

VARIABLES

- commençer par \$ - Pas de déclaration

Affectation : \$age = 21 + \$x ou \$nom = "Jean". \$y

Afficher contenu : - echo \$age ; - echo "Je {\$nom} et {\$age}";

Despécialisation : \ - echo 'Je'. \$nom. ' et ' \$age. ' ;

Passer des variables par l'URL : http://... tmc.php? nom = Moi & age = 18

tmc.php récupérera : \$_GET['nom'] → "Moi"
\$_GET['age'] → 18

TABLEAUX

Séquentiel :

indice : entiers (1^{er} case = 0)

affectation : \$t[0] = "france";
ou : \$t = array("france", "italie", ...);

ajout valeur fin tableau : \$t[] = "Russie";

taille tableau : count(\$t);

Parcourir TOUT un tableau :

```
foreach ($t as $elem) {
    ...
}
```

Associatifs

- indice : chaîne de caractères

- comple : clé ⇒ valeur

- clés : unique

affectation : \$t['fr'] = "france";

ou : \$t = array('fr' => "france", ...);

Parcourir tout le tableau :

```
foreach ($t as $cle => $val) {
    ...
}
```

Exemple de tableau associatif : \$_GET
\$_POST

FONCTIONS

Écriture :

Appel : <nom> (\$y);

Appel avec return :

\$b = <nom> (\$y);

```
function <nom> ($y) {
    <instructions>
    return $a;
}
```

\$y n'est pas obligatoire et
il peut y avoir plusieurs séparés
par une virgule

Inclure un autre .php : include("../autre.php");

Fonction header : rediriger vers une autre page web (avec ou sans paramètres)

Ex : header('Location : tmc.php? nom = moi');
exit();

FORMULAIRE

Ex : <form method='post' action='script.php'>
... <input name='qui' ... /> ... </form>

Script .php récupérera le tableau

\$cle => \$val avec \$cle = ce que contient le name et \$val ce qui est entré
dans le formulaire par l'utilisateur

SESSION

\$_SESSION : tableau associatif permettant de partager
des variables sur tout le site.

À initialiser au tout début avec : session_start(); (si utilisée)

Fermer la session : session_destroy(); (sinon détruite automatiquement après un certain temps
ou lors de la fermeture de la navigation)

COOKIES - fichier laissé sur le disque du client

- durée de vie limitée mais reste même si le navigateur est fermé.

- création: `setcookie(<nom>, <val>, <mot>)` (À mettre au tout début, avant le HTML)

- `<nom>, <val>`: nom et valeur du cookie

- `<mot>`: date de la mort du cookie (`time() + <nb secondes>`)

- Création d'une entrée (`<nom>, <val>`) dans le tableau associatif

`$_COOKIE`

(rechargement de la page)

- Mise à jour de `$_COOKIE` qui se chargeant de l'en-tête HTTP

MySQL Accès à la base - Dans un `base.php` qui sera inclus:

```
try { $bd = new PDO('mysql:host='. $server .'; dbname='. $base, $user, $password);  
    $bd->exec('SET NAMES utf8'); }  
catch (Exception $e) { die ("Erreur..."); }
```

REQUÊTES - Étapes à suivre pour exécuter une requête SQL

`$sql = "...";` mettre la requête SQL ici

`$data` est un tableau avec

`$req = $bd->prepare($sql);`

`n` cases pour les `n` enregistrements récupérés. Une case est un

`$req->execute();`

tableau associatif `nom du champ => valeur du champ`.

`$data = $req->fetchall();`

`$req->closeCursor();`

Faire de la sorte uniquement quand la requête SQL ne contient pas de valeur saisie dans un formulaire.

RAPPELS SQL

MàJ: `UPDATE nonTable`

`SET nonChamp = 'newVal'`

`WHERE champ X = Y`

Interrogation: `SELECT champ`

`SELECT *`
pour tout prendre
sinon séparé par
des virgules

`FROM nonTable`

`WHERE champ X = Y`

Ajouter des enregistrements:

```
INSERT INTO client (prenom, nom)  
VALUES ('Jean', 'Moulin'), ('Julie', 'Vie')
```

client est le nom de la table. `prenom` et `nom` sont des champs de la table client. Ici 2 enregistrements seront ajoutés

Supprimer des enregistrements:

`DELETE FROM nonTable`

`WHERE ID=3`

Sous le `WHERE`, toutes les lignes sont supprimées. Supprimer une table:

`DROP TABLE nonTable`

REQUÊTES SQL ANTI-INJECTION

`$sql = "SELECT * FROM AVION WHERE Loc=:ville";`

`$req = $bd->prepare($sql);`

`$marqueurs = array('ville' => $_POST['loc']);`

`$req->execute($marqueurs);`

← variable saisie dans un formulaire