

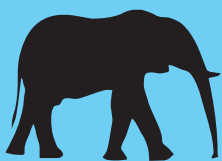


# Les 500 P &

3<sup>e</sup> édition

Cyril Pierre de Geyer  
Guillaume Ponçon

avec la contribution  
de Stéphane Mariel



EYROLLES



# Constantes et variables

## La constante « **CONSTANTE** » contient « 10 »

```
define('CONSTANTE', 10);  
echo CONSTANTE;
```

Les constantes sont généralement utilisées pour définir les paramètres d'une application web (racine du site, nom de la base de données...).

## La variable « **\$variable** » contient « 10 »

```
$variable = 10;  
echo $variable;
```

Toutes les variables commencent par un dollar (\$) en PHP.

## Opérateurs d'affectation

### Affecter une valeur à une variable

```
$chaine = 'de caractères !';  
$entier = 10;
```

### Conversion de type (transtypage)

PHP définit dynamiquement le type de la variable. Il est cependant possible de le forcer en utilisant le transtypage :

```
// (int), (float), (string), (bool)  
$int = (int) $var;
```

#### CONSEIL D'EXPERT Pourquoi convertir le type d'une variable ?

Selon son type, la valeur d'une variable n'aura pas le même comportement.

### Lier des chaînes de caractères (concaténation)

```
$table = 'avion';  
$sql = 'SELECT * FROM '.$table; // Concaténation  
$sql .= ' WHERE id = 345'; // Concaténation  
// $sql contient « SELECT * FROM avion WHERE id = 345 »
```

### Somme et incrémentation

```
$i = 5;  
$i += 5; // Correspond à $i = $i + 5; $i contient 10  
$i++; // Ajoute 1 à $i, utile dans les boucles
```

## Opérateurs de comparaison

Opérateur	Signification	Renvoie TRUE si	Exemple
==	Égal à	\$a est égal à \$b	\$a == \$b
<	Inférieur à	\$a est plus petit que \$b	\$a < \$b
>	Supérieur à	\$a est plus grand que \$b	\$a > \$b
<=	Inférieur ou égal à	\$a est inférieur ou égal à \$b	\$a <= \$b
>=	Supérieur ou égal à	\$a est supérieur ou égal à \$b	\$a >= \$b
!=	Différent de	\$a est différent de \$b	\$a != \$b
===	Identique, en type et valeur	\$a est égal à \$b et leur type est le même. ("5" est différent de 5)	\$a === \$b
!==	Différent en valeur ou en type	\$a a une valeur différente de \$b ou est de type différent	\$a !== \$b

## Conditions

### Condition simple : if

```
if (condition1) {  
    action  
}  
elseif (condition2) {  
    action2  
}  
else {  
    action3  
}
```



# Conditions

## EXEMPLE

```
// mt_rand(0,5) renvoie une valeur aléatoire comprise entre 0 et 5
if (mt_rand(0,5) == 5) {
    echo "La valeur aléatoire est 5";
} else {
    echo "La valeur aléatoire n'est pas 5";
}
```

## Condition multiple : switch

```
switch (variable) {
case <valeur1> :
    // action exécutée si variable vaut valeur1
    ....
    break; // sortir de la condition
case <valeur2> :
    ...
default :
    // action par défaut
}
```

## Opérateur ternaire

```
// if ... else ... en une seule instruction
echo isset($var) ? '$var contient ' . $var : '$var n\'existe pas';
```

# Boucles

## Boucle while

```
while (condition) {
    // action exécutée tant que condition est vraie
    ...
}
```

## EXEMPLE Pour boucler de 0 à 9

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

## Boucle for

```
for (initialisation; condition; incrementation) {
    // action exécutée tant que condition est vraie
}
```

## EXEMPLE Pour boucler de 0 à 9

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

## Boucle foreach : itérer sur un tableau ou une collection

Pour manipuler la n-ième valeur du tableau :

```
foreach ($array_or_collection as $element) {
    // action
}
```

Pour manipuler la n-ième clé *et* valeur du tableau :

```
foreach ($array as $key => $element) {
    // action
}
```

# Les superglobales

Les superglobales sont des tableaux prédéfinis accessibles partout dans le code. Elles contiennent des informations utiles sur votre environnement.



# Les superglobales

Superglobale	Contenu
<code>\$_GET[]</code>	Contient les données envoyées via l'URL (ex : <code>http://www.toto.com/index.php?id=15&amp;nom=titi</code> ).
<code>\$_POST[]</code>	Contient les données envoyées via la méthode POST (généralement utilisée pour les formulaires).
<code>\$_FILES[]</code>	Contient la liste et les caractéristiques des fichiers transmis au serveur par le visiteur.
<code>\$_SERVER[]</code>	Contient des informations sur la requête en cours, sur PHP et sur le serveur HTTP.
<code>\$_ENV[]</code>	Contient des informations « bas niveau » sur votre environnement (système d'exploitation).

## PRATIQUE Accès aux superglobales

Pour plus de sécurité, préférez l'utilisation de la fonction `filter_input()` plutôt qu'un accès direct aux superglobales `$_ENV`, `$_SERVER`, `$_POST` et `$_GET`.

## ASTUCE Connaître le contenu de son environnement

L'appel de la fonction `phpinfo()` affiche le contenu de `$_SERVER` et `$_ENV`. Il est possible d'utiliser les fonctions `print_r()` ou `var_dump()` pour afficher le contenu des autres superglobales.

## Déclaration et appel de fonctions

### Déclarer une fonction

```
function nom_de_la_fonction($arg1, $arg2, ...) {  
    // liste d'instructions  
}
```

### Argument optionnel, avec valeur par défaut

```
function nom_de_la_fonction($arg = 12) { ... }
```

### Retourner une valeur avec « return »

```
function nom_de_la_fonction($arg = 12) {  
    if ($arg == 12) {  
        return 'La valeur est 12';  
    }  
    return 'La valeur est différente de 12';  
}
```

### Appeler une fonction

```
// nom_de_la_fonction (argument1, argument2, ...);  
$chaine = nom_de_la_fonction(10);  
echo 'Retour de fonction : '.$chaine;
```

## Fonctions d'affichage

### Afficher simplement un texte

```
echo 'Best practices'; // Affiche « Best practices »
```

### Affichage formaté

```
// Affiche « PHP 05, PHP 5.10, PHP 101 »  
printf('PHP %2d, PHP %.2f, PHP %b', 5.1, 5.1, 5.1);
```

### Déboguer les variables

```
echo '<pre>'; // amélioration de l'affichage html  
print_r($tableau);  
var_dump($objet);  
echo '</pre>';
```

### Déclarer et afficher une chaîne

```
$cd = 'ef'; // la variable $cd contient 'ef'  
echo 'ab$cd'; // Affiche « ab$cd »  
echo "ab$cd"; // Affiche « abef »
```



# Fonctions d'affichage

## PRATIQUE Guillemets ou apostrophes ?

Avec des guillemets (""), PHP interprète le contenu (et donc remplace les variables par leur valeur et les codes de caractères spéciaux tels que `\n` par leur équivalent). Avec des apostrophes (' '), PHP ne fait que copier-coller le contenu. Avec des guillemets, si une variable ne doit pas être interprétée, placer un `\` devant :  
`$a = "le contenu de $age est $age.\n";`

## Dates

### Afficher une date

Le timestamp Unix est un entier qui représente une date (nombre de secondes depuis le début de l'époque Unix).

```
/* En ajoutant 3600 * 24 secondes, on ajoute une journée*/
$timestamp_demain = time() + 3600 * 24;
echo date('d/m/Y', $timestamp_demain); // Affiche 02/01/2015
```

### Paramètres de formatage de la fonction date

Code	Description	Code	Description
d	Jour du mois (ex : 01)	j	Jour du mois (ex : 1)
D	Jour de la semaine (ex : Mon)	z	Jour de l'année (ex : 34)
w	Jour de la semaine (numérique)	W	Numéro de semaine (ex : 42)
l	Jour de la semaine textuel	Y	Année (ex : 2015)
F	Mois textuel (ex : December)	y	Année (ex : 15)
m	Mois numérique (ex : 06)	n	Mois numérique (ex : 6)
M	Mois en trois lettres (ex : Jan)	t	Nombre de jours dans le mois
g	Heure au format 12h (ex : 6)	G	Heure au format 24h (ex : 18)
h	Heure au format 12h (ex : 06)	H	Heure au format 24h (ex : 06, 18)
i	Minutes (ex : 05)	s	Secondes (ex : 02)
c	Date ISO8601 (PHP 5)	r	Date RFC 2822

### Construire un timestamp Unix

```
$timestamp = mktime(0,0,0,7,14,2008); // 14/07/2008
echo date('l', $timestamp); // Affiche «Friday»
```

### Vérifier une date

```
checkdate($mois, $jour, $annee); // Renvoie TRUE si la date existe
```

## CONSEIL D'EXPERT Manipuler des dates en PHP

Depuis PHP 5.2, un objet `DateTime` permet de représenter et manipuler une date et une heure.

## Typage et arrondis numériques

### Vérifier le type d'une valeur

```
is_int($var); // vrai si $var contient un entier
is_numeric($var); // vrai si $var est un nombre
is_array($var); // vrai si $var contient un tableau
is_string($var); // vrai si $var est une chaîne
is_float($var); // vrai si nombre à virgule flottante
```

### Arrondir, tronquer des valeurs numériques

```
echo ceil(4.2); // affiche l'entier supérieur (5)
echo floor(4.2); // affiche l'entier inférieur (4)
echo round(4.2); // affiche l'entier le plus proche (4)
```

## Chaînes de caractères

### Taille d'une chaîne

```
echo strlen($str); // Affiche la longueur de $str
```

### Formatage d'une chaîne avec printf et ses dérivés

```
$str = "%. -25s %d\n";
printf ($str, 'Note des lecteurs', 20);
//Affiche: Note des lecteurs..... 20
```

### Position d'une chaîne dans une autre

```
echo strpos($s1, $s2); // Position de $s2 dans $s1
```



# Chaînes de caractères

## Remplacement

```
// Remplace 'old' par 'new' dans $str
$str = str_replace('old', 'new', $str);
// Renvoie les quatre premiers caractères à partir du deuxième
$str = substr($str, 2, 4);
```

## Nettoyage

Suppression des caractères superflus aux extrémités

```
$clean = trim(" Bonjour ! \n ");
```

## Formatage HTML

Conversion des caractères spéciaux en HTML

```
echo htmlspecialchars($str);
```

## Redirection d'une page vers une URL

```
header('Location: http://www.openska.com');
```

### CONSEIL D'EXPERT Chaînes de caractères unicode

Si vos données utilisent un jeu de caractères unicode tel que UTF-8, utilisez impérativement les fonctions multi-octets (mb\_substr, mb\_strlen, ...).

```
mb_internal_encoding('UTF-8');
echo mb_strlen("càd"); // affiche 3. strlen() afficherait 4
```

# Gestion des courriels

Lors de l'envoi de courriels, l'ensemble du traitement se fait par l'intermédiaire d'une seule fonction : mail().

## Envoi d'un courriel simple

```
$dest = "cyril@php.net,guillaume@openstates.com";
$sujet = "Mon premier courriel";
$corps = "Monsieur \nLigne2 \nJe remarque ...";
mail($dest, $sujet, $corps);
```

### PRATIQUE Signaler les sauts de ligne

Pour signaler les sauts de ligne dans le message, il suffit d'utiliser le caractère spécial \n et de faire en sorte que ces caractères soient interprétés en délimitant la chaîne avec des guillemets (") et non avec des apostrophes (').

## Envoi d'un courriel à plusieurs destinataires en changeant l'expéditeur

```
$dest = 'cyril@openska.com, luc@openska.com';
$sujet = 'Vos cotisations !';
$entete = "From: responsable@urssaf.fr\n";
$entete .= "Reply-to: adresseretour@toto.fr\n";
mail($dest, $sujet, "Bonjour ...", $entete);
```

## Envoi d'un courriel au format HTML

```
$dest = 'cyril@php.net';
$sujet = 'Courriel au format HTML';
$corps = '<html><body><b>Bonjour</b>...';
$entete = "Content-Type: text/html";
mail($dest, $sujet, $corps, $entete);
```

# Gestion des formulaires

## Formulaire minimal en HTML

```
<form method="post" action="form.php">
  <input type="text" name="email">
  <input type="submit" name="ok" value="OK">
</form>
```

## Récupération des informations en PHP

```
// Tableau associatif contenant les variables passées par l'URL (GET)
$_GET[]

// Tableau associatif contenant les variables passées par POST
$_POST[]

// Tableau contenant les fichiers envoyés via le formulaire
$_FILES['fichier'][]
```



# Gestion des formulaires

## PRATIQUE Filtrer et valider avec PHP 5.2 et +

Utilisez `filter_input_array()` pour récupérer les valeurs. Exemple :

```
$values = filter_input_array(INPUT_POST);
```

### Balise HTML pour l'upload de fichier

```
<form action="recept.php" method="POST"
      enctype="multipart/form-data">
<input type="file" name="fichier">
```

### Récupération du fichier en PHP

```
$fichier = $_FILES['fichier']['name'];
$taille = $_FILES['fichier']['size'];
$tmp = $_FILES['fichier']['tmp_name'];
$type = $_FILES['fichier']['type'];
$erreur = $_FILES['fichier']['error'];
move_uploaded_file($tmp, 'nouveau_nom');
```

### Balises HTML pour passer un tableau en paramètre

```
<input type="text" name="person[premier]">
<input type="text" name="person[nom]">
```

### Affichage du tableau passé en paramètre

```
echo 'Nom : ', $_GET['person']['nom'];
```

## Les sessions

Les sessions permettent de conserver les informations concernant un utilisateur tout au long de sa navigation (ex. : panier d'achat, espace sécurisé...).

### Initialiser une session

```
session_start();
```

**ATTENTION** Ne rien envoyer au navigateur avant l'initialisation des sessions.

### Mettre une variable dans la session

```
$_SESSION['auteurs'] = "Guillaume et Cyril";
var_dump($_SESSION);
echo 'hello ' . $_SESSION['auteurs'] . ' !<br />';
```

### Fermer une session

```
session_destroy();
```

## Gestion des tableaux

### Déclarer un tableau

```
$tab = array();
$tab2 = array(1, 2, "chaîne", 45.3);
```

### Ajouter des valeurs à un tableau

```
$tab[] = "Guillaume"; // clé implicite: 0
$tab[] = "Cyril";      // clé implicite: 1
```

### Tableau associatif (clés et valeurs liées)

```
// $tab['clé'] = "valeur";
$tab['nom'] = "Ponçon";
$tab['prenom'] = "Guillaume";
```

### Connaître le nombre d'éléments d'un tableau

```
$nb = count($tab); // nombre d'éléments
```

### Rechercher un élément dans un tableau

Recherche d'une clé à partir d'une valeur

```
$key = array_search('Guillaume', $tab);
```

### Vérifier si un tableau contient une valeur

```
// Est-ce qu'il y a « Linux » dans le tableau $tab ?
if (in_array('Linux', $tab)) {
    echo 'Trouvé';
}
```



# Gestion des tableaux

## Trier un tableau

```
sort($tab); // trie un tableau
asort($tab); // trie un tableau en conservant la liaison clé-valeur
rsort($tab); // trie un tableau en ordre inverse
```

## Supprimer une valeur

```
unset($tab['prenom']); // suppression
```

## Une variable est-elle un tableau ?

```
if (is_array($tab)) {... // est-ce un tableau ?
```

## Boucler sur un tableau

```
foreach ($tab as $key => $value) {
    echo "$key : $value\n";
}
```

# Gestion des fichiers

## Récupération expresse du contenu d'un fichier ou d'une URL

```
$cFile = file_get_contents('fichier.txt');
```

## Écriture expresse dans un fichier

```
$var = 'PHP est une plate-forme de développement';
file_put_contents('fichier.txt', $var);
```

## Lecture bufferisée d'un fichier

```
// Conseillé pour manipuler de gros fichiers
$fp = fopen('monfichier.txt', 'r');
while (!feof($fp)) {
    echo fgets($fp, 1024);
}
fclose($fp);
```

## Fonctions de manipulation de fichiers

Fonction	Effet
fopen()	Permet d'ouvrir un fichier. La fonction renvoie un descripteur à utiliser pour les traitements suivants.
fgets()	Renvoie une ligne du fichier. Prend en paramètre le file descriptor du fichier et le nombre maximum de caractères à lire.
fread()	Renvoie <i>n</i> caractères d'un fichier.
fwrite()	Écrit dans un fichier.
fclose()	Ferme la connexion au fichier.

## Modes d'ouverture de fichiers

Mode	Signification
r	Ouvre en lecture seule ; place le pointeur en début de fichier.
r+	Ouvre en lecture et écriture ; place le pointeur en début de fichier.
w	Ouvre en écriture seule ; place le pointeur au début du fichier et réduit la taille du fichier à 0. Si le fichier n'existe pas, il est créé.
w+	Ouvre en lecture et écriture ; place le pointeur au début du fichier et réduit la taille du fichier à 0. Si le fichier n'existe pas, il est créé.
a	Ouvre en écriture seule ; place le pointeur à la fin du fichier. Si le fichier n'existe pas, il est créé.
a+	Ouvre en lecture et écriture ; place le pointeur à la fin du fichier. Si le fichier n'existe pas, il est créé.

## Fonctions d'information

Fonction	Effet
feof(\$descr) et feof(\$descr)	Test de fin de ligne/fin de fichier
file_exists(\$nom_fichier)	Existence d'un fichier
filesize(\$nom_fichier)	Taille d'un fichier
stat(\$nom_fichier)	Informations diverses sur un fichier





# Gestion des fichiers

## Exploiter un fichier de tableau (CSV)

- CSV (*Comma Separated Values*) est un format standard d'échange pour les tableaux. Il s'agit simplement d'un tableau à deux dimensions représenté dans un fichier.

- Lecture d'un fichier CSV

```
$fp = fopen('fichier.csv', 'a+');
while ($tab = fgetcsv($fp, 1000)) {
    print_r($tab);
}
```

## Exploiter un fichier de configuration (INI)

- PHP offre un moyen simple d'utiliser des fichiers de configuration classique de syntaxe similaire au php.ini.

- Lecture d'un fichier .ini

```
$tab = parse_ini_file('monfichier.ini');
```

# Bases de données

L'extension PDO permet de se connecter à de nombreux SGBD du marché.

## Connexion à la base de données MySQL

La chaîne \$dsn (*Data Source Name*), contient le type de base de données et son nom.

```
$dsn = "mysql:host=sql.openstates.com;dbname=users";
$login = "user"; $pass = "pass";
// Connexion à la base de données
$pdo = new PDO($dsn, $login, $pass);
// Lancer une exception en cas d'erreur
$pdo->setAttribute(PDO::ATTR_ERRMODE,
    PDO::ERRMODE_EXCEPTION);
```

## Exécuter une requête SELECT

```
$sql = "SELECT name FROM users LIMIT 0, 10";
$users = $pdo->query($sql);
while ($user = $users->fetch()) {
    echo $user['name'];
}
```

## Exécuter une insertion paramétrée

```
$sql = "INSERT INTO user (name, email) VALUES (?, ?)";
$stmt = $pdo->prepare($sql);
$newUser = array('Guillaume', 'contact@openstates.com');
$stmt->execute($newUser);
```

### PERFORMANCES ET SÉCURITÉ

Les requêtes paramétrées (ou préparées) permettent d'exécuter plusieurs fois la même requête avec des données différentes et d'optimiser ainsi les performances et la sécurité (contre les injections SQL par exemple).

## Connexion à Oracle

```
$dsn = "oci:dbname="//oracle.openstates.com/XE";
$pdo = new PDO($dsn);
```

## Connexion à SQLite

```
$dsn = "sqlite2:ma_base.db";
$pdo = new PDO($dsn);
```

## Gestion des transactions

```
$pdo->beginTransaction();
try {    // Exécuter les requêtes ici
    $pdo->commit();    // Validation de toutes les requêtes
} catch (PDOException $e) {
    $pdo->rollback(); // Annulation de toutes les requêtes
}
```



# Programmation orientée objet

## Composition d'une classe

```
class ConfigManager extends Config {  
    private    $host;  
    protected $port;  
    public     $name;  
  
    public function __construct(...) { ... }  
}
```

- `ConfigManager` est le nom de la classe.
- `Config` est le nom de la classe mère, précédé du mot-clé `extends`. Les propriétés et méthodes `public` et `protected` de la classe mère sont accessibles dans la classe fille `ConfigManager`.
- `private`, `protected` et `public` sont appelées « portées ».
- `$host`, `$port` et `$name` sont les propriétés de la classe `ConfigManager`.
- `__construct` est une méthode de la classe `ConfigManager`.

## Portée (ou visibilité) des attributs (propriétés et méthodes)

- `private` : accessible uniquement dans la classe courante.
- `protected` : accessible dans la classe courante et les classes filles.
- `public` : accessible partout.

### TERMINOLOGIE OBJET

- Propriétés : variables d'une classe.
- Méthodes : fonctions d'une classe.
- Constructeur `__construct` : méthode spéciale appelée lors de l'instanciation de la classe (appel avec `new`).
- Destructeur `__destruct` : méthode spéciale appelée lors de la désinstanciation de la classe (suppression avec `unset` ou en fin de script).

## Manipuler une classe

- Instanciation d'une classe  
`$configManager = new ConfigManager();`
- Appel d'une méthode  
`$value = $configManager->getValue('host');`

# Les expressions régulières

Elles sont utilisées pour effectuer toutes sortes de manipulations sur les chaînes de caractères. Il existe deux types d'expressions régulières : les POSIX et les PCRE. Ces dernières étant plus performantes en PHP, elles seront utilisées dans nos exemples.

### CONSEIL Utiliser les expressions régulières avec précaution

Utiliser en priorité les fonctions natives de manipulation de chaînes (`substr`, `strtr`, etc.). Les expressions régulières sont pratiques mais pas toujours performantes.

## Symboles utilisés dans les expressions régulières

Symbole	Description
<code>^</code>	Marque le début d'une chaîne
<code>\$</code>	Marque la fin d'une chaîne
<code> </code>	Marque l'alternative (ou)
<code>-</code>	Indique un intervalle. Ex : <code>[a-d]</code> = entre a et d (a, b, c ou d)
<code>()</code>	Capture de motif (utilisé pour « matcher » un contenu.
<code>[]</code>	Classes de caractères. Ex : <code>[0-9a-f]</code> = un chiffre hexadécimal
<code>[:x:]</code>	Classe prédéfinie (x = alpha, digit, space, alnum, lower, upper, xdigit, blank, print)
<code>.</code>	Tout caractère
<code>?</code>	Facultatif (0 ou 1 occurrence)
<code>*</code>	Facultatif (0 ou plusieurs occurrences)
<code>+</code>	Obligatoire (1 ou plusieurs occurrences)
<code>{x}</code>	Doit apparaître exactement x fois (x étant un nombre)
<code>{x,}</code>	Doit apparaître au moins x fois
<code>{x,y}</code>	Doit apparaître entre x et y fois



# Les expressions régulières

## Fonctions PCRE usuelles

Fonction	Description
<code>preg_match (\$motif, \$chaine)</code>	Recherche le motif dans la chaîne.
<code>preg_match_all (\$motif, \$chaine)</code>	Recherche toutes les occurrences du motif dans la chaîne.
<code>preg_replace(\$motif, \$remplacement, \$chaine)</code>	Effectue un remplacement sur toutes les occurrences correspondant au motif dans la chaîne.

## Vérifier la validité syntaxique d'une adresse e-mail

```
$mail = 'guillaume.poncon@openstates.com'
$exp = '/^[a-z0-9\._-]+@[a-z0-9\._-]+\.[a-z]{2,4}$/i';
echo preg_match($exp, $mail);
```

## Extraire les balises HTML

```
$html = '<b>Best practices <i>PHP 5</i></b><br />';
$exp = '|<([>]+) */?>|U';
preg_match_all($exp, $html, $out);
var_dump($out);
```

## XML

### Lire un flux RSS avec SimpleXML

```
$url = "http://rss.news.yahoo.com/rss/world";
$feed = simplexml_load_file($url);
foreach($feed->channel->item as $item) {
    echo $item->title."\n";
}
```

### Compter le nombre d'éléments d'un flux RSS [DOM]

```
$url = "http://rss.news.yahoo.com/rss/world";
$doc = new DOMDocument();
$doc->loadXML(file_get_contents($url));
$xpath = new DOMXPath($doc);
$query = 'count(//channel/item)';
echo $xpath->evaluate($query); // PHP 5.1 ou +
```

## SOAP

SOAP est un protocole de communication très utilisé, basé sur XML. Dans l'exemple suivant, le serveur doit être accessible à travers l'URL `http://soapserver/` (spécifiée dans le client).

### Exemple de serveur SOAP

```
function double($x) {
    return (int) $x * 2;
}
$params = array('uri' => 'http://test-uri/');
$server = new SoapServer(null, $params);
$server->addFunction('double');
$server->handle();
```

### Exemple de client SOAP

```
$params = array('location' => 'http://soapserver/',
                'uri' => 'http://test-uri/');
$client = new SOAPClient(null, $params);
echo $client->double(2);
```

## Administration SQL

### Créer une base de données

```
CREATE DATABASE base;
```

### Créer une table

```
CREATE TABLE client(idClient int(11), nom char(60));
```



# Administration SQL

On peut utiliser le paramètre AUTO\_INCREMENT sur un champ pour avoir une clé primaire numérique.

```
CREATE TABLE client (  
    idClient int(11) NOT NULL auto_increment,  
    nom char(60),  
    PRIMARY KEY (idClient)  
);
```

## Modifier la structure d'une table

```
ALTER TABLE table  
ADD INDEX [nom_index] (nom_champ,...)
```

ou

```
ADD PRIMARY KEY (nom_champ,...)
```

ou

```
ADD UNIQUE [nom_index] (nom_champ,...)
```

ou

```
ADD FULLTEXT [nom_index] (nom_champ,...)
```

## Supprimer une table

```
DROP TABLE [IF EXISTS] table
```

## Insertion de données

### Insertion standard

```
INSERT [INTO] nom_de_table [(nom_colonne,...)]  
VALUES (...),...
```

Si un champ est défini comme étant auto-incrémenté, il suffit de lui donner la valeur NULL pour que MySQL se débrouille tout seul.

```
INSERT client VALUES (NULL, 'BOURDON')
```

### Insertion via une sous-requête

```
INSERT [INTO] nom_de_table [(nom_colonne,...)]  
SELECT...
```

### Insertion non complète via SET

```
INSERT [INTO] nom_de_table SET  
nom_colonne=(expression)...
```

### Insertion multiple en une passe

```
INSERT INTO table (champ1, champ2) VALUES  
( 'val11', 'val12'), ('val21', 'val22'), ('val31', 'val32')
```

## Modification et suppression de données

### Modifier des données (UPDATE)

```
UPDATE nom_de_table  
SET nom_colonne1=expr1 [, nom_colonne2=expr2, ...]  
[WHERE where_definition]
```

Pour mettre à jour un enregistrement, indiquer sa clé primaire.

```
UPDATE client SET nom='PIERRE' WHERE idClient=3;
```

### Remplacer des données (REPLACE)

```
REPLACE  
    [INTO] nom_de_table [(nom_de_colonne,...)]  
VALUES (expression,...), (...), ...
```

ou

```
REPLACE  
    [INTO] nom_de_table [(nom_de_colonne,...)]  
SELECT ...
```

ou

```
REPLACE  
    [INTO] nom_de_table  
    SET col_name=expression,  
    nom_de_colonne=expression,...
```



# Modification et suppression des données

**NOTE** Il s'agit d'une commande spécifique à MySQL. Avec d'autres SGBD, REPLACE fonctionne différemment.

## Supprimer des données (DELETE)

```
DELETE FROM nom_de_table
[WHERE clause_where]
[ORDER BY ...]
[LIMIT lignes]
```

## Récupération de données

### Récupérer des données (SELECT)

```
SELECT champ
[FROM table_1]
[WHERE condition]
```

```
mysql> select titre,prix from anaska_formation;
```

titre	prix
Apache	900
Cocoon	2590
Dreamweaver niveau 1	1230
Dreamweaver niveau 2	950
Flash niveau 1	1230
Flash niveau 2	1320
GIMP	950
Webmaster Technique	20
PHP : Internet et Extranet Dynamiques	20
MySQL : Administration et utilisation	20
J2EE et STRUTS	1350
Java niveau 1	1950
Java : Servlets JSP	2175
Java : SWING Client Riche	2325
Linux : installation, administration et configuration réseau	2050
Linux : Administration niveau 2	2050

### Sélectionner l'ensemble des champs d'une table

```
SELECT * FROM table
```

### Nommer un champ avec « AS »

```
SELECT MIN(nom_champ) AS minimum FROM table
```

### Trier avec « ORDER BY »

```
SELECT * FROM table ORDER BY champ [DESC|ASC]
```

### Limiter le nombre de lignes résultat avec « LIMIT »

```
SELECT * FROM table LIMIT nombre
// LIMIT <à partir de>, <nombre de lignes>
SELECT * FROM table LIMIT 5, 10
```

### Compter le nombre de lignes

```
SELECT count(*) AS nbr FROM table WHERE condition
```

### Retenir la valeur minimale ou maximale

```
SELECT MIN(nom_champ) AS minimum from table
SELECT MAX(nom_champ) AS maximum from table
```

### Enlever les doublons (DISTINCT)

```
SELECT DISTINCT champ FROM table WHERE condition
```

## Filtrer avec la clause WHERE

### Syntaxe générale de WHERE

```
[SELECT | UPDATE | DELETE]
WHERE condition
```

ou

```
WHERE champ IN ('valeur1','valeur2','valeur3')
```

ou

```
WHERE champ NOT IN ('valeur1','valeur2','valeur3')
```

ou

```
WHERE champ BETWEEN 'limite1' AND 'limite2'
```

ou

```
WHERE champ LIKE 'expression'
```



## Les outils pour développer en PHP

## Éditeurs spécialisés

- NetBeans IDE
- Eclipse + module PHPEclipse
- PHPEdit ou ZendStudio

## Accélérateurs

- Alternative PHP Cache (APC)
- Zend Server
- OPcache (PHP > 5.2)

## Débogueurs et profileurs

- Advanced PHP Debugger (APD)
- Xdebug + Kcachegrind/Wincachegrind
- PHP Debugger DBG
- Xhprof

Warning: <code>ldap_bind()</code> function <code>ldap_bind()</code> Unable to bind to server: Can't contact LDAP server in /web/ggn/pr/fraternity/classes/ldap.php on line 145		
Call Stack		
#	Function	Location
1	{main()}	/web/ggn/pr/fraternity/home/parameters/ldap_edit.php:0
2	LDAP->edit()	/web/ggn/pr/fraternity/home/parameters/ldap_edit.php:28
3	LDAP->search()	/web/ggn/pr/fraternity/classes/ldap.php:47
4	LDAP->rssearch()	/web/ggn/pr/fraternity/classes/ldap.php:126
5	LDAP->open()	/web/ggn/pr/fraternity/classes/ldap.php:170
6	ldap_bind()	/web/ggn/pr/fraternity/classes/ldap.php:145
Dump \$_GET Dump \$_POST Dump \$_COOKIE		
Variables in local scope (#5)		
Variable		Value
<code>\$id_ldap</code>	<code>resource(219) ldap (link)</code>	
<code>\$f</code>	<code>stdClass Object</code>	

## ERREURS CLASSIQUES ET CONSEILS

- Parse error : vérifiez s'il ne manque pas un ; aux lignes précédant la ligne indiquée dans le message d'erreur.
- Cannot send session cache limiter - headers already sent : attention, ne faites pas d'écho et ne laissez pas les retours à la ligne avant l'appel à `session_start()`, `header()`, etc.
- Warning: `mysql_fetch_array()`: supplied argument is not a valid MySQL result resource : votre requête ne renvoie aucun résultat exploitable. Vérifiez :
  1. votre connexion à la base ;
  2. votre requête SQL.
- Call to a member function on a non-object : vous essayez de faire appel à une méthode dont l'objet n'existe pas. Vérifiez que vous avez bien instancié votre objet.
- Failed opening required "lib.php" (include\_path=".:var/www/lib") : PHP n'a pas réussi à ouvrir un fichier (généralement une bibliothèque). Vérifiez que votre fichier est bien présent. Vous pouvez aussi utiliser la directive de configuration `include_path` (dans votre fichier `php.ini`).

## MÉTHODE DE RÉOLUTION D'UN PROBLÈME

### Quelques pistes

- Affichez la requête SQL et vérifiez-la dans un outil tel que PHPMyAdmin.
- Utilisez `var_dump()` pour afficher le contenu et le type de votre variable.
- Vérifiez que vous êtes bien connecté à votre base de données.

[illegible]

## Méthode générique

- Faites un copier-coller du message d'erreur dans votre moteur de recherche.
- Rendez-vous sur [www.php.net](http://www.php.net) et faites une recherche sur la fonction qui pose problème. Consultez les commentaires des utilisateurs.
- Allez sur un forum (ex : [www.phpfrance.com](http://www.phpfrance.com)) et faites une recherche. Si aucune réponse ne convient, postez un message en indiquant bien le problème et son contexte.
- Rendez-vous sur le channel IRC [www.phpfrance.com](http://www.phpfrance.com) sur un serveur undernet.

### INFO Travailler les versions 5.3 à 5.6 de PHP

D'importantes notions ont été proposées telles que les espaces de noms, les traits ou les générateurs, ainsi que de nouvelles syntaxes principalement liées à la manipulation des tableaux et à la POO. Ces ajouts sont incompatibles avec les versions de PHP5 antérieures à leurs apparitions.

Chez le même éditeur...

E. DASPET, C. PIERRE DE GEYER, **PHP 5 avancé**,  
N°13435, 2012, 870 pages  
S. MARIEL, **Cahier du programmeur PHP 5**,  
N°11234, 2005, 276 pages  
C. PIERRE DE GEYER, J. PAULI, G. PLESSIS, **Performances  
PHP**, N°12800, 2012, 300 pages  
D. SEGUY, P. GAMACHE, **Sécurité PHP 5 et MySQL**,  
N°13339, 2011, 277 pages  
J-F. LÉPINE, **Mémento Industrialisation PHP 5**,  
N°13480, 2012, 14 pages

Code éditeur : G13602  
SBN : 978-2-212-13602-9

Conception : Nord Compo

