

# Cours : Les variables, structures de contrôle et fonctions en PHP

---

## Objectifs pédagogiques :

- Comprendre le rôle des variables en PHP.
  - Maîtriser les structures conditionnelles et les boucles.
  - Définir et utiliser des fonctions en PHP pour organiser le code.
- 

## 1. Les variables en PHP

### Définition :

Une **variable** est un espace mémoire nommé, utilisé pour stocker une valeur temporairement.

### Syntaxe :

- Les noms de variables commencent toujours par un \$.
- Elles peuvent contenir des lettres, chiffres, et underscores, mais ne doivent pas commencer par un chiffre.

```
$nom = "Alice";  
$age = 22;  
$estEtudiant = true;
```

### Types de variables :

- **Chaînes de caractères (string)** : "texte"
- **Entiers (int)** : 42
- **Nombres à virgule flottante (float)** : 3.14
- **Booléens (bool)** : true ou false
- **Tableaux (array)** : array("un", "deux")

### Concaténation :

```
$prenom = "Jean";  
echo "Bonjour " . $prenom;
```

---

## 2. Les structures conditionnelles

### if, else, elseif :

```
$age = 20;
```

```

if ($age < 18) {
    echo "Mineur";
} elseif ($age == 18) {
    echo "Tout juste majeur";
} else {
    echo "Majeur";
}

```

*Opérateurs de comparaison :*

Opérateur	Signification
-----------	---------------

==	égal à
----	--------

!=	différent de
----	--------------

<, >, <=, >= infériorité/supériorité

```

switch:
$jour = "lundi";
switch ($jour) {
    case "lundi":
        echo "Début de semaine";
        break;
    case "vendredi":
        echo "Fin de semaine";
        break;
    default:
        echo "Jour normal";
}

```

---

### 3. Les boucles

Les boucles permettent de répéter une instruction tant qu'une condition est vraie.

```

while:
$i = 1;
while ($i <= 5) {
    echo $i;
    $i++;
}

```

```

do...while:
$i = 1;
do {
    echo $i;
    $i++;
} while ($i <= 5);

```

```

for:
for ($i = 1; $i <= 5; $i++) {
    echo $i;
}

```

*foreach (pour les tableaux) :*

```

$couleurs = array("rouge", "vert", "bleu");

```

```
foreach ($couleurs as $couleur) {  
    echo $couleur;  
}
```

---

## 4. Les fonctions en PHP

### Définition :

Une **fonction** est un bloc de code réutilisable, que l'on peut appeler autant de fois que nécessaire.

### Syntaxe :

```
function direBonjour($nom) {  
    return "Bonjour, " . $nom . " !";  
}  
  
echo direBonjour("Marie");
```

- Une fonction peut prendre des **paramètres**.
- Elle peut **retourner une valeur** avec `return`.

### Fonction sans paramètre :

```
function salutation() {  
    echo "Bienvenue !";  
}  
salutation();
```

### Portée des variables :

- **Locale** : à l'intérieur d'une fonction.
- **Globale** : accessible dans tout le script si déclarée `global`.

```
$nom = "Lucie";  
  
function afficherNom() {  
    global $nom;  
    echo $nom;  
}  
afficherNom();
```

---

## 5. Bonnes pratiques

- Nommez vos variables et fonctions de manière claire.
  - Séparez votre code en fonctions pour le rendre plus lisible.
  - Commentez vos fonctions si elles effectuent des tâches complexes.
  - Évitez de répéter du code : réutilisez vos fonctions !
-

## ✔ Conclusion

Dans ce chapitre, vous avez appris à manipuler les **variables**, à utiliser les **structures conditionnelles**, à maîtriser les **boucles**, et à écrire des **fonctions réutilisables**. Ces éléments sont la base de toute logique dans vos scripts PHP. Vous êtes maintenant capable de créer des programmes interactifs et dynamiques !