

## Intégration de PHP dans HTML

PHP s'exécute côté serveur, ce qui signifie que le navigateur reçoit uniquement du HTML généré. Voici un exemple simple :

### Exemple :

```
<!DOCTYPE html>
<html lang="fr">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Introduction à PHP</title>
</head>
<body>
  <h1>Bienvenue en PHP</h1>
  <?php
    echo "<p>Ceci est un paragraphe généré en PHP.</p>";
  ?>
</body>
</html>
```

Dans cet exemple, le code PHP est placé à l'intérieur des balises <?php ... ?>.

## 2. Déclaration et affichage des variables

### Syntaxe :

En PHP, une variable commence par \$ et son type est déterminé automatiquement.

### Exemple :

```
<?php
  $nom = "Alice";
  $age = 25;
  $prix = 9.99;
  $estMajeur = true;

  echo "<p>Nom : $nom</p>";
  echo "<p>Âge : $age ans</p>";
  echo "<p>Prix : $prix €</p>";
  echo "<p>Est majeur : " . ($estMajeur ? "Oui" : "Non") . "</p>";
?>
```

### Explication :

- PHP ne nécessite pas de déclaration de type.
- Les chaînes de caractères sont entourées de guillemets (" ou ').
- Les booléens s'affichent sous forme de 1 (true) ou 0 (false).

- Le point (.) est utilisé pour la concaténation.

### 3. Les structures conditionnelles (if, else, elseif)

Elles permettent d'exécuter du code selon certaines conditions.

**Exemple :**

```
<?php
    $age = 18;

    if ($age < 18) {
        echo "<p>Vous êtes mineur.</p>";
    } elseif ($age == 18) {
        echo "<p>Vous venez d'atteindre la majorité.</p>";
    } else {
        echo "<p>Vous êtes majeur.</p>";
    }
?>
```

*Explication :*

- if teste une condition.
- elseif permet d'ajouter une condition supplémentaire.
- else est exécuté si aucune des conditions précédentes n'est vraie.

### 4. Les boucles (while, for, foreach)

#### 4.1 La boucle while

Elle s'exécute tant qu'une condition est vraie.

**Exemple :**

```
<?php
    $i = 1;
    while ($i <= 5) {
        echo "<p>Numéro : $i</p>";
        $i++;
    }
?>
```

#### 4.2 La boucle for

Elle est utilisée lorsqu'on connaît le nombre d'itérations.

**Exemple :**

```
<?php
```

```
for ($i = 1; $i <= 5; $i++) {  
    echo "<p>Itération $i</p>";  
}  
?>
```

#### 4.3 La boucle foreach

Elle sert à parcourir un tableau.

**Exemple :**

```
<?php  
$fruits = ["Pomme", "Banane", "Cerise"];  
  
foreach ($fruits as $fruit) {  
    echo "<p>Fruit : $fruit</p>";  
}  
?>
```

### 5. Les tableaux en PHP

#### 5.1 Les tableaux simples

Un tableau simple est une liste ordonnée d'éléments.

**Exemple :**

```
<?php  
$nombres = [10, 20, 30, 40, 50];  
  
echo "<p>Premier élément : " . $nombres[0] . "</p>"; // Affiche 10  
?>
```

#### 5.2 Les tableaux associatifs

Un tableau associatif utilise des clés personnalisées.

**Exemple :**

```
<?php  
$personne = [  
    "nom" => "Alice",  
    "age" => 25,  
    "ville" => "Paris"  
];  
  
echo "<p>Nom : " . $personne["nom"] . "</p>";
```

```
echo "<p>Âge : " . $personne["age"] . " ans</p>";  
echo "<p>Ville : " . $personne["ville"] . "</p>";  
?>
```

### 5.3 Parcourir un tableau associatif avec foreach

```
<?php  
foreach ($personne as $cle => $valeur) {  
    echo "<p>$cle : $valeur</p>";  
}  
?>
```

## 1. Ajout et suppression d'éléments

**array\_push()** : Ajouter un élément à la fin

```
<?php  
$fruits = ["Pomme", "Banane"];  
array_push($fruits, "Orange", "Mangue");  
print_r($fruits);  
?>
```

**Résultat** : ["Pomme", "Banane", "Orange", "Mangue"]

**array\_unshift()** : Ajouter un élément au début

```
<?php  
$fruits = ["Pomme", "Banane"];  
array_unshift($fruits, "Fraise");  
print_r($fruits);  
?>
```

**Résultat** : ["Fraise", "Pomme", "Banane"]

**array\_pop()** : Supprimer le dernier élément

```
<?php  
$fruits = ["Pomme", "Banane", "Orange"];  
array_pop($fruits);  
print_r($fruits);  
?>
```

**Résultat** : ["Pomme", "Banane"]

**array\_shift()** : Supprimer le premier élément

```
<?php  
$fruits = ["Pomme", "Banane", "Orange"];  
array_shift($fruits);
```

```
print_r($fruits);  
?>
```

**Résultat :** ["Banane", "Orange"]

## 2. Rechercher dans un tableau

`in_array()` : Vérifier si une valeur existe

```
<?php  
$fruits = ["Pomme", "Banane", "Orange"];  
if (in_array("Banane", $fruits)) {  
    echo "Banane est dans le tableau."  
}  
?>
```

**Résultat :** "Banane est dans le tableau."

`array_search()` : Trouver l'index d'une valeur

```
<?php  
$fruits = ["Pomme", "Banane", "Orange"];  
$index = array_search("Orange", $fruits);  
echo "Index de Orange : $index";  
?>
```

**Résultat :** Index de Orange : 2

## 3. Découpage et fusion de tableaux

`array_slice()` : Extraire une portion d'un tableau

```
<?php  
$nombres = [10, 20, 30, 40, 50];  
$partie = array_slice($nombres, 1, 3);  
print_r($partie);  
?>
```

**Résultat :** [20, 30, 40] (démontre à l'index 1, prend 3 éléments)

`array_splice()` : Supprimer et remplacer une portion

```
<?php  
$nombres = [10, 20, 30, 40, 50];  
array_splice($nombres, 1, 2, [100, 200]);  
print_r($nombres);  
?>
```

**Résultat :** [10, 100, 200, 40, 50] (remplace 20, 30 par 100, 200)

`array_merge()` : Fusionner deux tableaux

```
<?php
    $tab1 = [1, 2, 3];
    $tab2 = [4, 5, 6];
    $fusion = array_merge($tab1, $tab2);
    print_r($fusion);
?>
```

**Résultat :** [1, 2, 3, 4, 5, 6]

array\_combine() : Créer un tableau associatif

```
<?php
    $cles = ["nom", "age", "ville"];
    $valeurs = ["Alice", 25, "Paris"];
    $assoc = array_combine($cles, $valeurs);
    print_r($assoc);
?>
```

**Résultat :**

["nom" => "Alice", "age" => 25, "ville" => "Paris"]