

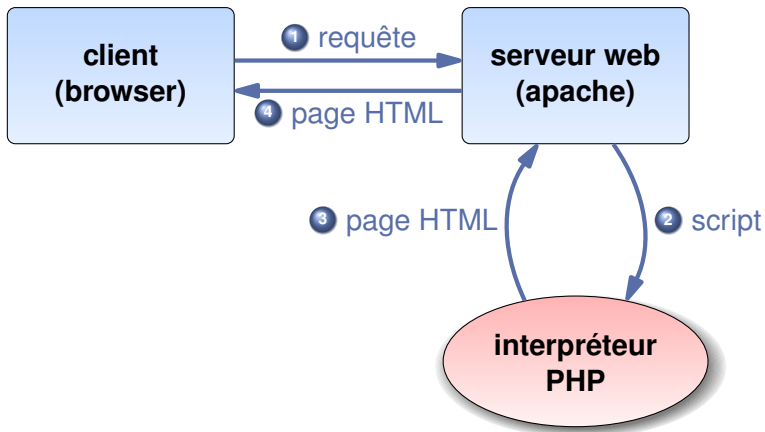
## Cours 2 : Le langage



Applications web et mobiles

Christophe Gonzales

# Architecture autour de PHP



⇒ But : les scripts PHP produisent du texte HTML !  
ils sont aussi utiles pour produire des données JSON

# Scripts PHP

The screenshot shows an IDE window titled "php [~/enseignement/mobile-19-20/prog/php] - .../cours1.php - WebStorm". The main editor displays the following PHP code:

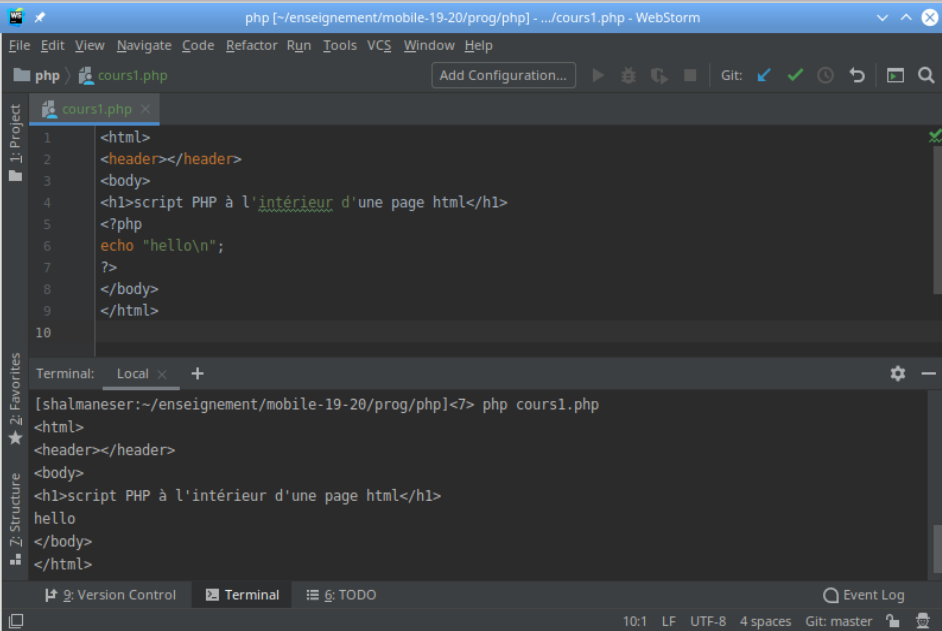
```
1 <?php // début de script PHP
2
3 // les expressions se terminent par des ;
4 echo "hello\n";
5
6 ?> // fin de script
7
```

Below the editor is a terminal window with the following output:

```
Terminal: Local x +
[shalmaneser:~/enseignement/mobile-19-20/prog/php]<4> php cours1.php
hello
[shalmaneser:~/enseignement/mobile-19-20/prog/php]<5> █
```

The IDE interface includes a menu bar (File, Edit, View, Navigate, Code, Refactor, Run, Tools, VCS, Window, Help), a toolbar with icons for running, debugging, and Git, and a sidebar with project and structure views. The status bar at the bottom shows "7:1 LF UTF-8 4 spaces Git: master".

# PHP à l'intérieur de balises HTML



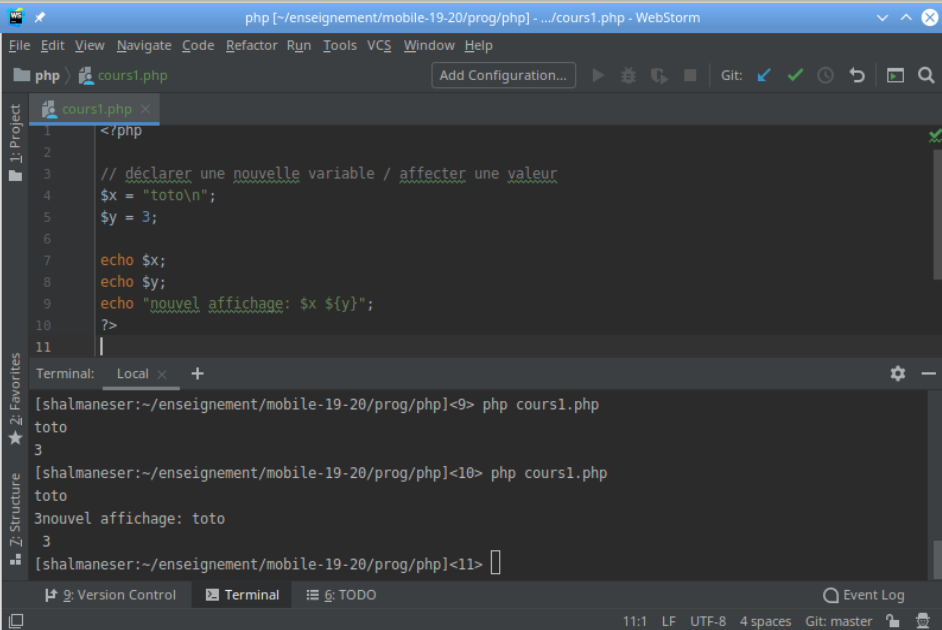
The screenshot shows a code editor window with the following PHP code inside HTML tags:

```
1 <html>
2 <header></header>
3 <body>
4 <h1>script PHP à l'intérieur d'une page html</h1>
5 <?php
6 echo "hello\n";
7 ?>
8 </body>
9 </html>
10
```

The terminal window below shows the command and output:

```
Terminal: Local x +
[shalmaneser:~/enseignement/mobile-19-20/prog/php]<?> php cours1.php
<html>
<header></header>
<body>
<h1>script PHP à l'intérieur d'une page html</h1>
hello
</body>
</html>
```

# Déclarer des variables/affecter des valeurs



The screenshot shows an IDE window titled "php [~/enseignement/mobile-19-20/prog/php] - .../cours1.php - WebStorm". The editor displays the following PHP code in "cours1.php":

```
1 </php
2
3 // déclarer une nouvelle variable / affecter une valeur
4 $x = "toto\n";
5 $y = 3;
6
7 echo $x;
8 echo $y;
9 echo "nouvel affichage: $x ${y}";
10 ?>
11 |
```

The terminal window below shows the execution of the script:

```
Terminal: Local x +
[shalmaneser:~/enseignement/mobile-19-20/prog/php]<9> php cours1.php
toto
3
[shalmaneser:~/enseignement/mobile-19-20/prog/php]<10> php cours1.php
toto
3nouvel affichage: toto
3
[shalmaneser:~/enseignement/mobile-19-20/prog/php]<11> |
```

The status bar at the bottom indicates the time is 11:1, the file is LF, the encoding is UTF-8, there are 4 spaces, and the Git status is master.

# Les chaînes de caractères

The screenshot shows a code editor with the following PHP code in `string.php`:

```
1 <?php
2 $x = 3;
3
4 // chaînes de caractères interprétées (entre guillemets) :
5 $y = "toto $x\n";
6 // chaînes non interprétées (entre quotes) :
7 $z = 'toto $x\n';
8
9 echo $y;
10 echo $z;
11 echo $x . $x; // '.' => concaténation de chaînes de caractères
12 ?>
13
```

The terminal output shows the execution of the script:

```
[shalmaneser:~/enseignement/mobile-19-20/prog/angular/cours3]<1> cd ../../php
[shalmaneser:~/enseignement/mobile-19-20/prog/php]<2> php string.php
toto 3
toto $x\n33
[shalmaneser:~/enseignement/mobile-19-20/prog/php]<3>
```

# Les nombres

The screenshot shows the WebStorm IDE interface. The main editor displays a PHP file named `cours1.php` with the following code:

```
1 <?php
2 $x = 3; // ceci est un entier
3 $y = 4.65; // ceci est un flottant
4
5 echo is_int($x);
6 echo "=====1";
7 echo is_int($y);
8 echo "#####1";
9 echo is_float($y);
10 ?>
11
12
```

Below the editor is a terminal window with the following output:

```
Terminal: Local x +
[shalmaneser:~/enseignement/mobile-19-20/prog/php]<20> php cours1.php
1=====1#####1
[shalmaneser:~/enseignement/mobile-19-20/prog/php]<21> █
```

The IDE interface includes a menu bar (File, Edit, View, Navigate, Code, Refactor, Run, Tools, VCS, Window, Help), a toolbar with icons for running, debugging, and Git, and a sidebar with panels for Project, Favorites, and Structure. The status bar at the bottom shows the current time (11:1), encoding (LF), character set (UTF-8), indentation (4 spaces), and the current Git branch (master).

# Les tableaux

The screenshot shows an IDE window titled "php [-/enseignement/mobile-19-20/prog/php] - .../cours1.php - WebStorm". The code in "cours1.php" is as follows:

```
1 <?php
2 // création de tableaux :
3 $stab1 = array ('x', 'y'); $stab2 = array ('key1' => 'val1', 'key2' => 'val2');
4 $stab3 = ['x', 'y']; $stab4 = [3 => 'val1', 7 => 'val2']; // syntaxe depuis PHP 5.4
5
6 // ajout/ modification d'éléments :
7 $stab1[4] = 'z'; $stab3[] = 'z';
8
9 // accès aux éléments
10 echo $stab1[0] . ' ' . $stab4[3] . "\n";
11 print_r ($stab2);
12 ?>
```

The terminal output shows the execution of the script:

```
[shalmaneser:~/enseignement/mobile-19-20/prog/php]<62> php cours1.php
x val1
Array
(
    [key1] => val1
    [key2] => val2
)
```

The IDE interface includes a sidebar with "Project", "Favorites", and "Structure" views. The bottom status bar shows "All files are up-to-date (4 minutes ago)", "12:3", "LF", "UTF-8", "4 spaces", "Git: master", and "Event Log".



# Les constantes

The screenshot shows an IDE window titled "php [~/enseignement/mobile-19-20/prog/php] - .../cours1.php - WebStorm". The editor displays the following PHP code:

```
1 <?php
2 // déclarer une constante :
3 define ("MYCONST", "toto\n");
4 define ("CST2", 3.4);
5
6 // utiliser une constante :
7 echo MYCONST;
8 echo CST2;
9 ?>
10
11
```

Below the editor is a terminal window with the following content:

```
Terminal: Local x +
[shalmaneser:~/enseignement/mobile-19-20/prog/php]<27> php cours1.php
toto
3.4
[shalmaneser:~/enseignement/mobile-19-20/prog/php]<28> 
```

The IDE interface includes a menu bar (File, Edit, View, Navigate, Code, Refactor, Run, Tools, VCS, Window, Help), a breadcrumb path (php > cours1.php), and a status bar at the bottom showing "10:1 LF UTF-8 4 spaces Git: master".

# Les fonctions

The screenshot shows an IDE window titled "php [~/enseignement/mobile-19-20/prog/php] - .../cours1.php - WebStorm". The editor displays the following PHP code:

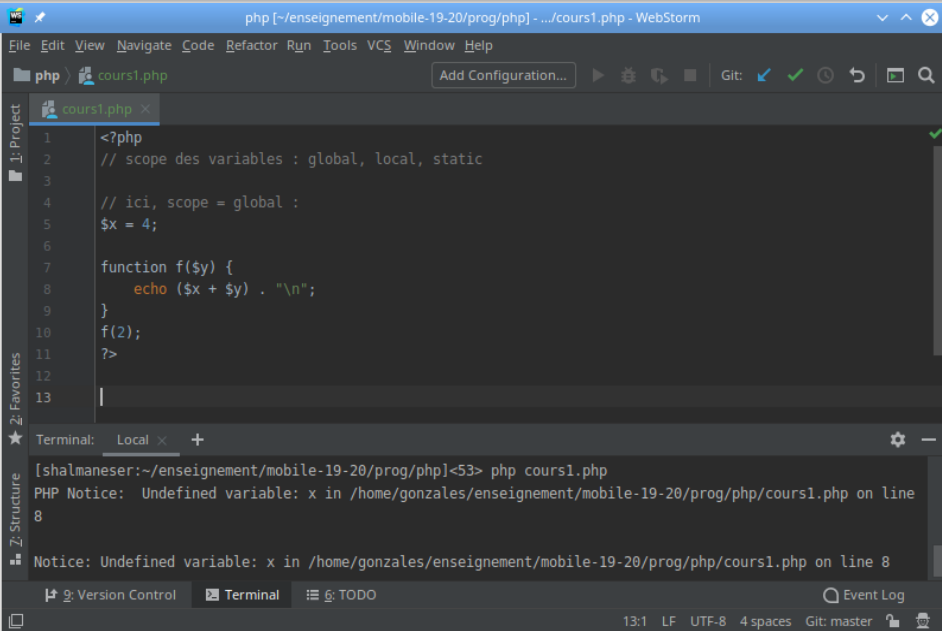
```
1 <?php
2 // déclarer une fonction :
3 function f () {
4     echo "toto\n";
5 }
6 function g($x, $y=4) { // 4 = valeur par défaut de $y
7     return $x + $y;
8 }
9
10 // utiliser une fonction :
11 f();
12 echo g(3,2) . "\n";
13 echo g(3);
14 ?>
```

The terminal at the bottom shows the execution of the script:

```
[shalmaneser:~/enseignement/mobile-19-20/prog/php]<43> php cours1.php
toto
5
7
[shalmaneser:~/enseignement/mobile-19-20/prog/php]<44> 
```

The IDE interface includes a menu bar (File, Edit, View, Navigate, Code, Refactor, Run, Tools, VCS, Window, Help), a toolbar with icons for running, debugging, and Git, and a sidebar with Project, Favorites, and Structure views. The status bar at the bottom shows "14:3 LF UTF-8 4 spaces Git: master".

# La portée des variables (1/4)



The screenshot shows a code editor window with the following content:

```
1 <?php
2 // scope des variables : global, local, static
3
4 // ici, scope = global :
5 $x = 4;
6
7 function f($y) {
8     echo ($x + $y) . "\n";
9 }
10 f(2);
11 ?>
```

The terminal output shows the following error message:

```
[shalmaneser:~/enseignement/mobile-19-20/prog/php]<53> php cours1.php
PHP Notice: Undefined variable: x in /home/gonzales/enseignement/mobile-19-20/prog/php/cours1.php on line 8

Notice: Undefined variable: x in /home/gonzales/enseignement/mobile-19-20/prog/php/cours1.php on line 8
```

# La portée des variables (2/4)

The screenshot shows a code editor window for a file named `scope3.php`. The code defines a function `f($y)` that attempts to access a global variable `$x` and declare a global variable `$z`. The function prints the values of `$x` and `$z` concatenated with `$y`. The terminal output shows the execution of the script, displaying the values of `$x` and `$z` as `4` and `6` respectively.

```
<?php
// ici, scope = global :
$x = 4;
$z = 6;

function f($y) {
    // pour accéder à x, qui n'est pas défini dans la fonction :
    // soit utiliser $GLOBALS['x']
    // soit déclarer global $z;
    global $z;

    echo ($GLOBALS['x'] + $y) . "\n"; // $GLOBALS : tableau des variables globales
    echo ($z + $y) . "\n";
}

f(2);
?>
```

Terminal: Local (2) × +

```
[shalmaneser:~/enseignement/mobile-19-20/prog/php]<1> php scope3.php
6
8
```

# La portée des variables (3/4)

The screenshot shows an IDE window titled "php [~/enseignement/mobile-19-20/prog/php] - .../cours1.php - WebStorm". The editor displays the following PHP code:

```
1 <?php
2 // scope des variables : global, local, static
3 $x = 4;
4 function f() {
5     $x = 5; // variable locale à la fonction, différente de la variable globale
6     {
7         $x = 6; // même variable qu'au dessus
8     }
9     echo $x . "\n";
10 }
11 f();
12 echo $x;
13 ?>
14 |
```

The terminal output shows the execution of the script:

```
[shalmaneser:~/enseignement/mobile-19-20/prog/php]<58> php cours1.php
6
4
[shalmaneser:~/enseignement/mobile-19-20/prog/php]<59> |
```

The IDE interface includes a menu bar (File, Edit, View, Navigate, Code, Refactor, Run, Tools, VCS, Window, Help), a toolbar with icons for running, debugging, and Git, and a bottom status bar showing "14:1 LF UTF-8 4 spaces Git: master".

# La portée des variables (4/4)

The screenshot shows an IDE window titled "php [~/enseignement/mobile-19-20/prog/php] - .../cours1.php - WebStorm". The editor displays the following PHP code:

```
1 <?php
2 // scope des variables : global, local, static
3 $x = 4;
4 function f() {
5     static $x = 5; // variable statique
6     $x ++;
7     echo $x . "\n";
8 }
9
10 f();
11 f();
12 ?>
13 |
```

Below the editor is a terminal window with the following output:

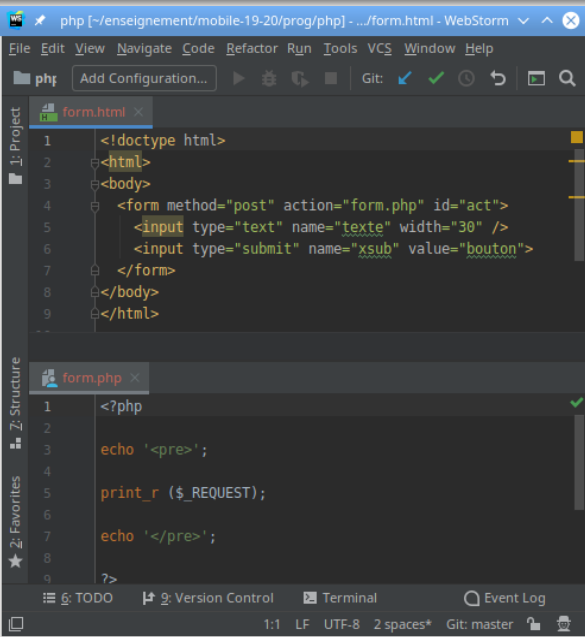
```
Terminal: Local x +
[shalmaneser:~/enseignement/mobile-19-20/prog/php]<61> php cours1.php
6
7
[shalmaneser:~/enseignement/mobile-19-20/prog/php]<62> |
```

The IDE interface includes a menu bar (File, Edit, View, Navigate, Code, Refactor, Run, Tools, VCS, Window, Help), a toolbar with icons for running, debugging, and Git, and a sidebar with Project, Favorites, and Structure views. The status bar at the bottom shows "13:1 LF UTF-8 4 spaces Git: master".

# Les variables superglobales

- ▶ **Superglobal** : accessible dans tous les scopes
- ▶ Ce sont des tableaux (`$GLOBALS [ 'x' ]`)
- ▶ Variables contenant les infos transmises par l'utilisateur au serveur web :
  - ▶ `$_POST`
  - ▶ `$_GET`
  - ▶ `$_REQUEST = $_POST + $_GET`
- ▶ Variables contenant les infos sur le serveur web :
  - ▶ `$_SERVER`
- ▶ Variables contenant les infos sur les fichiers uploadés :
  - ▶ `$_FILES`
- ▶ Variables de session et cookies :
  - ▶ `$_SESSION`
  - ▶ `$_COOKIE`

# Forms HTML et variables superglobales PHP



The screenshot shows an IDE window with two tabs: 'form.html' and 'form.php'. The 'form.html' tab contains the following HTML code:

```
1 <!doctype html>
2 <html>
3 <body>
4 <form method="post" action="form.php" id="act">
5 <input type="text" name="texte" width="30" />
6 <input type="submit" name="xsub" value="bouton">
7 </form>
8 </body>
9 </html>
```

The 'form.php' tab contains the following PHP code:

```
1 <?php
2
3 echo '<pre>';
4
5 print_r ($_REQUEST);
6
7 echo '</pre>';
8
9 ?>
```

sss      bouton

```
Array
(
    [texte] => sss
    [xsub] => bouton
)
```



# Backend PHP et transfert d'informations en JSON

The image displays two overlapping windows. The background window is Postman, showing a REST client request configuration. The foreground window is a code editor showing PHP code that generates a JSON response.

**Postman Request Configuration:**

- Method: POST
- URL: 127.0.0.1/mobile/cours/json.php
- Environment: No Environment
- Body: none
- Body Type: JSON
- Body Content:

```
1 {
2   "cle1": "xxx",
3   "cle2": "yyy"
4 }
```

**PHP Code (php - json.php):**

```
1 <?php
2 header('header: 'Content-type:application/json;charset=utf8');
3
4 echo json_encode([
5   'cle1' => 'xxx',
6   'cle2' => 'yyy'
7 ]);
8
9 >
```

# Les boucles

The screenshot shows an IDE window titled "php [~/enseignement/mobile-19-20/prog/php] - .../cours1.php - WebStorm". The code in "cours1.php" is as follows:

```
1 <?php
2 $tab = array ('x', 'y', 'z');
3
4 // boucle pour itérer sur les éléments de $tab :
5 foreach ($tab as $val) { echo "$val "; }
6 echo "\n";
7 foreach ($tab as $key => $val) { echo "$key = $val "; }
8 echo "\n";
9
10 // boucle à la C :
11 for ($i = 0; $i < count($tab); $i++) { echo "${tab[$i]} "; }
12 ?>
13
```

The terminal output shows the execution of the script:

```
[shalmaneser:~/enseignement/mobile-19-20/prog/php]<66> php cours1.php
x y z
0 = x 1 = y 2 = z
x y z
[shalmaneser:~/enseignement/mobile-19-20/prog/php]<67> █
```

The IDE interface includes a menu bar (File, Edit, View, Navigate, Code, Refactor, Run, Tools, VCS, Window, Help), a toolbar with "Add Configuration...", and a sidebar with "1: Project", "2: Favorites", and "3: Structure". The bottom status bar shows "All files are up-to-date (14 minutes ago)", "13:1 LF UTF-8 4 spaces Git: master", and "1 Event Log".

# Les alternatives

The screenshot shows a code editor window with the following content:

```
1 <?php
2 $x = 3;
3
4 // if then else :
5 if ( $x == 4 ) echo "x vaut 4\n";
6 elseif ($x == 3) echo "x vaut 3\n";
7 else echo "x est différent\n";
8
9 switch ($x) { // switch : similaire au langage C
10     case 3:
11         echo "x = 3"; break;
12     default:
13         echo "x != 3";
14 }
15 ?>
```

The terminal output shows the execution of the script:

```
[shalmaneser:~/enseignement/mobile-19-20/prog/php]<69> php cours1.php
x vaut 3
x = 3
[shalmaneser:~/enseignement/mobile-19-20/prog/php]<70> 
```

The interface includes a menu bar (File, Edit, View, Navigate, Code, Refactor, Run, Tools, VCS, Window, Help), a toolbar with an "Add Configuration..." button and icons for Git, Run, and other actions, and a sidebar with "Project", "Favorites", and "Structure" views.

# Les importations

The screenshot shows an IDE window titled "php [-/enseignement/mobile-19-20/prog/php] - .../cours1.php - WebStorm". The main editor displays the content of "cours1.php":

```
1 <?php
2 // importation d'un script php dans un autre :
3 // require_once : importe exactement 1 fois
4 require_once 'myscript.php';
5 echo "fin\n";
6 require_once 'myscript.php';
7 ?>
```

Below it, the content of "myscript.php" is shown:

```
1 <?php
2 echo "je suis dans myscript\n";
3 ?>
```

The terminal at the bottom shows the execution of the script:

```
[shalmaneser:~/enseignement/mobile-19-20/prog/php]<3> php cours1.php
je suis dans myscript
fin

[shalmaneser:~/enseignement/mobile-19-20/prog/php]<4> |
```

The IDE interface includes a menu bar (File, Edit, View, Navigate, Code, Refactor, Run, Tools, VCS, Window, Help), a toolbar with an "Add Configuration..." button, and a sidebar with "Project" and "Structure" views. The status bar at the bottom indicates "WebStorm 2019.3.2 available: // Update... (today 8:49 AM)", "9:1 LF UTF-8 4 spaces", and "Git: master".

Requêtes MySQL  $\implies$  utiliser un « *database abstraction layer* »

► ici : utilisation de PDO (PHP Data Objects)

► **Avantages :**

- sécurité (« *prepared statements* »)
- facilité d'utilisation (« *helpers* »)
- réutilisabilité (« *API unifiée de bases de données* »)
- applicabilité (« *compatible avec de nombreuses bases : MySQL, SQLite, Firebird, Oracle, etc.* »)

# Petit aperçu de PDO : création d'une instance

```
<?php
// creation de l'instance PDO et connexion à la BD
$dsn = "mysql:host=$mysqlHost;" .
       "dbname=$mysqlDatabase;" .
       "charset=$charset";

// les options
$opt = [
    PDO::ATTR_ERRMODE => PDO::ERRMODE_EXCEPTION,
    PDO::ATTR_DEFAULT_FETCH_MODE => PDO::FETCH_ASSOC,
    PDO::ATTR_EMULATE_PREPARES => false
];

$PDO = new PDO($dsn, $mysqlLogin, $mysqlPassword, $opt);
?>
```

# Petit aperçu de PDO : utilisation de l'instance

```
<?php
```

```
// définition de la requête
```

```
$query = "SELECT * FROM $mysqlTable ".  
         "WHERE field1=? AND field2 IN (?,?)";  
$data = ['f1', 3, 5];
```

```
// envoi et exécution de la requête à la base
```

```
$statement = $PDO->prepare($query); // préparation  
$exec = $statement->execute($data); // exécution
```

```
// récupération du résultat
```

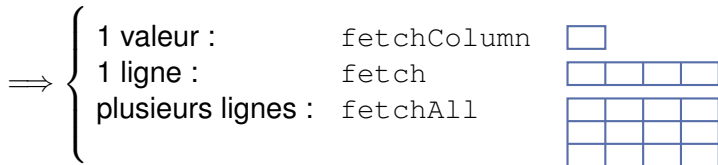
```
$resultats = $statement->fetchAll(PDO::FETCH_ASSOC);
```

```
// affichages
```

```
print_r($resultats);
```

```
?>
```

## ► Récupérer des données de la BD



## ► Options pour fetch et fetchAll :

- PDO::FETCH\_NUM : **indices = nombres**
- PDO::FETCH\_ASSOC : **indices = noms des champs dans la BD**

PDO::FETCH\_NUM :

```
array (  
  0 => 'toto',  
  1 => 'titi'  
);
```

PDO::FETCH\_ASSOC :

```
array (  
  'nom' => 'toto',  
  'prenom' => 'titi'  
);
```



- ▶ **Documentation :**

<https://www.php.net/manual/fr/funcref.php>

- ▶ **Visualisation des erreurs :**

En début de programme :

```
ini_set('display_errors', 'On');  
error_reporting(E_ALL);
```