Résumé du PHP

Variables en PHP

Les variables sont comme des contenants qui portent un nom. Un seul contenu est acceptable par contenant. Lors de la programmation on manipule des données. Ces données doivent toujours être déposées dans un contenant une fois le traitement terminé. En PHP, les variables sont désignées par le signe de $.

J'aime à les comparer à des contenants tels que les bols en cuisine qui permettent de maintenir temporairement des contenus destinés à des opérations de cuisine subséquentes.

Remplir une variable (opération d'affectation)

Pour déposer un contenu dans un contenant, il faut utiliser l'opérateur d'affectation qui est le =, et le transfert de données s'effectue TOUJOURS de droite à gauche. En plaçant un nouveau contenu dans le contenant, l'ancien contenu est effacé et disparait.

Par exemple ici on place une chaîne de caractères dans une variable que l'on appelle $sql.

```
$sql = "SELECT      FROM province";
```

Ici on place le résultat d'une opération mathématique (appel de fonction) dans une variable appelé $nombre_arrondi.

```
$nombre_arrondi = round(  .  );
```

Une fois que une donnée est dans un contenant, il est possible de la réutiliser plus tard dans des opérations ou dans des fonctions.

Utilisation des variables

Voici un exemple d'utilisation de la données dans une **opération**, prenez note de la mémorisation du résultat dans l'autre variable.

```
$total = $premier_nombre + $nombre_arrondi;
```

Enfin il est également possible d'utiliser une donnée provenant d'une variable dans un appel de fonction, ce qui équivaut à déverser notre contenu à notre 'travailleur spécialisé' comme ceci:

```
$resultat =    floor($total);
```

Prenez encore note de la mémorisation du résultat dans l'autre variable.

Affichage des variables

Pour afficher une variable en PHP, on peut simplement faire :

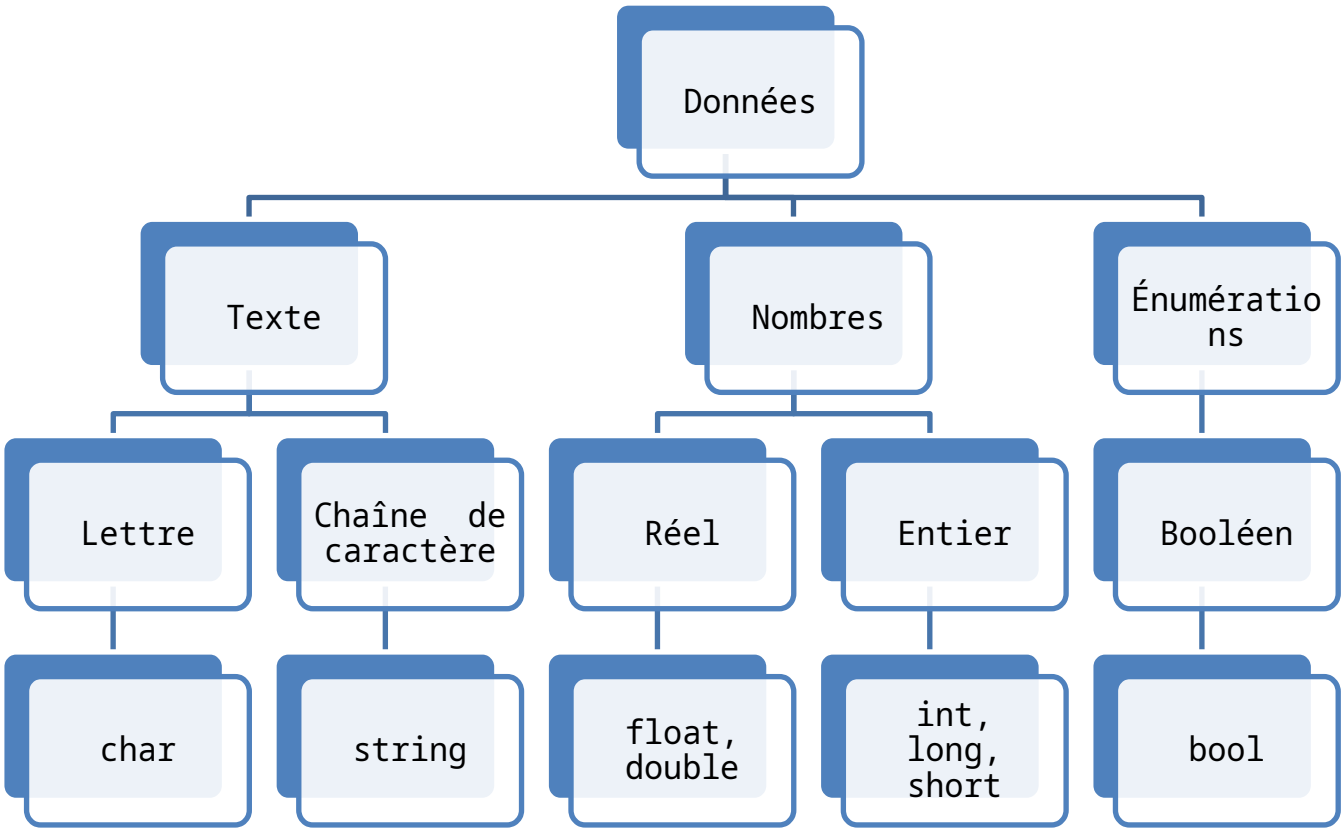
```
echo $sql;
```

Cependant, pour que cela fonctionne, il faut que ce soit une variable simple et non pas un tableau.
Enfin, il existe aussi, à travers le HTML, la notation courte (short tag notation) qui remplace le echo par =

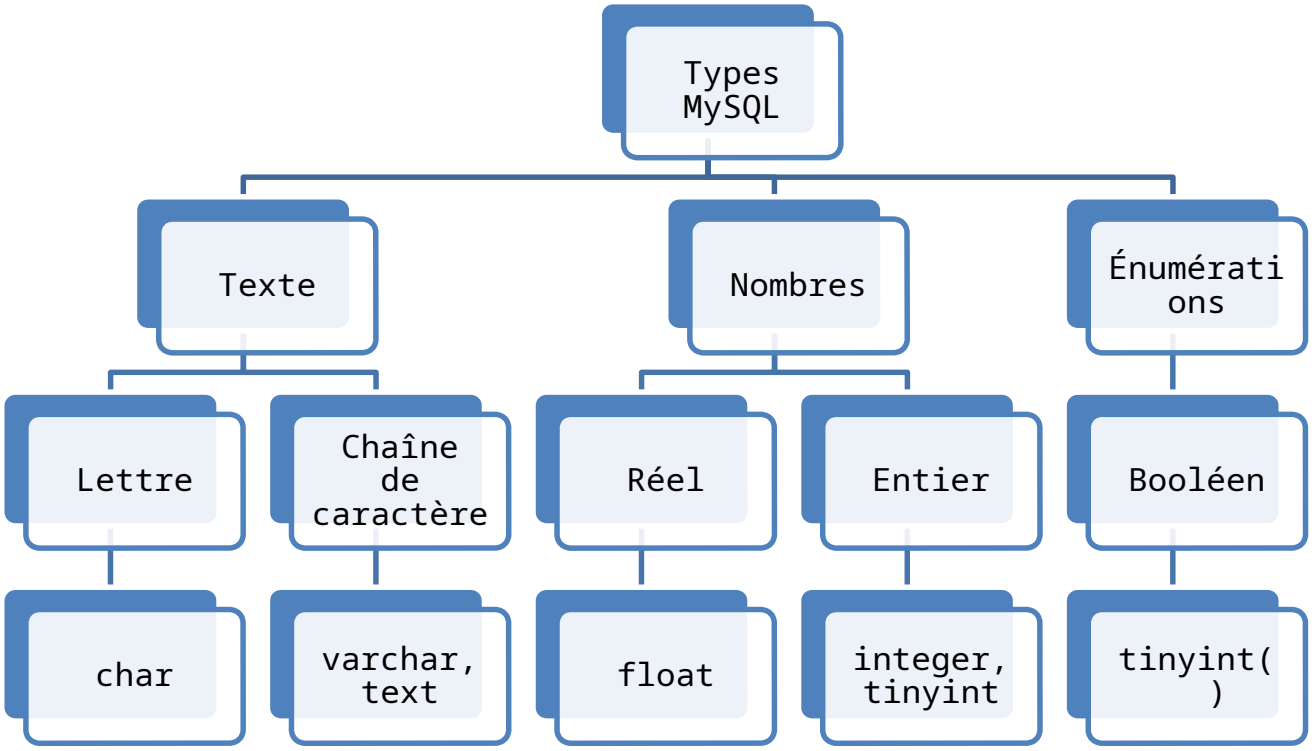
```
<?=$sql?>
```

Les sortes de variables

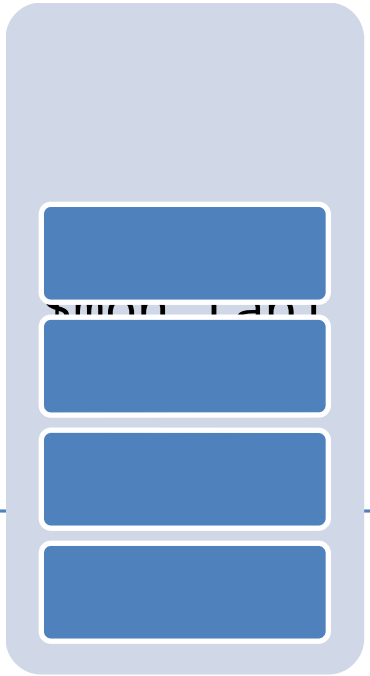
Bien que le langage PHP ne demande pas de préciser quelle sorte de données on dépose dans nos contenant (variables), il se trouve qu'il existe différents types de données, comme dans la base de données. Il est utile de les connaître afin de pouvoir lire le manuel PHP pour l'utilisation des fonctions.



Parallèle avec MySQL



Les variables en tableau



Nous avons dit dans le chapitre précédent que une variable ne contient qu'une seule donnée. Cependant il existe une autre structure, appelé tableau (array en anglais) qui permet d'avoir plusieurs espaces mémoire dans une seule variable.

Cependant, avec ce type de structure, on ne peut plus accéder à la variable simplement en donnant son nom, il faut également indiquer la position dans le tableau, celles-ci sont numérotées à partir de .

Lecture d'un élément du tableau

N'importe où dans le code il est possible de lire un élément du tableau en utilisant son nom et l'indice de sa position, par exemple:

```
$mon_tableau[  ]
```

Affichage d'un élément du tableau

L'affichage d'un élément est simplement le fait de fournir sa coordonnées à la commande echo comme ceci:

```
echo  $mon_tableau[  ];
```

Ce qui affichera à l'écran.

Pour afficher tous les éléments du tableau d'un coup, pour tester, on peut utiliser la fonction print_r( ) ainsi:

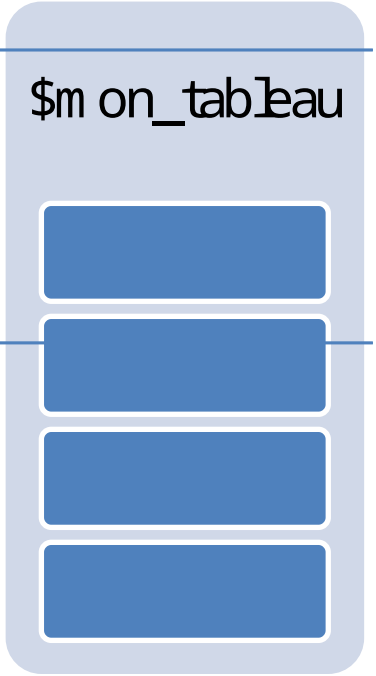
```
print_r($mon_tableau) ;
```

Opération sur un élément du tableau

Je peux pratiquer des opérations sur les éléments du tableau en les utilisant comme des variables normales. Par exemple (prenez note dans cet exemple du fait que je récupère mon résultat IMMÉDIATEMENT dans une variable appelée $somme):

```
$somme = $mon_tableau[ ] + $mon_tableau[ ] + $mon_tableau[ ] + $mon_tableau[ ];
```

Écriture dans un tableau




```
$mon_tableau[ ] = ;
```

Ceci efface l'ancienne valeur de la position et inscrit la nouvelle valeur, un dans cet espace mémoire.

Les tableaux dans le manuel PHP

Il arrive que le manuel PHP désigne pour une fonction le type de variable attendu ou le type de variable retourné comme étant **array**. Il s'agit simplement d'un tableau.

Il faut lire la documentation de la fonction plus en détail pour savoir quel genre de contenu est attendu.

Tableaux associatifs

Les tableaux traditionnels ont des indices tels que , , ,.... Cependant il existe une nouvelle sorte de tableau où les indices des données sont plutôt des chaînes de caractères. Ce sont les tableaux associatifs. Des synonymes sont hashtable et map.

Par exemple, voici un tableau associatif dans lequel, au lieu de classer les données sous forme de position , , et , j'ai plutôt utilisé des noms pour les positions. Par exemple dans un menu, je souhaite classer les éléments du menu selon "entrée", "plat", "breuvage" et "dessert".

Lecture dans un tableau associatif

Pour lire une donnée dans un tableau associatif, c
tableau régulier on utilise les crochets carrés et
l'indice entre ces crochets. La seule différence c
que l'indice soit un chiffre, l'indice est un texte entre
guillemets. Donc cela donne:

```
$mon_menu["dessert"]
```

Affichage d'un élément d'un tableau associatif

Il est possible de faire un affichage d'un élément du tableau, par exemple pour afficher le breuvage de ce menu on ferait ceci:

```
echo $mon_menu["breuvage"];
```

Opérations sur un élément de tableau associatif

On pourrait vouloir construire une chaîne de caractère contenant entre autre choses des valeurs provenant de notre tableau, avec l'opération de **concaténation** (opérateur .), comme ceci:

```
$mon_menu["entree"] = "soupe";
$mon_menu["plat"] = "poulet";
$mon_menu["breuvage"] = "café";
$mon_menu["dessert"] = "tarte";

$commande = "Je désire commander un " . $mon_menu["entree"] . " en entrée. Comme plat principal je
          prendrai " . $mon_menu["plat"] . " accompagné d'un " .
          $_mon_menu["breuvage"] . ". Enfin, pour dessert je
          souhaite avoir un(e) " . $mon_menu["dessert"] . ". Merci
          beaucoup !";

echo $commande ;
```

Écriture dans un tableau associatif

Il est possible, comme dans n'importe quelle variable ou tableau, d'écrire dans un tableau associatif de la manière suivante:

```
$mon_menu["dessert"] = "gateau";
```

Des tableaux associatifs célèbres

$_GET et $_POST sont des tableaux associatifs déjà créés pour nous et qui contiennent déjà certaines données de la manière suivante:

- $_GET contient les variables qui ont été passées en paramètre à notre script via l'URL de la manière suivante: detail-pays.php?id_pays=
- $_POST contient les valeurs des champs remplis dans un formulaire. Le nom des input, textarea et select devient le nom de cette variable dans le tableau $_POST.

Fonction
- Travailleur spécialisé
    - Données à leur remettre
    - Résultat à recevoir

Utilisation de fonctions

Une fonction est comme un travailleur spécialisé, à qui, on doit souvent donner des données à travailler et aussi récupérer le résultat du travail (dans un contenant, donc une variable).

→Une fonction s'utilise avec les parenthèses

QUESTION : Comment puis-je savoir si je dois donner des données à ma fonction et si je dois récupérer le résultat ?

→Il faut consulter le manuel (site php.net) sur la page de documentation de la fonction

→On y retrouve l'information concernant la liste des données à envoyer et leur type (string, float ou autre)

→On y retrouve aussi l'information sur la nature des informations retournées, mais la décision finale de récupérer ou non le résultat de l'appel d'une fonction est selon l'usage qu'on souhaite en faire plus tard dans le programme. Si j'en ai besoin, je le garde en mémoire.

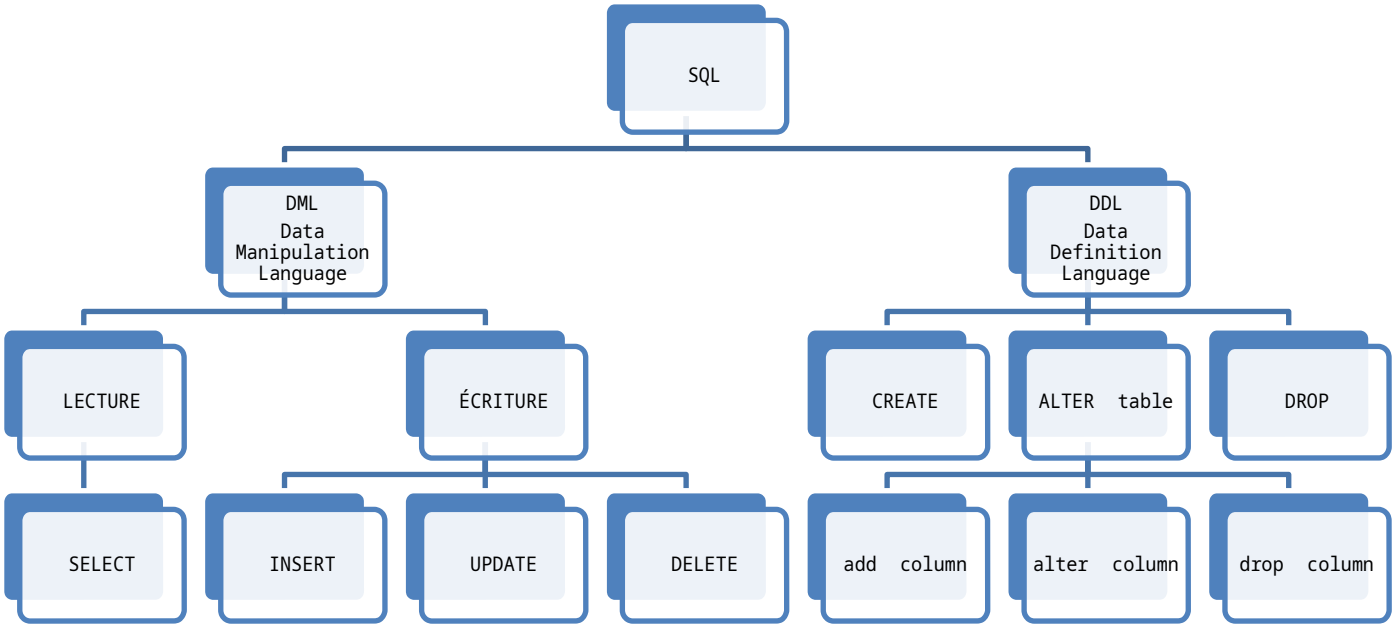
Exemples d'appels de fonction

```
$resultat = round($donnee) ;

$resultat = mysql_query("SELECT FROM province") ;
mysql_query("INSERT INTO province(Name,Country)
VALUES('blabla','Canada')") ;
```

Révision SQL

Les types de commandes



SELECT

Syntaxe du SELECT

```
SELECT  liste_des_colonnes
FROM  table
WHERE  condition_qui_discrimine_les_lignes
```

Utilisation du SELECT

Pour écrire une requête SELECT, il faut connaître la structure de la table de données, et tout spécialement le nom de la table et le nom des colonnes.

Par exemple, pour sélectionner le nom de toute les provinces, dans la base de données mondiale, on fait comme ceci:

```
SELECT  Name  FROM  province
```

Pour sélectionner toutes les données de la province Albania, on fait comme ceci:

```
SELECT      FROM  province  WHERE  Name  =  'Albania'
```

➔On utilise souvent SELECT dans les pages web spécialement lors de la réception par GET d'un paramètre

Les conditions en SQL (versus PHP)

Type de comparaison ou combinaison	Symbole utilisé en SQL	Parallèle avec le PHP
Comparaison d'égalité	=	==
Comparaison d'inégalité	<>	!=
Comparaison de plus petit ou plus grand	<= < >= >	<= < >= >
Comparaison de chaînes avec %	LIKE	preg_match()
Combinaison ET	AND	&&
Combinaison OU	OR	||

➔ Les conditions SQL s'utilisent à la suite d'un WHERE dans les commandes SELECT, UPDATE et DELETE.

INSERT

```
INSERT INTO province(Name,Country,Population,Area,Capital)
VALUES('Larive','Canada','      ','          ','Centre-Rive')
```

➔ On utilise souvent INSERT dans les pages web spécialement lors de la réception par POST de nouvelles données (mais pas toujours, par exemple un formulaire de recherche ne mène pas à un INSERT mais à un SELECT)

UPDATE

Update peut mettre à jour un ou plusieurs champs sur une ou plusieurs rangées. Lors de l'utilisation d'un formulaire de modification, on met à jour tous les champs sur une seule rangée. Lors de corrections sur la base de données, comme lorsque en classe on a changé CND pour Canada, on a changé un seul champs sur toute les rangées.

Pour changer tous les champs sur une seule rangée:

```
UPDATE province SET Name = 'Larive', Country = 'Québec', Population
= '          ', Area = '           ', Capital = 'Larive' WHERE Name
= 'Larive'
```

Pour changer un seul champs mais sur toute les rangées, bien on nomme un seul champs et on n'inclue pas de WHERE. Si on souhaite changer un seul champs mais sur plusieurs rangées, on inclue un WHERE qui va sélectionner ces plusieurs rangées.

```
UPDATE province SET Country = 'Canada' WHERE Country = 'CND'
```

DELETE

Effacer sert à supprimer, de manière définitive et irréversible, une ou plusieurs rangées d'une table MySQL. La condition sert à sélectionner quelles rangées seront supprimées.

```
DELETE FROM province WHERE Name = 'Larive'
```

Révision WEB

Dans un site web dynamique (c'est ainsi qu'on appelle les sites web avec des bases de données), dont les pages sont dynamique (c'est à dire qu'on a un seul script pour faire apparaître plusieurs pages différentes), donc dans un site web dynamique on a généralement des listes de données cliquables.

Prenons l'exemple d'un site sur le cinéma dans lequel on a une liste de films cliquables et lorsqu'on clic sur un film on va sur une page détaillée qui présente toutes les informations de ce film. On imagine également que si tous ces films sont présentés sur la page publique, il y a une personne qui les a entré dans la base de données. Nous avons deux sortes d'utilisateurs:

- le visiteur du site web public qui va voir les données et cliquer pour en savoir plus
- l'administrateur du site web qui va vouloir ajouter des films, les modifier et les effacer via le panneau admin

La partie publique du site – atelier sur $_GET

Dans l'atelier sur le GET on avait une situation ou l'on programmait un site web avec une liste de données, chacune d'elle étant cliquable pour amener vers sa page détaillée. Seulement deux scritps sont programmés pour créer ce site:

- un script qui présente la liste de toutes les provinces
- un script qui présente, pour une province indiquée en paramètre, toutes les informations détaillées

SELECT Name FROM province

Stoqué dans une variable comme $sql

MYSQL

SQL envoyé à mysql_query()
on récupère le résultat de mysql_query dans une variable
mysql_fetch_object() en boucle pour aller chercher toutes les provinces une par une

Dans l'hyperlien en HTML, on affiche le nom de la province

Entre les balises A
dans le href...

Page de la

SELECT FROM province WHERE Name = 'Albania'

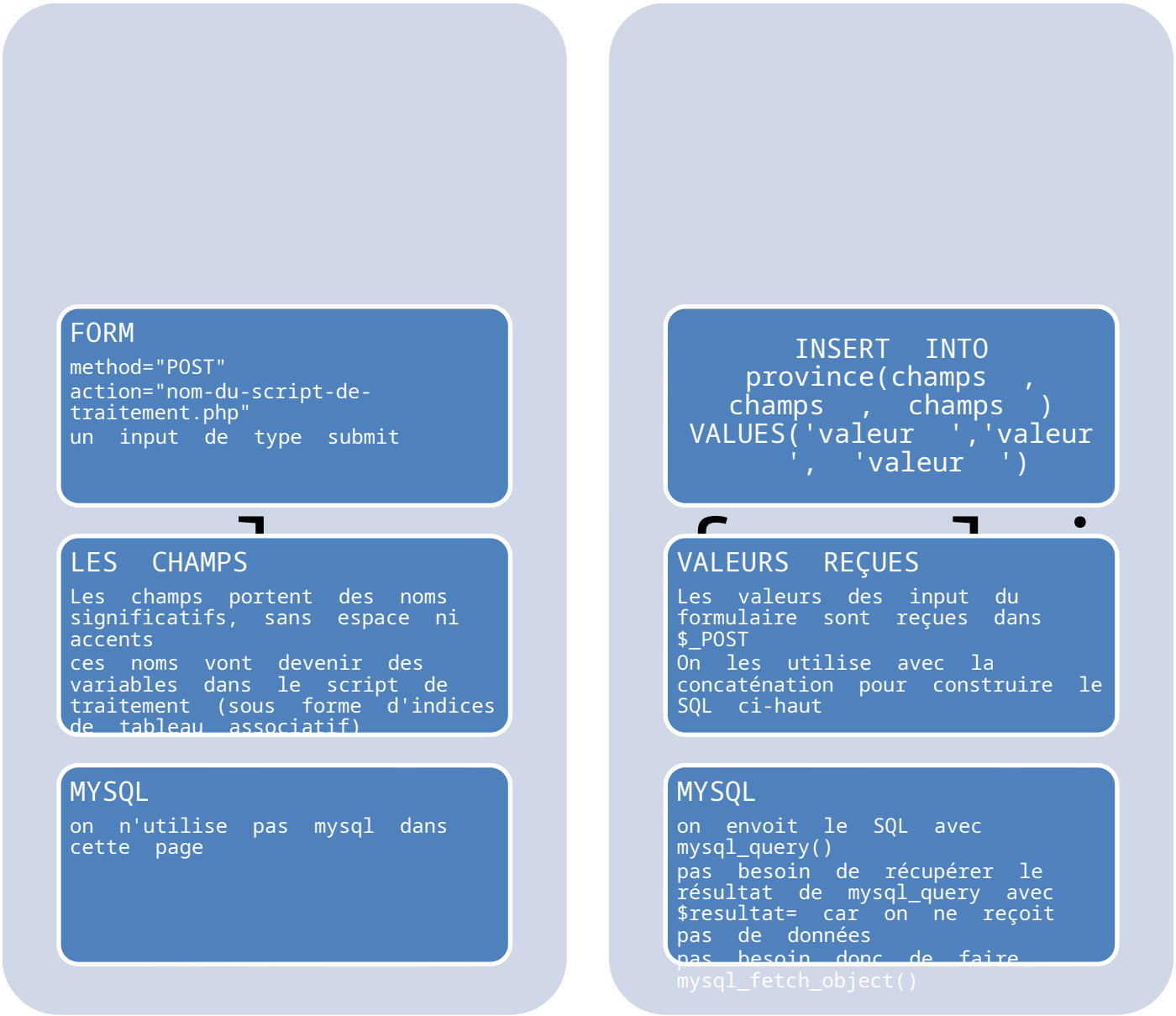
DONNÉES reçues

Les paramètres de l'URL sont reçus dans $_GET
On utilise la valeurs dans $_GET par concaténation pour construire le SQL ci-haut.

MYSQL

Envoyé à mysql_query()
On récupère le résultat de mysql_query dans une variable
Pas de boucle avec mysql_fetch_object( ) car un seul résultat attendu !

La partie admin du site – atelier sur $_POST



Fichier formulaire-film.html

```
<form action="traitement-film.php" meth

                <input type="text" name="
                <input type="text" name="
                <input type="text" name="
    <textarea name="resume"></textarea>

                <input type="submit"/>

</form>
```

Titre:	Avatar
Catégorie:	Science-Fiction
Réalisateur:	Cameroon
Résumé:	blablabla

Soumettre

Fichier traitement-film.php

```
$sql = "INSERT INTO movie(title, category, conceptor, excerpt)
VALUES('".
        $_POST["titre"] ."','  ". $_POST["categorie"] ."','
    ". $_POST["realisateur"] ."','  ".
    $_POST["resume"] ."')";

echo $sql ;
```

affiche

INSERT INTO movie(title, category,conceptor,exerpt)
VALUES('Avatar','Science-Fiction','Cameroon','blablabla')