    "Ven" => "Vendredi",
    "Sam" => "Samedi",
    "Dim" => "Dimanche");
echo "Jour Dim :".$day["Lun"]. "<BR>";

```

?>

### Les tableaux multidimensionnels

#### Exemple

<?

```

$villes = Array(
    "Nord" => Array(
        "Lille",
        "Valenciennes",
        "Seclin"),
    "Ariège" => Array(
        "Foix",
        "Ganac")
    ) ;
echo $villes["Nord"][1];

```

?>

## Les fonctions relatives aux tableaux

array— Crée un tableau

array\_push — Empile un ou plusieurs éléments à la fin d'un tableau  
array\_pop — Dépile un élément de la fin d'un tableau  
array\_unshift — Empile un ou plusieurs éléments au début d'un tableau  
array\_shift— Dépile un élément au début d'un tableau  
array\_slice — Extrait une portion de tableau  
array\_splice — Efface une portion de tableau et la remplace  
array\_merge— Rassemble deux ou plusieurs tableaux en un seul  
array\_keys — Retourne les clés d'un tableau  
array\_values — Retourne les valeurs d'un tableau  
array\_walk— Exécute une fonction sur chacun des membres d'un tableau  
arsort— Trie un tableau en ordre inverse  
asort— Trie un tableau en ordre  
compact— Crée un tableau contenant les variables et leur valeur  
count— Compte le nombre d'élément d'un tableau  
current—Transforme une variable en tableau  
each— Retourne chaque paire clé/valeur d'un tableau  
end— Positionne le pointeur de tableau en fin de tableau  
extract— Importe des variables dans la table des symboles d'un tableau  
in\_array— Retourne vrai si une valeur appartient à un tableau  
key— Retourne la clé d'un tableau associatif  
ksort— Trie un tableau suivant les clés  
list— Transforme une liste de variables en tableau  
next— Avance le pointeur interne d'un tableau  
pos— Retourne l'élément courant d'un tableau  
prev — Recule le pointeur courant de tableau  
range— Crée un tableau contenant un intervalle d'entiers  
reset— Remet le pointeur interne de tableau au début  
rsort— Trie en ordre inverse  
shuffle— Mélange les éléments d'un tableau  
sizeof— Retourne le nombre d'éléments d'un tableau  
sort— Trie le tableau  
uasort— Trie un tableau en utilisant une fonction de comparaison définie par l'utilisateur  
uksort — Trie les clés d'un tableau en utilisant une fonction de comparaison définie par l'utilisateur  
usort — Trie les valeurs d'un tableau en utilisant une fonction de comparaison définie par l'utilisateur

### Exemple 1. Exemple array()

```

$fruits = array(
    "fruits" => array("a"=>"orange", "b"=>"banane", "c"=>"pomme"),
    "nombres" => array(1, 2, 3, 4, 5, 6),
    "trous" => array("premier", 5 => "deuxieme", "troisieme")
);
  
```

## Envoyer des données vers le navigateur

Trois fonction permettent d'envoyer du texte vers le navigateur

echo : envoie du texte sans formatage

print : envoie du texte sans formatage

printf : permet de spécifier un format de sortie pour éviter d'envoyer la valeur telle quelle.

#### Exemple

```
print("<TABLE BORDER='1'>\n ");
for ($i=1 ; $i<= 10 ; $i++)
{
    print("<TR>\n");
    for($j=1 ; $j<=10 ; $j++)
    {
        print("<TD>");
        print($i * $j);
        print("</TD>");
    }
    print("</TR>\n");
}
print("</TABLE>\n");
```

## Récupérer des données à partir d'un formulaire

Pour récupérer des données à partir d'un formulaire, il suffit d'ajouter le caractère "\$" devant le nom donné au contrôle.

Par exemple, si vous tapez : "<INPUT TYPE=TEXT NAME=nom>" comme code HTML vous récupérez la valeur de ce champ en écrivant dans le code PHP : "\$nom"

#### Exemple

```
switch($op)
{
    case « VIEW »:
        echo « Votre nom : $nom »;
        break;
    default:
        echo "<FORM METHOD=POST ACTION='?op=VIEW'>";
        echo "Votre nom : <INPUT TYPE=TEXT NAME=nom>";
        echo "<INPUT TYPE=SUBMIT VALUE='VALIDER'>";
        echo "</FORM>\n";
}
break;
```

## Expressions régulières

Les expressions régulières sont utilisées pour effectuer des manipulations complexes de chaînes de caractères. Les fonctions sont :

[ereg\(\)](#)

[ereg\\_replace\(\)](#)

[eregi\(\)](#)

[eregi\\_replace\(\)](#)

[split\(\)](#)

#### Exemple 1. Expressions régulières

```
ereg("abc", $string);
/* Retourne true si "abc"
   est trouvé quelque part dans la chaîne $string. */
ereg("^abc", $string);
/* Retourne true si "abc "
   est trouvé au début de la chaîne $string. */
ereg("abc$", $string);
/* Retourne true si "abc "
   est trouvé à la fin de la chaîne $string. */
eregi("(ozilla.[23]|MSIE.3)", $HTTP_USER_AGENT);
```

```

/* Retourne true si le navigateur client
   est Netscape 2, 3 ou MSIE 3. */
ereg("([[:alnum:]]+) ([[:alnum:]]+) ([[:alnum:]]+)",
     $string,$regs);
/* Introduit trois mots séparés par des espaces
   dans les chaînes $regs[1], $regs[2] et $regs[3]. */
$string = ereg_replace("^","<BR>",$string);
/* Insère une balise <BR> au début de la chaîne $string. */

$string = ereg_replace("$","<BR>",$string);
/* Insère une balise <BR> à la fin de la chaîne $string. */
$string = ereg_replace("\n","", $string);
/* Supprime toutes les nouvelles lignes de $string. */

```

L'exemple suivant prend une date au format ISO (YYYY-MM-DD) et l'affiche sous la forme DD.MM.YYYY :

#### Exemple 1. ereg()

```

if ( ereg( "[0-9]{4}-[0-9]{1,2}-[0-9]{1,2}", $date, $regs ) ) {
    echo "$regs[3].$regs[2].$regs[1]";
} else {
    echo "Format de date invalide : $date";
}

```

## Les instructions de contrôle

Les instructions de contrôles permettent d'exécuter des blocs de code sous certaines conditions. Cela permet d'avoir des scripts beaucoup plus simples et plus efficaces.

### L'instruction if

```

if (expr1)
{
    // bloc exécuté si expr1 est vraie (true)
}
else if (expr2)
{
    // bloc exécuté si expr1 est fausse (false) et si
    // expr2 est vraie (true)
}
else
{
    // bloc exécuté si expr1 et expr2 sont
// fausses (false)
}

```

### L'instruction switch

```

switch (expr1)
{
    case case-expr:
        break;
    default:
        break;
}

```

### L'instruction while

```

while (expr)
{
    Zéro ou plusieurs expressions
}

```

Cette boucle est très utile lorsque vous ne savez pas exactement combien de fois vous devrez exécuter le code (exemple : lecture d'un fichier, récupération d'enregistrements dans une base de données).

### L'instruction break

Cette instruction permet d'arrêter une boucle pendant son exécution.

```
while (true)
{
    echo "texte affiché" ;
    break;
}
```

### L'instruction do...while

Cette instruction fait la même chose que *while* excepté que les instructions sont exécutées au moins une fois.

```
do
{
    Zéro ou plusieurs expressions
} while (expr);
```

### L'instruction for

```
for (initialisation ; expression ; incrémentation)
{
    Zéro ou plusieurs expressions
}
```

Exemple

```
for ($i=0 ; $i<10 ; $i++)
{
    echo "$i";
}
```

### L'instruction exit

Contrairement à l'instruction *break*, *exit* permet d'interrompre l'exécution du code. Mais *exit* a la particularité de forcer l'arrêt de toutes les exécutions.

## Les variables d'environnement

PHP propose plusieurs variable d'environnement, créées automatiquement lorsque vous lancez une nouvelle procédure.

Exemple

```
print(« Vous utilisez : »);
print($HTTP_USER_AGENT);
print(« pour voir cette page.<BR>\n »);
```

Exemples d'autres variables disponibles :

```
HTTP_GET_VARS
HTTP_POST_VARS
HTTP_COOKIE_VARS
```

## Les fonctions include et require

Les fonctions *include* et *require* sont quasiment identiques.

Quand la fonction *require* est traitée, elle est remplacée par le fichier qu'elle désigne. De son côté, la fonction *include* fonctionne plus comme une branche.

## Lecture et écriture de fichiers

Les fichiers ouvrables peuvent être mentionnées par un chemin relatif ou un chemin absolu (exemple : `http://www.mciflash.com/fichier.txt`).

Exemple



```

    $file="fichier.txt";
// Ouvrir un fichier en écriture
    $myFile=fopen("$file","w");
    if (!$myFile) die("Erreur<BR>`$file` ne peut être créé\n ");
// Ecriture de quelques lignes
    fputs($myFile, "Ligne de texte\n");
    fputs($myFile, "autre Ligne de texte\n");
// Ferme le fichier
    fclose($myFile);

```

Mode d'accès aux fichiers :

"r" : read : ouverture en mode lecture

"w" : write : ouverture en mode écriture

"a" : append : ouverture en mode ajout

## Fonctions sur les dossiers

**chdir**— Change le dossier courant de PHP

**closedir**— Ferme le pointeur sur le dossier

**opendir**— Ouvre un dossier, et récupère un pointeur dessus

**readdir**— Lit une entrée du dossier

**rewinddir** -- Retourne à la première entrée du dossier

Exemple 1. Liste de tous les fichiers du dossier courant

```

<?
    $test=opendir('.');
    echo "dossier courant: $test\n";
    echo "Fichiers:\n";
    while ($file = readdir($test)) {
        echo "$file\n";
    }
    closedir($test);
?>

```

## Fonctions relatives au système de fichiers

Certaines fonctions dépendent du système (unix , windows) et peuvent être inopérantes.

**basename** \_ sépare le nom du fichier et le nom du répertoire

**chgrp** \_ change le groupe propriétaire du fichier

**chmod** \_ change le mode du fichier

**chown** \_ change le possesseur du fichier

**clearstatcache** \_ efface le cache de la fonction "stat"

**copy** \_ copie un fichier

**delete** \_ efface un fichier

**dirname** \_ renvoie le nom du dossier

**diskfree** \_ renvoie l'espace disque disponible dans le répertoire

**fclose** \_ ferme un fichier

**feof** \_ test pour savoir si le pointeur est à la fin du fichier

**fgetc** \_ renvoie le caractère sur lequel se trouve le pointeur du fichier

**fgetcsv** \_ renvoie la ligne courante sur lequel se trouve le pointeur du fichier et cherche dans le résultat les champs CSV

**fgets** \_ renvoie la ligne courante sur lequel se trouve le pointeur du fichier

**fgetss** \_ renvoie la ligne courante sur lequel se trouve le pointeur du fichier et élimine les tags HTML

file \_ lit le fichier et renvoie le résultat dans un tableau  
file\_exists \_ vérifie si un fichier existe  
fileatime \_ renvoie la date à laquelle on a accédé au fichier pour la dernière fois  
filectime \_ renvoie l'heure à laquelle on a accédé à l'inode pour la dernière fois  
filegroup \_ renvoie le group qui possède le fichier  
fileinode \_ renvoie le numéro d'inode du fichier  
filemtime \_ renvoie la date de dernière modification du fichier  
fileowner \_ renvoie le nom du possesseur du fichier  
fileperms \_ renvoie les permissions affectées au fichier  
filesize \_ renvoie la taille du fichier  
filetype \_ renvoie le type du fichier  
flock \_ verrouille le fichier  
fopen \_ ouverture d'un fichier ou d'une URL  
fpasssthru \_ affiche la partie du fichier situé après le pointeur du fichier  
fputs \_ écrit dans le fichier vers lequel le pointeur est dirigé  
fread \_ lecture du fichier en mode binaire.  
fseek \_ modifie le pointeur de fichier  
ftell \_ renvoie la position du pointeur du fichier  
fwrite \_ écriture du fichier en mode binaire.  
set\_file\_buffer \_ Mets en place un buffer sur le pointeur de fichier courant  
is\_dir \_ indique si le nom de fichier est un dossier  
is\_executable \_ indique si le fichier est exécutable  
is\_file \_ indique si le fichier est un fichier  
is\_link \_ indique si le fichier est un lien symbolique  
is\_readable \_ indique un fichier est autorisé en lecture  
is\_writable \_ indique un fichier est autorisé en écriture  
link \_ crée un lien "hard"  
linkinfo \_ renvoie les informations à propos d'un lien  
mkdir \_ crée un dossier  
pclose \_ ferme un processus de pointeur de fichier  
popen \_ crée un processus de pointeur de fichier  
readfile \_ affiche un fichier  
readlink \_ renvoie le nom du fichier vers lequel pointe un lien symbolique  
rename \_ renomme un fichier  
rewind \_ Remplace le pointeur de fichier au début.  
rmdir \_ efface un dossier  
stat \_ renvoie les informations à propos d'un fichier  
lstat \_ renvoie les informations à propos d'un fichier ou d'un lien symbolique  
symlink \_ crée un lien symbolique  
tempnam \_ crée un fichier  
touch \_ affecte une nouvelle date de modification du fichier  
umask \_ change le "umask" courant  
unlink \_ efface le fichier

Sous windows, les caractères (/) et backslash (\) sont utilisés comme séparateur de répertoire. Sous les autres OS, seul le caractère slash (/) est utilisé.

Exemple 1. exemple d'utilisation de la fonction basename()

```
$path = "/home/httpd/html/index.PHP";
$file = basename($path); // $file est égale à "index.PHP"
```

### Exemple 2. fopen()

```
$fp = fopen("/home/webmaster/file.txt", "r");
$fp = fopen("http://www.free.fr/", "r");
$fp = fopen("ftp://user:password@example.com/", "w");
```

## Fonctions HTTP

Ces fonctions permettent de travailler sur les informations transmises au navigateur, via le protocole HTTP.

**header** -- Envoie un entête HTTP

**setcookie** -- Envoie un cookie

### Exemple

```
header("Location: http://www.php.net");
/* Redirige un navigateur vers le site web de PHP */
exit;
/* On s'assure que le code ci-après n'est pas exécuté lors de la
redirection */
```

### Exemple

```
setcookie("monCookie", "Test Value");
setcookie("monCookie", $value, time()+3600); /* expire dans 1 heure */
```

**NB** : la valeur du cookie sera automatiquement encodée lors de l'envoi, et décodé lors de la réception. Pour afficher le contenu du test précédent dans un script, utilisez simplement :

```
echo $monCookie;
echo $HTTP_COOKIE_VARS["monCookie"];
```

## Fonctions de calendrier

JDToGregorian — Convertit le nombre de jours du calendrier julien en date grégorienne.

GregorianToJD — Convertit une date grégorienne en nombre de jours du calendrier julien

JDToJulian — Convertit le nombre de jours du calendrier julien en date du calendrier julien

JulianToJD — Convertit une date du calendrier julien en nombre de jours du calendrier julien

JDToJewish — Convertit le nombre de jours du calendrier julien en date du calendrier juif

JewishToJD — Convertit une date du calendrier juif en nombre de jours du calendrier julien

JDToFrench — Convertit le nombre de jours du calendrier julien en date du calendrier

français républicain

FrenchToJD — Convertit une date du calendrier français républicain en nombre de jours du calendrier julien

JDMonthName — Retourne le nom du mois

JDDayOfWeek — Retourne le numéro du jour de la semaine

easter\_date — Retourne un timestamp UNIX pour Pâques, à minuit, pour une année donnée

easter\_days — Retourne le nombre de jours entre le 21 Mars et Pâques, pour une année donnée.

### Exemple 1. Fonction de calendrier

```
<?
$jd = GregorianToJD(10,11,1970);
echo("$jd\n");
$gregorian = JDToGregorian($jd);
echo("$gregorian\n");
?>
```

### Exemple 2. easter\_date() exemple

```

echo date( "M-d-Y", easter_date(1999) );          /* "04 avril 1999" */
echo date( "M-d-Y", easter_date(2000) );          /* "23 avril 2000" */
echo date( "M-d-Y", easter_date(2001) );          /* "15 avril 2001" */
Exemple 3. easter_date() exemple
echo easter_days(1999);                          /* 14, i.e. Avril 4 */
echo easter_days(1492);                          /* 32, i.e. Avril 22 */
echo easter_days(1913);                          /* 2, i.e. Mars 23 */

```

## Fonctions de dates et heures

`checkdate` -- Valide une date/heure

`date` -- Formate une date/heure locale

Les caractères suivants sont utilisés pour spécifier le format :

a - "am" ou "pm"

A - "AM" ou "PM"

d - Jour du mois, sur deux chiffres (éventuellement avec un zéro) : "01" à "31"

D - Jour de la semaine, en trois lettres (et en anglais) : par exemple "Fri" (pour Vendredi)

F - Mois, textuel, version longue; en anglais, i.e. "January" (pour Janvier)

h - Heure, au format 12h, "01" à "12"

H - heure, au format 24h, "00" à "23"

g - Heure, au format 12h sans les zéros initiaux, "1" à "12"

G - Heure, au format 24h sans les zéros initiaux, "0" à "23"

i - Minutes; "00" à "59"

j - Jour du mois sans les zéros initiaux: "1" à "31"

l - Jour de la semaine, textuel, version longue; en anglais, i.e. "Friday" (pour Vendredi)

L - Booléen pour savoir si l'année est bisextile ("1") ou pas ("0")

m - Mois; i.e. "01" à "12"

n - Mois sans les zéros initiaux; i.e. "1" à "12"

M - Mois, en trois lettres (et en anglais) : par exemple "Jan" (pour Janvier)

s - Secondes; i.e. "00" à "59"

S - Suffixe ordinal d'un nombre, en anglais, sur deux lettres : i.e. "th", "nd"

t - Nombre de jour dans le mois donnée, i.e. "28" à "31"

U - Secondes depuis une époque

w - Jour de la semaine, numérique, i.e. "0" (Dimanche) to "6" (Samedi)

Y - Année, 4 chiffres; i.e. "1999"

y - Année, 2 chiffres; i.e. "99"

z - Jour de l'année; i.e. "0" à "365"

Z - Décalage horaire en secondes (i.e. "-43200" à "43200")

Exemple 1. `date()` exemple

```

print (date("l dS of F Y h:i:s A"));
print ("July 1, 2000 is on a " . date("l", mktime(0,0,0,7,1,2000)));

```

`Strftime`- Formate une date/heure locale avec les options locales

`gmstrftime` -- Formate une date/heure GMT/CUT

`getdate` -- Retourne la date/heure

`gettimeofday` -- Retourne l'heure actuelle

`gmtime` -- Formate une date/heure GMT/CUT

`time` -- Retourne le timestamp UNIX actuel

`microtime` -- Retourne le timestamp UNIX actuel avec microsecondes

# Images

Vous pouvez utiliser les fonctions PHP pour obtenir les tailles des images aux formats JPEG, GIF, et PNG.

getImageSize — retourne la taille d'une image GIF, JPG ou PNG

ImageArc— dessine une ellipse partielle

ImageChar— dessine un caractère horizontalement

ImageCharChapter— dessine un caractère verticalement

ImageColorAllocate— alloue une couleur pour une image

ImageColorTransparent — définit la couleur transparente

ImageCopyResized — copie et redimensionne une partie d'une image

ImageCreate — crée une nouvelle image

ImageCreateFromGif — crée une nouvelle image à partir d'un fichier ou d'une URL

ImageDashedLine — dessine une ligne pointillée

ImageDestroy — détruit une image

ImageFill — remplit

ImageFilledPolygon — dessine un polygone plein

ImageFilledRectangle— dessine un rectangle plein

ImageFillToBorder— remplir avec une région avec une couleur spécifique

ImageFontHeight — retourne la hauteur de la police

ImageFontWidth— retourne la largeur de la police

ImageGif— envoie une image vers un navigateur ou un fichier

ImageInterlace — active ou désactive l'entrelacement

ImageLine— dessine une ligne

ImageLoadFont— charge une nouvelle police

ImagePolygon— dessine un polygone

ImageRectangle — dessine un rectangle

ImageSetPixel— dessine un pixel

ImageString— dessine une chaîne horizontale

ImageStringChapter — dessine une chaîne ligne verticale

ImageSX — retourne la largeur d'une image

ImageSY— retourne la hauteur d'une image

ImageTTFBBox— retourne le rectangle entourant un texte et dessiné avec une police TrueType

ImageTTFText — dessine un texte avec une police TrueType

ImageColorAt— retourne l'index de la couleur d'un pixel donné

ImageColorClosest— retourne l'index de la couleur la plus proche d'une couleur donnée

ImageColorExact— retourne l'index de la couleur donnée

ImageColorResolve— retourne l'index de la couleur donnée, ou la plus proche possible

ImageColorSet— change la couleur dans une palette à l'index donné

ImageColorsForIndex— retourne la couleur associée à un index

ImageColorsTotal— calcule le nombre de couleur d'une palette

ImagePSLoadFont— charge une police PostScript Type 1 depuis un fichier

ImagePSFreeFont — libère la mémoire occupée par une police PostScript Type 1

ImagePSEncodeFont— change le codage vectoriel d'un caractère dans une police

ImagePSText — dessine un texte sur une image avec une police PostScript Type1

ImagePSBBox — retourne le rectangle entourant un texte et dessiné avec une police PostScript Type1

### Exemple 1. GetImageSize

```
<? $size = GetImageSize("img/image.jpg"); ?>
<IMG SRC="img/image.jpg" <? echo $size[3]; ?>>
```

## Options PHP & informations

error\_log — envoie un message d'erreur quelque part.

error\_reporting — établit le niveau d'erreur à prendre en compte.

getenv — retourne la valeur de la variable d'environnement.

get\_cfg\_var — retourne la valeur de l'option de configuration du PHP.

get\_current\_user — retourne le nom de l'utilisateur qui a lancé le script.

get\_magic\_quotes\_gpc — retourne la configuration actuel de l'option "magic\_quotes\_gpc".

get\_magic\_quotes\_runtime — retourne la configuration actuelle de l'option magic\_quotes\_runtime.

getlastmod — retourne la date de dernière modification de la page.

getmyinode — retourne l'inode du script.

getmypid — retourne le numéro de processus actuel.

getmyuid — retourne l'UID du propriétaire du script actuel.

getrusage — retourne le niveau d'utilisation des ressources.

phpinfo — Affiche de nombreuses information relatives au PHP.

phpversion — retourne le numéro de version courante du PHP.

extension\_loaded — détermine si une extension est chargée ou non.

putenv — établit la valeur d'une variable d'environnement.

set\_magic\_quotes\_runtime — établit la configuration de l'option "magic\_quotes\_runtime".

set\_time\_limit — limite le temps maximum d'exécution d'un script.

### Exemple 1. exemple de la fonction phpversion()

```
// affiche le numéro de version courante du PHP.
echo "Version courante du PHP: ".phpversion();
```

## Gestion des sessions

La gestion des sessions avec PHP est un moyen de sauver des informations entre deux accès. Cela permet notamment de construire des applications personnalisées, et d'accroître l'attrait de votre site.

Chaque visiteur qui accède à votre site se voit assigné un numéro d'identifiant. Celui ci est enregistré soit dans un cookie, chez le client, soit dans l'URL.

Il y a deux mode de propagation de l'identifiant de session :

- Cookies
- URL

### Exemple 1. Compter le nombre de hit d'un utilisateur.

```
<?php
session_register("compteur");
$compteur++;
?>
Salut visiteur, vous avez vu cette page <? echo $compteur; ?> fois.<p>
<?
# le <?=%SID?> est nécessaire pour transmettre l'identifiant de session
# au cas ou les utilisateurs auront inactivé les cookies
?>
Pour continuer, , <A HREF="nextpage.php?<?=%SID?>">clique ici </A>
```

### Exemple 2. Utilisation de session\_set\_save\_handler()

```

<?php
function open ($save_path, $session_name) {
    echo "open ($save_path, $session_name)\n";
    return true;
}
function close () {
    echo "close\n";
    return true;
}
function read ($key) {
    echo "write ($key, $val)\n";
    return "foo|i:1;";
}
function write ($key, $val) {
    echo "write ($key, $val)\n";
    return true;
}
function destroy ($key)
    return true;
}
function gc ($maxlifetime) {
    return true;
}
session_set_save_handler ("open", "close", "read", "write", "destroy",
"gc");
session_start ();
$foo++;
?>

```

session\_start \_ Initialise les données de session

session\_destroy \_ Détruit toutes les données enregistrées, pour la session courante.

session\_name \_ Affecte et/ou retourne le nom de la session courante.

session\_module\_name \_ Affecte et/ou retourne le module de la session courante.

session\_save\_path \_ Affecte et/ou retourne le chemin de sauvegarde de la session courante.

session\_id \_ Affecte et/ou retourne l'identifiant de session courante.

session\_register \_ Enregistre une variable dans la session courante.

session\_unregister \_ Supprime une variable dans la session courante

session\_is\_registered \_ Indique si une variable a été enregistrée dans la session ou pas.

session\_decode \_ Décode les données de session à partir d'une chaîne.

session\_encode \_ Encode les données de session dans une chaîne

Exemple 1. session\_name()

```

$username="foo";
if(isset($username)) {
    session_name($username);
}
echo "Your username is " . session_name();

```

## Fonctions concernant les variables

gettype — Retourne le type de la variable.

intval — Retourne la valeur numérique de la variable. (Retourne une valeur de type integer)

doubleval — Retourne la valeur numérique de la variable (Retourne une valeur de type double).

empty — Détermine si une variable est affectée.

`is_array` — Détermine si une variable est un tableau.  
`is_double` — Détermine si une variable est de type double.  
`is_float` — Détermine si une variable est de type double.  
`is_int` — Détermine si une variable est de type integer.  
`is_integer` — Détermine si une variable est de type integer.  
`is_long` — Détermine si une variable est de type integer.  
`is_object` — Détermine si une variable est de type object.  
`is_real` — Détermine si une variable est de type real.  
`is_string` — Détermine si une variable est de type string.  
`isset` — Détermine si une variable est affectée.  
`settype` — Affecte un type à une variable.  
`strval` — Retourne la valeur de la variable (Renvoie une chaîne de caractères).  
`unset` — Désaffecte une variable.

## Fonctions mathématiques

`Abs` — valeur absolue  
`Acos` — arc cosinus  
`Asin` — arc sinus  
`Atan` — arc tangent  
`Atan2` — arc tangent de deux variables  
`base_convert` — converti un nombre entre deux bases arbitraires  
`BinDec` — converti de binaire en décimal  
`Ceil` — arrondi à l'entier supérieur  
`Cos` — cosinus  
`DecBin` — converti de décimal en binaire  
`DecHex` — converti de décimal en hexadécimal  
`DecOct` — converti de décimal en octal  
`Exp` — e à la puissance de (exponentielle de)  
`Floor` — arrondi à l'entier inférieur  
`getrandmax` — la plus grande valeur aléatoire possible  
`HexDec` — converti de hexadécimal en décimal  
`Log` — logarithme naturel  
`Log10` — logarithme de base 10  
`max` — la plus grande valeur  
`min` — la plus petite valeur  
`mt_rand` — génère une meilleure valeur aléatoire  
`mt_srand` — initialise une meilleure valeur aléatoire  
`mt_getrandmax` — la plus grande valeur aléatoire possible  
`number_format` — formate un nombre par groupe de millier  
`OctDec` — converti de octal en décimal  
`pi` — retourne la valeur de pi  
`pow` — calcul de puissance  
`rand` — génère une valeur aléatoire  
`round` — arrondi  
`Sin` — sinus  
`Sqrt` — racine carrée  
`srand` — sème le générateur de nombre aléatoire



Tan — tangent

## Fonctions diverses

connection\_aborted — Retourne vrai si le client a abandonné la connexion

connection\_status — Retourne les bits de status de la connexion

connection\_timeout — Retourne true si le script a dépassé les délais (timed out)

eval — Evalue une chaîne comme un script PHP

die — Affiche un message et termine le script courant

exit — Termine le script courant

function\_exists — Retourne true si la fonction a été définie

ignore\_user\_abort — Met l'option que indique que la déconnexion du client entraîne la fin du script

leak — Fuite de mémoire

pack — Conditionne des données dans une chaîne binaire

register\_shutdown\_function — Enregistre une fonction pour qu'elle s'exécute à l'extinction

serialize — génère une représentation enregistrable d'une valeur

sleep — Retarde l'exécution

unpack — Déconditionne des données depuis une chaîne binaire

unserialize — lit une représentation enregistrée d'une valeur

uniqid — génère un ID unique

usleep — Retarde l'exécution en microsecondes

Exemple 1 eval()- concaténation de texte

```
<?
$string = 'tasse';
$name = 'cafe';
$str = 'Ceci est une $string avec mon $name dedans.<br>';
echo $str;
eval( "\$str = \"\$str\";" );
echo $str;
?>
```

Exemple 2 die

```
<?
$filename = '/path/to/data-file';
$file = fopen($filename, 'r') or die "impossible d'ouvrir le fichier
($filename)";
?>
```

## Fonctions MySQL

Ces fonctions vous permettent d'accéder aux bases de données MySQL.

mysql\_affected\_rows — Retourne le nombre de ligne affectées par la dernière opération

mysql\_close — Ferme une connexion MySQL

mysql\_connect — Ouvre une connexion MySQL

mysql\_create\_db — Créer une base de données MySQL

mysql\_data\_seek — Déplace le pointeur de résultat interne

mysql\_db\_query — Envoie une requête à un serveur MySQL.

mysql\_drop\_db — Efface une base de données MySQL

mysql\_errno — Retourne le numéro d'erreur de la dernière opération

mysql\_error — Retourne le texte de l'erreur de la dernière opération

mysql\_fetch\_array — Retourne un résultat sous la forme d'un tableau associatif.

mysql\_fetch\_field — Retourne les informations concernant les colonne d'un résultat sous la forme d'un objet

mysql\_fetch\_lengths — Retourne la taille de chaque colonne d'un résultat

mysql\_fetch\_object — Retourne une ligne de résultat sous la forme d'un objet

mysql\_fetch\_row — Retourne une ligne de résultat sous la forme d'un tableau associatif

mysql\_field\_name — Retourne le nom d'un champs à partir de son index

mysql\_field\_seek — Affecte la position dy pointeur de résultat

mysql\_field\_table — Retourne de la talbe qui contient le champs spécifié

mysql\_field\_type — Retourne le type du champs spécifié

mysql\_field\_flags — Retourne les flags associés à un champs dans un résultat

mysql\_field\_len — Retourne la taille du champs spécifié

mysql\_free\_result — Libère la mémoire

mysql\_insert\_id — Retourne l'identifiant généré par la dernière requête INSERT

mysql\_list\_fields — Liste les champs du résultat MySQL

mysql\_list\_dbs — Liste les bases de données disponibles sur le serveur MySQL

mysql\_list\_tables — Liste les tables d'une base de données

mysql\_num\_fields — Retourne le nombre de champs d'un résultat

mysql\_num\_rows — Retourne le nombre de ligne d'une résultat

mysql\_pconnect — Ouvre une connexion persistante à un serveur MySQL

mysql\_query — Envoie une requête SQL à un serveur MySQL

mysql\_result — Retourne un champs d'un résultat

mysql\_select\_db — Selectionne une base dedonnées MySQL

mysql\_tablename — Retourne le nom de la table qui contient le champs spécifié

Example 1. mysql fetch object

```
<?
mysql_connect($host,$user,$password);
$result = mysql_db_query("base","select * from table");
while($row = mysql_fetch_object($result)) {
    echo $row->user_id;
    echo $row->fullname;
}
mysql_free_result($result);
?>
```

Example 2. mysql field types

```
<?
mysql_connect("localhost:3306");
mysql_select_db("wisconsin");
$result = mysql_query("SELECT * FROM onek");
$fields = mysql_num_fields($result);
$rows   = mysql_num_rows($result);
$i = 0;
$table = mysql_field_table($result, $i);
echo "Your '". $table. "' table has ". $fields. " fields and ". $rows. "
records <BR>";
echo "The table has the following fields <BR>";
while ($i < $fields) {
    $type = mysql_field_type ($result, $i);
    $name = mysql_field_name ($result, $i);
    $len  = mysql_field_len ($result, $i);
    $flags = mysql_field_flags ($result, $i);
    echo $type." ". $name." ". $len." ". $flags." <BR>";
    $i++;
}
```

```

}
mysql_close();
?>
Exemple 1. exemple mysql_tablename()
<?
mysql_connect ("localhost:3306");
$result = mysql_listtables ("france");
$i = 0;
while ($i < mysql_num_rows ($result)) {
    $tb_names[$i] = mysql_tablename ($result, $i);
    echo $tb_names[$i] . "<BR>";
    $i++;
}
?>

```

## Fonctions ODBC

odbc\_autocommit \_ Mode auto validation

odbc\_binmode \_ Gestion des colonnes de données binaires

odbc\_close \_ Ferme une connexion ODBC

odbc\_close\_all \_ Ferme toutes les connexions ODBC

odbc\_commit \_ Valide une transaction ODBC

odbc\_connect \_ Connexion à une source

odbc\_cursor \_ Lecture du pointeur de fiche courante (cursorname)

odbc\_do \_ synonyme pour odbc\_exec()

odbc\_exec \_ Prépare et exécute une requête SQL

odbc\_execute \_ Exécute une requête SQL préparée

odbc\_fetch\_into \_ Lit une ligne de résultat, et la place dans un tableau.

odbc\_fetch\_row \_ Lit une ligne de résultat

odbc\_field\_name \_ Lit le nom de la colonne

odbc\_field\_type \_ Type de données d'un champs

odbc\_field\_len \_ Lit la longueur d'un champs

odbc\_free\_result \_ Libère les ressources associées à un résultat

odbc\_longreadlen \_ Gestion des colonnes de type LONG

odbc\_num\_fields \_ Nombre de colonne dans un résultat

odbc\_pconnect \_ Ouvre une connexion persistante à une source de données

odbc\_prepare \_ Prépare une commande pour l'exécution

odbc\_num\_rows \_ Nombre de ligne dans un résultat

odbc\_result \_ Lit les données de résultat

odbc\_result\_all \_ Affiche le résultat sous la forme d'une table HTML

odbc\_rollback \_ Annule toutes les transactions

odbc\_setoption \_ Modifie les paramètres ODBC.

## Fonctions de réseau

fsockopen — Ouvre une socket de connexion Internet ou Unix.

pfsockopen — Ouvre une socket de connexion Internet ou Unix persistante.

set\_socket\_blocking — Met la socket en mode bloquant ou non bloquant

gethostbyaddr — Retourne le nom d'hôte correspondant à une IP

gethostbyname — Retourne l' IP correspondant à un nom d'hôte.

gethostbyname1 — Retourne la liste des IP correspondant à un nom d'hôte.

checkdnsrr — Résolution DNS d'une adresse IP.

getmxrr — Retourne les enregistrements MX d'un hôte donné.  
openlog — Ouvre la connexion à l'historique système  
syslog — Génère un message dans l'historique système  
closelog — Ferme la connexion à l'historique système  
debugger\_on — Active le debugger interne PHP  
debugger\_off — Desactive le debugger interne PHP

## Exemples de script

```
<?
echo "mon premier script PHP";
/* avec un commentaire sur plusieurs ligne
suite du commentaire */
?>
```

```
-----
<?
echo "type de données", "<br>";
$var1=NULL;
$var2=" ";
$var3="texte";
$var4=100;
define ("cst", "valeur constante");
echo gettype ($var1), "<br>";
echo gettype ($var2), "<br>";
echo gettype ($var3), "<br>";
echo gettype ($var4), "<br>";
echo (cst);
?>
```

```
-----
<?
echo "variable locale et globale", "<br>";
$var=100;
function test1()
{
return $var;
}
$result= test1()+100;
echo "$result";
?>
```

```
-----
<?
echo "variable locale et globale", "<br>";
$var=100;
function test1()
{
global $var
return $var;
}
$result= test1()+100;
echo "$result";
?>
```

```
-----
<?
echo "variable statique", "<br>";
function test1()
{
```

```

$var=100;
++$var;
return $var;
}
$result= test1();
$result1= test1();
echo "$result", "<br>";
echo "$result1";
?>
-----
<?
echo "variable statique", "<br>";
function test1()
{
static $var=100;
++$var;
return $var;
}
$result= test1();
$result1= test1();
echo "$result", "<br>";
echo "$result1";
?>
-----
<?
echo "test logique", "<br>";
$var1 = 100;
$var2 = 100;
if ($var1< $var2)
{
    echo " les 2 variables sont identiques";
}
else
{
    echo "<H1>","les 2 variables sont différentes","</H1>";
}
?>
-----
<?
echo "test logique", "<br>";
$var1 = 100;
$var2 = 110;
if ($var1> $var2)
{
    echo " var1 n'est pas > à var2";
}
else if ($var1<$var2)
{
    echo " var1 n'est pas < à var2";
}
else
{
    echo " var1 est = à var2";
}
// mettre $var1 pour voir la valeur de la variable
?>
-----
<?

```

```

echo "choix alternatif", "<br>";
$var1 = 2;
Switch ($var1)
{
case 0:
    echo " si var est 0";
    break;
case 1:
    echo " si var est 1";
    break;
case 2:
    echo " si var est 2";
    break;
case 3:
    echo " si var est 3";
    break;
}
// à faire sans les break pour voir ce qui ce passe
?>
-----
<?
echo "travail sur les chaines de texte", "<br>";
$chain1="bonjour vous";
$chain2="Bonjour vous";
$test= strcmp($chain1,$chain2);
if ($test==0)
{
echo "les 2 chaines sont identiques";
}
else
{
echo "les 2 chaines sont différentes";
}
//avec la fonction strcmp php est sensible à la casse
?>
-----
<?
echo "travail sur les chaines de texte", "<br>";
$chain1="bonjour vous";
$chain2="bonjour vous";
$test= strcasecmp($chain1,$chain2);
if ($test==0)
{
echo "les 2 chaines sont identiques";
}
else
{
echo "les 2 chaines sont différentes";
}
//avec la fonction strcasecmp php est insensible à la casse
?>
-----
<?
echo "travail sur les chaines de texte", "<br>";
$chain1="travail sur les chaines de texte";
$posit=12;
$longueur=11;
$chain2= substr($chain1,$posit,$longueur);

```

```

echo $chain2;
?>
-----
<?
echo "travail sur les chaines de texte", "<br>";
$chain1="travail sur les chaines de texte";
$posit=12;
$longueur=11;
$chain2= substr($chain1,$posit,$longueur);
// plusieurs arguments
echo $chain2," et autre texte", "<br>";
// un seul arguments
print ($chain2);
echo "<br>";
// plusieurs arguments
printf ("%s%s%d%f", $chain2,$chain1,$posit,$longueur);
?>
-----
<?
echo "travail sur les chaines de texte html", "<br>";
$chain1="<html><head><title>Ma page en php</title>
</head><body bgcolor='#FF0000'></body></html>";
echo htmlspecialchars($chain1);
// a faire avec et sans htmlspecialchars
?>
-----
<?
echo "travail sur les fichiers", "<br>";
$fileop=fopen("http://localhost/donnees.txt", "r");
while (!feof ($fileop) {
$chaine=fgetc($fileop);
echo $chaine;
}
fclose ($fileop);
//il faut bien sur un fichier donnees.txt enregistré à la racine du site
?>
-----
<?
echo "travail sur les fichiers", "<br>";
$path="http://localhost/DONNEES.TXT";
$mode="r";
$fp=fopen($path, $mode);
while (!feof ($fp))
{
$chr=fgetc($fp);
//avec 80 longueur de ligne
echo $chr, "<br>";
}
fclose($fp);
echo " - et maintenant le est fichier fermé";
// fgets ($fp,80)ligne par ligne
?>
-----
<?
echo "travail sur les fichiers", "<br>";
$path="http://localhost/DON.bin";
$mode="a+";
$str="je peut mettre ce texte dans un fichier";

```

```

if ($fp=fopen($path,$mode))
{
    echo "le fichier est ouvert";
    fwrite($fp,$str);
}
else
    echo " le fichier est pas ouvert ";
    fclose($fp);
?>
-----
<?
echo "travail sur les fichiers","<br>";
$path="http://localhost/DON.bin";
readfile($path);
// pour avoir la taille du fichier
$taille=readfile($path);
echo "<br>",$taille;
?>
-----
<?
echo "travail sur les dates","<br>";
echo date("D d M H : i : s a"),"<br>";
echo date("D"),"<br>";
echo date("M"),"<br>";
echo date("m"),"<br>";
echo date("Y"),"<br>";
?>
-----
<?
echo "travail sur les dates","<br>";
$aujourd'hui= mktime (0,0,0, date ("m"), date("d"), date ("Y"));
echo $aujourd'hui;
echo "<br>",date ("m");
echo "<br>",date ("d");
$jourc= strftime ("%j",$aujourd'hui);
echo "<br>",$jourc;
?>
-----
<?
echo "travail sur les SGBD avec mysql","<br>";
$MySQL_host="localhost";
//$MySQL_User="nomuser";
//$MySQL_Passw="motdepasse";

$cbase=connect();

function connect()
{
    global $MySQL_host;
    If (! $lienb=@mysql_connect("$MySQL_host")){
        //devient ("$MySQL_host","$MySQL_user","$Mysql_Passw")
        echo "pas possible",$Mysql_host;
        exit;
    }
}
return $lienb;
}
echo $cbase;
?>

```



```

-----
<?
echo "travail sur les SGBD avec mysql", "<br>";
$MySQL_Host="localhost";
$MySQL_User="";
$MySQL_Passw="";

$dblk=connect();

function connect()
{
global $MySQL_Host, $MySQL_User, $MySQL_Passw;
if (! $linkid=@mysql_connect("$MySQL_Host", "$MySQL_User", "$MySQL_Passw"))
{
    echo "Impossible d'établir la connexion à ", $MySQL_Host, "<br>";
    exit;
}
return $linkid;
}
echo $dblk;
?>
-----

```

### **création d'une base mysql**

```

<?
include("connection_base.php");
include("mysql_func.php");
$lk=connect();
$db= "test2";
if ($succ=mysql_create_db($db, $lk)) {
    echo "La base de données ", $db, " a été créée";
}
?>
-----

```

### **avec le fichier connection\_base.php**

```

<?
$MySQL_host="localhost";
//$MySQL_User="nomuser";
//$MySQL_Passw="motdepasse";
?>
-----

```

### **et avec le fichier mysql\_func.php**

```

<?
function connect()
{
global $MySQL_host;
If (! $lienb=@mysql_connect("$MySQL_host")){
echo "pas possible", $Mysql_host;
exit;
}
return $lienb;
}
?>
-----

```

### **Création d'une table à une base de données**

```

<?
include("connection_base.php");
include("mysql_func.php");

```

```

$lk=connect();
$db= "test2";
$sql = "CREATE TABLE article (NumArt BIGINT NOT
NULL AUTO_INCREMENT PRIMARY KEY,NumCde VARCHAR(25),
NomArt VARCHAR(100),TexteArt MEDIUMTEXT,PrixArt
DECIMAL(8,2),NumGrArt BIGINT,NumSGrArt BIGINT)";

// pour ajouter une table
$sajouttable=mysql_db_query($db, $sql);
return $ajouttable;
?>

```

---

### Remplir une table avec un fichier de données externe

```

<?
include("connection_base.php");
include("mysql_func.php");
$lk=connect();
$db= "test2";
$sql = "LOAD DATA LOCAL INFILE 'c:/phpdev/www/data.txt'
INTO TABLE article FIELDS TERMINATED BY ';'
(NumCde, NomArt, TexteArt, PrixArt, NumGrArt, NumSGrArt)";
// pour ajouter les données
$sajoutdon=mysql_db_query($db, $sql);
return $ajoutdon;

?>

```

---

### Sélection de données par ordre SQL

```

<?
include("connection_base.php");
include("mysql_func.php");
include("foncsortie.php");
//pour commencer on sélectionne tout
$sql="SELECT * FROM article";
//après on peut restreindre la sélection avec par exemple l'ordre suivant
//$sql="SELECT Numcde, PrixArt FROM article where Numcde>10";
$res=mysql_db_query("test2",$sql);
sortie($res);
?>

```

---

### Avec le fichier fonctionsortie.php qui permet de mettre en forme les données

```

<?
function sortie( $result) {
$nombre=mysql_num_fields($result);
echo "<table width=90% border=0 cellpadding='2' cellspacing='2'>";
echo "<tr bgcolor=#D0D0D0>";
for ($i=0;$i<$nombre;$i++) {
    echo "<th>";
    echo mysql_field_name($result,$i);
    echo "</th>";
}
echo "</tr>";
echo "<tr>";
while ($row = mysql_fetch_array($result,MYSQL_ASSOC)) {
foreach ($row as $elem) {
    echo "<td bgcolor='#E8E8E8'><font size='-1'>$elem</font></td>";
}
}

```

```
echo "</tr>";  
}  
echo "</table>";  
}  
?>
```