

TP — Fondamentaux PHP

Variables • Conditions • Boucles • Fonctions • Formulaires • PDO

6 exercices

Progressif

~2h00

Débutant

Prérequis : avoir PHP installé (via XAMPP, WAMP ou PHP en ligne de commande). Créer un dossier tp_php/ et travailler dans ce dossier. Ouvrir chaque fichier dans le navigateur via http://localhost/tp_php/

Exercice 1 — Variables, types et affichage

Facile

15 min

Objectif

Créer votre premier fichier PHP, déclarer des variables de différents types et les afficher.

Consignes

Créer un fichier ex1.php et écrire le code suivant :

```
<?php
// 1. Déclarez ces variables
$prenom    = "Alice";
$age       = 20;
$moyenne   = 15.5;
$estEtudiant = true;
```

```
// 2. Affichez chaque variable avec echo
echo "Prénom : " . $prenom . "<br>";
echo "Age : " . $age . " ans<br>";
echo "Moyenne : " . $moyenne . "<br>";
echo "Etudiant : " . ($estEtudiant ? "Oui" : "Non") . "<br>";
```

```
// 3. Affichez le type de chaque variable avec gettype()
echo "Type de prenom : " . gettype($prenom) . "<br>";
echo "Type de age : " . gettype($age) . "<br>";
?>
```

Resultat attendu : Prénom : Alice | Age : 20 ans | Moyenne : 15.5 | Etudiant : Oui | Type de prenom : string | Type de age : integer

Pour aller plus loin

- Ajoutez une variable \$ville et affichez-la
- Utilisez var_dump() à la place de echo pour voir plus de détails sur chaque variable

- Essayez de faire une addition entre \$age et \$moyenne et affichez le résultat

```
ex1.php X
php ynov > ex1.php
2
3 $prenom = "Alice";
4 $age = 20;
5 $moyenne = 15.5;
6 $estEtudiant = true;
7 $ville = "Paris";
8
9 echo "Prenom : " . $prenom . "<br>";
10 echo "Age : " . $age . "ans<br>";
11 echo "Moyenne : " . $moyenne . "<br>";
12 echo "Etudiant : " . ($estEtudiant ? "Oui" : "Non") . "<br>";
13 echo "Ville : " . $ville . "<br>";
14
15
16 echo "Type de prenom : " . gettype($prenom) . "<br>";
17 echo "Type de age : " . gettype($age) . "<br>";
18
19 echo "var_dump($prenom)";
20 echo "var_dump($age)";
21 echo "var_dump($moyenne)" . "<br>";
22
23 $addition = $age + $moyenne;
24 echo "Age + Moyenne : " . "$addition";
25
26
```

```
Prenom : Alice
Age : 20ans
Moyenne : 15.5
Etudiant : Oui
Ville : Paris
Type de prenom : string
Type de age : integer
var_dump(Alice)var_dump(20)var_dump(15.5)
Age + Moyenne : 35.5
```

Objectif

Utiliser les structures conditionnelles pour prendre des décisions dans votre code.

Consignes

Créer un fichier ex2.php :

```
<?php
$note = 14;
```

```
// 1. Afficher la mention selon la note
if ($note >= 16) {
    echo "Mention : Très Bien<br>";
} elseif ($note >= 14) {
    echo "Mention : Bien<br>";
} elseif ($note >= 12) {
    echo "Mention : Assez Bien<br>";
} elseif ($note >= 10) {
    echo "Mention : Passable<br>";
} else {
    echo "Mention : Insuffisant<br>";
}
```

```
// 2. Vérifier si la note est valide (entre 0 et 20)
if ($note >= 0 && $note <= 20) {
    echo "Note valide<br>";
} else {
    echo "Note invalide !<br>";
}
```

```
// 3. Switch sur le jour de la semaine
$jour = "lundi";
switch ($jour) {
    case "lundi":
    case "mardi":
    case "mercredi":
    case "jeudi":
    case "vendredi":
        echo "Jour de semaine<br>"; break;
    case "samedi":
    case "dimanche":
        echo "Week-end !<br>"; break;
    default:
        echo "Jour inconnu<br>";
}
?>
```

Resultat attendu : Mention : Bien | Note valide | Jour de semaine

Questions

1. Changez \$note à 8 — quelle mention s'affiche ?

```
ex2.php X
php ynov > ex2.php
1  <?php
2  |
3  $note = 8;
4
5  if ($note >= 16) {
6      echo "Mention : Très Bien<br>";
7  } elseif ($note >= 14) {
8      echo "Mention : Bien<br>";
9  } elseif ($note >= 12) {
10     echo "Mention : Assez Bien<br>";
11 } elseif ($note >= 10) {
12     echo "Mention : Passable<br>";
13 } else {
14     echo "Mention : Insuffisant<br>";
15 }
16
17
18 if ($note >= 0 && $note <= 20) {
19     echo "Note valide<br>";
20 } else {
21     echo "Note invalide !<br>";
22 }
23
24
25 $jour = "lundi";
26 switch ($jour) {
27     case "lundi":
28     case "mardi":
29     case "mercredi":
```

Mention : Insuffisant

Note valide

Jour de semaine

2. Changez \$note à 25 — que se passe-t-il ?

```

<?php
$note = 25;

if ($note >= 16) {
    echo "Mention : Très Bien<br>";
} elseif ($note >= 14) {
    echo "Mention : Bien<br>";
} elseif ($note >= 12) {
    echo "Mention : Assez Bien<br>";
} elseif ($note >= 10) {
    echo "Mention : Passable<br>";
} else {
    echo "Mention : Insuffisant<br>";
}

if ($note >= 0 && $note <= 20) {
    echo "Note valide<br>";
} else {
    echo "Note invalide !<br>";
}

$jour = "lundi";
switch ($jour) {
    case "lundi":
    case "mardi":
    case "mercredi":

```

Mention : Très Bien

Note invalide !

Jour de semaine

3. Changez \$jour à 'samedi' — que s'affiche-t-il ?

ex2.php

php ynov > ex2.php

```
1  <?php
2
3  $note = 25;
4
5  if ($note >= 16) {
6      echo "Mention : Très Bien<br>";
7  } elseif ($note >= 14) {
8      echo "Mention : Bien<br>";
9  } elseif ($note >= 12) {
10     echo "Mention : Assez Bien<br>";
11 } elseif ($note >= 10) {
12     echo "Mention : Passable<br>";
13 } else {
14     echo "Mention : Insuffisant<br>";
15 }
16
17
18 if ($note >= 0 && $note <= 20) {
19     echo "Note valide<br>";
20 } else {
21     echo "Note invalide !<br>";
22 }
23
24
25 $jour = "samedi";
26 switch ($jour) {
27     case "lundi":
28     case "mardi":
29     case "mercredi":
30     case "jeudi":
31     case "vendredi":
```

Mention : Très Bien

Note invalide !

Week-end !

Exercice 3 — Boucles for, while et foreach

Moyen

25 min

Objectif

Maîtriser les trois types de boucles PHP les plus utilisés.

Consignes

Créer un fichier ex3.php :

```
<?php
// PARTIE 1 - Boucle for
// Afficher les nombres de 1 à 10
echo "<h3>Boucle for :</h3>";
for ($i = 1; $i <= 10; $i++) {
    echo $i . " ";
}
echo "<br>";
```

```
// Afficher la table de multiplication de 3
echo "<h3>Table de 3 :</h3>";
for ($i = 1; $i <= 10; $i++) {
    echo "3 x " . $i . " = " . (3 * $i) . "<br>";
}
```

```
// PARTIE 2 - Boucle while
// Compter à rebours de 5 à 0
echo "<h3>Compte à rebours :</h3>";
$compteur = 5;
while ($compteur >= 0) {
    echo $compteur . " ";
    $compteur--;
}
echo "<br>";
```

```
// PARTIE 3 - Boucle foreach sur un tableau
$etudiants = ["Alice", "Bob", "Carla", "David", "Emma"];
echo "<h3>Liste des étudiants :</h3>";
foreach ($etudiants as $etudiant) {
    echo "- " . $etudiant . "<br>";
}
```

```
// Tableau associatif
$notes = ["Alice" => 15.5, "Bob" => 12, "Carla" => 17];
echo "<h3>Notes :</h3>";
foreach ($notes as $nom => $note) {
    echo $nom . " : " . $note . "/20<br>";
}
?>
```

Resultat attendu : Table de 3 de $3 \times 1 = 3$ à $3 \times 10 = 30$ | Compte à rebours : 5 4 3 2 1 0 | Liste des 5 étudiants | Notes de Alice, Bob, Carla

Défi bonus

- Calculez et affichez la moyenne des notes du tableau \$notes

```
$somme = 0;
$notes = ["Alice" => 15.5, "Bob" => 12, "Carla" => 17];
echo "<h3>Notes :</h3>";
foreach ($notes as $nom => $note) {
    echo $nom . " : " . $note . "/20<br>";
    $somme = $somme + $note;
}

$nbnote = count($notes);
$moyenne = $somme / $nbnote;

echo "La moyenne est : " . $moyenne;

?>
```

Boucle for :

1 2 3 4 5 6 7 8 9 10

Table de 3 :

3 x 1 = 3
3 x 2 = 6
3 x 3 = 9
3 x 4 = 12
3 x 5 = 15
3 x 6 = 18
3 x 7 = 21
3 x 8 = 24
3 x 9 = 27
3 x 10 = 30

Compte à rebours :

5 4 3 2 1 0

Liste des étudiants :

- Alice
- Bob
- Carla
- David
- Emma

Notes :

Alice : 15.5/20
Bob : 12/20
Carla : 17/20
La moyenne est : 14.833333333333

- Affichez uniquement les étudiants dont la note est supérieure à 13

```
$somme = 0;
$notes = ["Alice" => 15.5, "Bob" => 12, "Carla" => 17];
echo "<h3>Notes :</h3>";
foreach ($notes as $nom => $note) {
    if ($note > 13) {
        echo $nom . " : " . $note . "/20<br>";
    }
    $somme = $somme + $note;
}

$nbnote = count($notes);
$moyenne = $somme / $nbnote;

echo "La moyenne est : " . $moyenne;

?>
```

Notes :

Alice : 15.5/20

Carla : 17/20

La moyenne est : 14.833333333333

Objectif

Créer et utiliser des fonctions PHP pour éviter la répétition de code.

Consignes

Créer un fichier ex4.php :

```
<?php
// 1. Fonction simple - calculer la moyenne
function calculerMoyenne($notes) {
    $total = 0;
    foreach ($notes as $note) {
        $total += $note;
    }
    return $total / count($notes);
}
```

```
// 2. Fonction avec valeur par défaut
function saluer($prenom, $civilite = "M.") {
    return "Bonjour " . $civilite . " " . $prenom . " !";
}
```

```
// 3. Fonction qui retourne une mention
function getMention($note) {
    if ($note >= 16) return "Tres Bien";
    if ($note >= 14) return "Bien";
    if ($note >= 12) return "Assez Bien";
    if ($note >= 10) return "Passable";
    return "Insuffisant";
}
```

```
// 4. Appel des fonctions
$mesNotes = [14, 16, 12, 18, 10];
$moyenne = calculerMoyenne($mesNotes);
echo "Moyenne : " . $moyenne . "<br>";
echo "Mention : " . getMention($moyenne) . "<br>";
```

```
echo saluer("Alice") . "<br>";
echo saluer("Sophie", "Mme") . "<br>";
?>
```

Resultat attendu : Moyenne : 14 | Mention : Bien | Bonjour M. Alice ! | Bonjour Mme Sophie !

Défi bonus

- Créez une fonction `estAdmis($note)` qui retourne `true` si la note `>= 10`, `false` sinon

- Créez une fonction `afficherTableau($tableau)` qui affiche tous les éléments d'un tableau avec echo

Exercice 5 — Formulaire HTML + traitement PHP

Moyen

30 min

Objectif

Créer un formulaire HTML, récupérer les données avec `$_POST` et les valider en PHP.

Fichier `ex5_form.html`

Créer ce fichier HTML :

```
<!DOCTYPE html>
<html lang="fr">
<head><meta charset="UTF-8"><title>Formulaire étudiant</title></head>
<body>
  <h2>Inscription étudiant</h2>
  <form method="POST" action="ex5_traitement.php">
    <label>Prénom : <input type="text" name="prenom"
required></label><br><br>
    <label>Email : <input type="email" name="email"
required></label><br><br>
    <label>Note : <input type="number" name="note" min="0" max="20"
step="0.5"></label><br><br>
    <button type="submit">Envoyer</button>
  </form>
</body>
</html>
```

Fichier `ex5_traitement.php`

Créer ce fichier PHP pour traiter le formulaire :

```
<?php
// Vérifier que le formulaire a bien été soumis
if ($_SERVER["REQUEST_METHOD"] === "POST") {
```

```
  // Récupérer et sécuriser les données
  $prenom = htmlspecialchars(trim($_POST["prenom"]));
  $email = filter_var($_POST["email"], FILTER_VALIDATE_EMAIL);
  $note = (float) $_POST["note"];
```

```
  // Validation
  $erreurs = [];
```

```
  if (empty($prenom)) {
    $erreurs[] = "Le prénom est obligatoire";
  }
  if ($email === false) {
    $erreurs[] = "Email invalide";
  }
  if ($note < 0 || $note > 20) {
```

```
$erreurs[] = "Note invalide (doit etre entre 0 et 20)";  
}
```

```
// Afficher les erreurs ou le résultat  
if (!empty($erreurs)) {  
    foreach ($erreurs as $erreur) {  
        echo "<p style='color:red'>" . $erreur . "</p>";  
    }  
} else {  
    $mention = ($note >= 16) ? "Tres Bien" : ($note >= 14 ? "Bien" :  
        ($note >= 12 ? "Assez Bien" : ($note >= 10 ? "Passable"  
: "Insuffisant")));
```

```
echo "<h2>Inscription validée !</h2>";  
echo "<p>Prénom : <strong>" . $prenom . "</strong></p>";  
echo "<p>Email : <strong>" . $email . "</strong></p>";  
echo "<p>Note : <strong>" . $note . "/20</strong></p>";  
echo "<p>Mention : <strong>" . $mention . "</strong></p>";  
}  
?>
```

Resultat attendu : Après soumission : affichage du prénom, email, note et mention. Si le formulaire est incomplet : messages d'erreur en rouge.

Attention : Ne jamais afficher directement \$_POST['prenom'] sans htmlspecialchars() — risque d'attaque XSS !

Inscription étudiant

Prénom :

Email :

Note :

Inscription validée !

Prénom : **dylan**

Email : **dylan.spinelli@gmail.com**

Note : **15/20**

Mention : **Bien**

Objectif

Connecter PHP à une base de données MySQL, insérer et lire des données de façon sécurisée avec PDO.

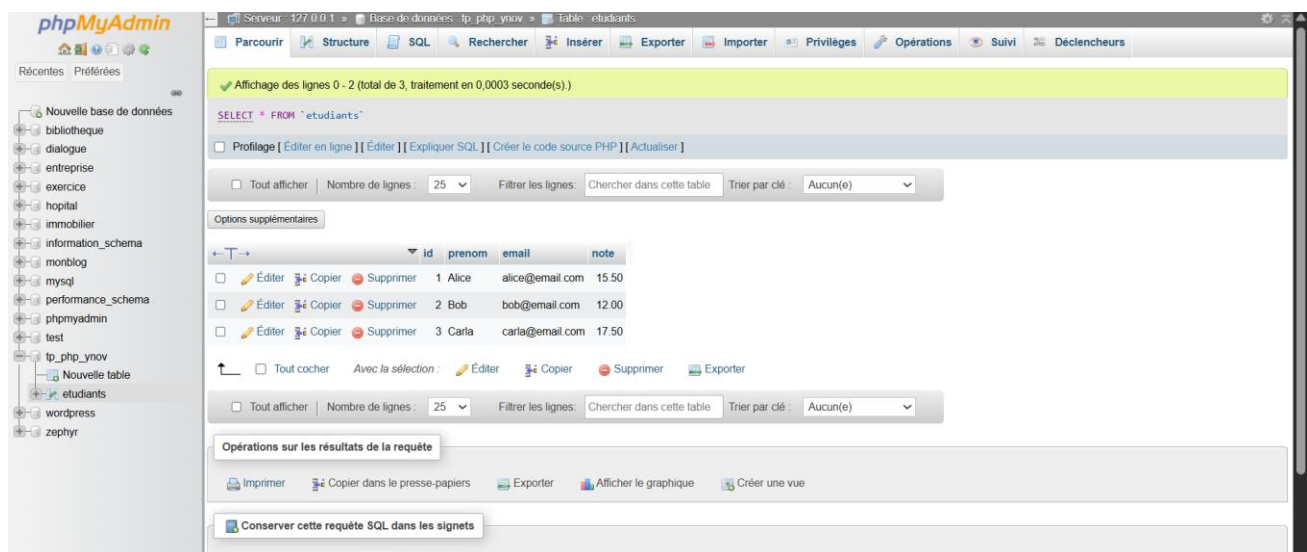
Étape 1 — Créer la base de données

Dans phpMyAdmin ou MySQL, exécuter ce SQL :

```
CREATE DATABASE tp_php;  
USE tp_php;
```

```
CREATE TABLE etudiants (  
  id      INT AUTO_INCREMENT PRIMARY KEY,  
  prenom  VARCHAR(100) NOT NULL,  
  email   VARCHAR(150) NOT NULL,  
  note    DECIMAL(4,2)  
);
```

```
-- Insérer quelques données de test  
INSERT INTO etudiants (prenom, email, note) VALUES  
  ('Alice', 'alice@email.com', 15.5),  
  ('Bob', 'bob@email.com', 12.0),  
  ('Carla', 'carla@email.com', 17.5);
```



The screenshot shows the phpMyAdmin interface. The left sidebar displays a tree view of databases, with 'tp_php_ynov' selected. The main area shows the 'etudiants' table structure and data. The table has columns 'id', 'prenom', 'email', and 'note'. The data is as follows:

id	prenom	email	note
1	Alice	alice@email.com	15.50
2	Bob	bob@email.com	12.00
3	Carla	carla@email.com	17.50

Étape 2 — Fichier connexion.php

Créer un fichier connexion.php réutilisable :

```
<?php
function getConnexion() {
    $dsn = "mysql:host=localhost;dbname=tp_php;charset=utf8mb4";
    try {
        $pdo = new PDO($dsn, "root", "");
        $pdo->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
        return $pdo;
    } catch (PDOException $e) {
        die("Erreur connexion : " . $e->getMessage());
    }
}
?>
```

Étape 3 — Fichier ex6_liste.php

Afficher la liste des étudiants depuis la base :

```
<?php
require_once "connexion.php";
$pdo = getConnexion();

// SELECT - récupérer tous les étudiants
$stmt = $pdo->query("SELECT * FROM etudiants ORDER BY note DESC");
$etudiants = $stmt->fetchAll(PDO::FETCH_ASSOC);

echo "<h2>Liste des étudiants</h2>";
echo "<table border='1'>";
echo "<tr><th>ID</th><th>Prénom</th><th>Email</th><th>Note</th></tr>";

foreach ($etudiants as $etudiant) {
    echo "<tr>";
    echo "<td>" . $etudiant["id"] . "</td>";
    echo "<td>" . htmlspecialchars($etudiant["prenom"]) . "</td>";
    echo "<td>" . htmlspecialchars($etudiant["email"]) . "</td>";
    echo "<td>" . $etudiant["note"] . "/20</td>";
    echo "</tr>";
}
echo "</table>";
?>
```

Étape 4 — Fichier ex6_ajout.php

Ajouter un étudiant avec une requête préparée (sécurisée) :

```
<?php
require_once "connexion.php";
$pdo = getConnexion();
```

```
// INSERT avec requête préparée – anti-injection SQL
$stmt = $pdo->prepare(
    "INSERT INTO etudiants (prenom, email, note) VALUES (:prenom, :email,
:note)"
);
```

```
$stmt->execute([
    ":prenom" => "David",
    ":email"   => "david@email.com",
    ":note"   => 13.5
]);
```

```
echo "Etudiant ajouté avec succès ! ID = " . $pdo->lastInsertId();
?>
```

Resultat attendu : ex6_liste.php : tableau avec Alice (15.5), Bob (12.0), Carla (17.5). ex6_ajout.php : Etudiant ajouté avec succès ! ID = 4

Attention : Toujours utiliser des requêtes préparées avec :parametre pour les données venant de l'utilisateur. Ne jamais concatener directement une variable dans le SQL !

Etudiants ajoutés avec succès ! ID = 5

Liste des étudiants

ID	Prénom	Email	Note
3	Carla	carla@email.com	17.50/20
1	Alice	alice@email.com	15.50/20
4	David	david@email.com	13.50/20
2	Bob	bob@email.com	12.00/20

Défi final

4. Modifiez ex6_liste.php pour afficher la mention à côté de chaque note
5. Créez ex6_recherche.php : un formulaire avec un champ 'prénom' qui cherche dans la BDD
6. Créez ex6_supprimer.php : supprimer un étudiant par son ID avec DELETE préparé

Récapitulatif des notions vues

Exercice	Notion	Fonctions / mots-clés
Ex. 1	Variables et types	<code>echo</code> , <code>var_dump()</code> , <code>gettype()</code>
Ex. 2	Conditions	<code>if</code> / <code>elseif</code> / <code>else</code> / <code>switch</code>
Ex. 3	Boucles	<code>for</code> / <code>while</code> / <code>foreach</code>
Ex. 4	Fonctions	<code>function</code> , <code>return</code> , paramètres par défaut
Ex. 5	Formulaires	<code>\$_POST</code> , <code>htmlspecialchars()</code> , <code>filter_var()</code>
Ex. 6	Base de données	<code>PDO</code> , <code>prepare()</code> , <code>execute()</code> , <code>fetchAll()</code>

Bon courage ! PHP est un langage très pratique — la pratique est la meilleure façon d'apprendre.