

# Fiche de Procédure : Les Bases du PHP

## 1. Qu'est-ce que PHP ?

PHP (Hypertext Preprocessor) est un langage de programmation côté serveur, principalement utilisé pour créer des sites web dynamiques. Il permet de générer du HTML, de gérer des formulaires, d'interagir avec des bases de données, etc.

## 2. Structure de base d'un fichier PHP

Un fichier PHP porte l'extension `.php` et contient du code PHP entre les balises `<?php ... ?>`.

```
<?php
// Ceci est un commentaire
echo "Bonjour, PHP !";
?>
```

Le code PHP peut être mélangé avec du HTML :

```
<!DOCTYPE html>
<html lang="fr">
<head>
  <meta charset="UTF-8">
  <title>Exemple PHP</title>
</head>
<body>
  <h1>
    <?php echo "Bienvenue sur mon site !"; ?>
  </h1>
</body>
</html>
```

### 3. Les variables et types de base

Type	Exemple
String	<code>\$nom = "Marie";</code>
Integer	<code>\$age = 25;</code>
Float	<code>\$prix = 9.99;</code>
Boolean	<code>\$actif = true;</code>
Array	<code>\$liste = array(1, 2, 3);</code>

Remarque : Les variables commencent toujours par \$.

### 4. Les structures de contrôle

Condition (if, else, elseif)

```
$age = 20;
if ($age >= 18) {
    echo "Majeur";
} else {
    echo "Mineur";
}
```

Boucle (for)

```
for ($i = 0; $i < 5; $i++) {
    echo "Tour numéro $i<br>";
}
```

Boucle (while)

```
$compteur = 0;
while ($compteur < 3) {
    echo "Compteur : $compteur<br>";
    $compteur++;
}
```

## 5. Fonctions

```
function direBonjour($prenom) {  
    echo "Bonjour $prenom !";  
}
```

```
// Appel de la fonction  
direBonjour("Lucie");
```

## 6. Tableaux

Tableau simple

```
$fruits = array("Pomme", "Banane", "Cerise");  
echo $fruits[1]; // Affiche "Banane"
```

Tableau associatif

```
$personne = array(  
    "nom" => "Marie",  
    "age" => 25  
);  
echo $personne["nom"]; // Affiche "Marie"
```

## 7. Formulaires et récupération de données

Formulaire HTML

```
<form method="post" action="traitement.php">  
    <input type="text" name="prenom">  
    <button type="submit">Envoyer</button>  
</form>
```

## Traitement PHP

```
<?php
if (isset($_POST['prenom'])) {
    $prenom = $_POST['prenom'];
    echo "Bonjour $prenom !";
}
?>
```

## 8. Exemple complet

```
<?php
$nom = "Marie";
$age = 25;
echo "<h1>Bonjour $nom !</h1>";
echo "<p>Vous avez $age ans.</p>";
?>
```

## 9. Bonnes pratiques

- Toujours valider et sécuriser les données utilisateur (ex : `htmlspecialchars()`).
- Indenter et commenter le code pour la lisibilité.
- Utiliser des noms de variables explicites.
- Séparer la logique PHP et la présentation HTML autant que possible.
- Tester chaque script sur un serveur local (ex : XAMPP, WAMP, MAMP).

## 10. Pour aller plus loin

- Manipuler les fichiers (`fopen`, `fwrite`, etc.).
- Gérer les sessions et cookies (`$_SESSION`, `$_COOKIE`).
- Interagir avec une base de données MySQL (PDO, MySQLi).
- Organiser le code avec des fonctions, des classes et l'orienté objet.
- Découvrir les frameworks PHP (Laravel, Symfony).

Astuce :

Pour tester rapidement du PHP, il faut un serveur local (Apache, Nginx) car le navigateur seul ne peut pas exécuter du PHP.