



Chapitre : Les Fonctions en PHP



Par Robert DIASSÉ

Les Fonctions en PHP

Les fonctions sont des blocs de code réutilisables qui effectuent une tâche spécifique. En PHP, il existe une variété de fonctions prédéfinies, mais vous pouvez également créer vos propres fonctions pour modulariser votre code et le rendre plus lisible. Dans ce cours, nous explorerons les différents types de fonctions, de la création de fonctions simples à celles avec des paramètres et des valeurs de retour.

I. Introduction aux Fonctions en PHP

1.1 Fonctions Prédéfinies

PHP propose une multitude de fonctions prédéfinies couvrant divers domaines tels que les chaînes de caractères, les tableaux, la manipulation de fichiers, et bien plus encore. Exemples :

```
// Fonctions pour les chaînes de caractères
$longueur = strlen("Hello, world!"); // Retourne la longueur de la chaîne
$enMajuscules = strtoupper("Bonjour"); // Convertit la chaîne en majuscules

// Fonctions pour les tableaux
$elements = [1, 2, 3, 4, 5];
$taille = count($elements); // Retourne la taille du tableau
$somme = array_sum($elements); // Calcule la somme des éléments du tableau
```

1.2 Déclaration de Fonction

Pour créer une fonction personnalisée, utilisez le mot-clé `function`. Voici une fonction simple qui affiche un message :

```
function afficherMessage() {
    echo "Bonjour, bienvenue!";
}
```

```
}

// Appel de la fonction
afficherMessage();
```

II. Types de Fonctions en PHP

2.1 Fonctions sans Paramètre ni Valeur de Retour

```
function saluer() {
    echo "Bonjour!";
}

// Appel de la fonction
saluer();
```

2.2 Fonctions avec Paramètres

```
function multiplier($a, $b) {
    $produit = $a * $b;
    echo "Le produit est : $produit";
}

// Appel de la fonction
multiplier(5, 3);
```

2.3 Fonctions avec Valeur de Retour

```
function calculerSomme($x, $y) {
    return $x + $y;
}

// Appel de la fonction
$resultat = calculerSomme(10, 15);
echo "La somme est : $resultat";
```

2.4 Fonctions avec Paramètres par Défaut

```
function saluerUtilisateur($nom = "Invité") {
    echo "Bonjour, $nom!";
}

// Appel de la fonction
```

```
saluerUtilisateur(); // Affiche "Bonjour, Invité!"
saluerUtilisateur("John"); // Affiche "Bonjour, John!"
```

III. Fonctions Utilisateur Avancées

3.1 Fonctions Variables (Fonctions Anonymes ou closure)

```
$ajouter = function ($a, $b) {
    return $a + $b;
};

// Appel de la fonction variable
$resultat = $ajouter(3, 4);
echo "La somme est : $resultat";
```

3.2 Fonctions Récursives

Les fonctions récursives s'appellent elles-mêmes. Exemple :

```
function factorielle($n) {
    if ($n === 0 || $n === 1) {
        return 1;
    } else {
        return $n * factorielle($n - 1);
    }
}

// Appel de la fonction récursive
$resultat = factorielle(5);
echo "La factorielle est : $resultat";
```

IV. Fonctions à Nombre Variable d'Arguments

Dans PHP, vous pouvez déclarer une fonction avec un nombre variable d'arguments en utilisant le mot-clé `...`. Ces arguments sont ensuite traités comme un tableau à l'intérieur de la fonction.

```
function afficherArguments(...$args) {
    foreach ($args as $arg) {
        echo "Argument : $arg<br>";
    }
}

// Appel de la fonction
afficherArguments("PHP", "est", "puissant");
```

V. Fonctions avec Utilisation des Pointeurs

Les pointeurs peuvent être utilisés pour modifier directement les variables en mémoire. En PHP, la notion de pointeur est implémentée en utilisant les références.

```
function doubler(&$valeur) {  
    $valeur *= 2;  
}  
  
$nombre = 5;  
doubler($nombre);  
echo "Le double est : $nombre";
```

VI. Fonctions avec Types Forts

À partir de PHP 7, vous pouvez définir les types attendus pour les arguments et les valeurs de retour d'une fonction.

```
declare(strict_types=1);  
  
function additionner(int $a, int $b): int {  
    return $a + $b;  
}  
  
$resultat = additionner(5, 3);  
echo "La somme est : $resultat";
```

VII. Fonctions de Rappel (Callback)

Les fonctions de rappel sont des fonctions que vous pouvez passer comme arguments à d'autres fonctions.

```
function appliquerOperation($a, $b, callable $operation) {  
    return $operation($a, $b);  
}  
  
function additionner($a, $b) {  
    return $a + $b;  
}  
  
function multiplier($a, $b) {  
    return $a * $b;  
}  
  
// Utilisation de fonctions de rappel  
$resultat1 = appliquerOperation(5, 3, 'additionner');  
$resultat2 = appliquerOperation(5, 3, 'multiplier');
```

```
echo "Addition : $resultat1<br>";  
echo "Multiplication : $resultat2";
```

Exercices Corrigés

Exercice 1 : Calcul de Moyenne

```
function calculerMoyenne($note1, $note2, $note3) {  
    return ($note1 + $note2 + $note3) / 3;  
}  
  
$moyenne = calculerMoyenne(14, 18, 16);  
echo "La moyenne est : $moyenne";
```

Exercice 2 : Générateur de Salutation

```
function genererSalutation($nom, $langue = "fr") {  
    switch ($langue) {  
        case "fr":  
            return "Bonjour, $nom!";  
        case "en":  
            return "Hello, $nom!";  
        default:  
            return "Salut, $nom!";  
    }  
}  
  
$salutationFr = genererSalutation("Marie");  
$salutationEn = genererSalutation("John", "en");  
  
echo "$salutationFr<br>";  
echo "$salutationEn";
```

Exercice 3 : Fonction à Nombre Variable d'Arguments

```
function calculerProduit(...$nombres) {  
    $produit = 1;  
    foreach ($nombres as $nombre) {  
        $produit *= $nombre;  
    }  
    return $produit;  
}  
  
$resultat = calculerProduit(2, 3, 4);  
echo "Le produit est : $resultat";
```

Exercice 4 : Utilisation des Pointeurs

```
function incrementer(&$valeur) {  
    $valeur++;  
}  
  
$compteur = 10;  
incrementer($compteur);  
echo "La valeur incrémentée est : $compteur";
```

Exercice 5 : Fonction avec Types Forts

```
declare(strict_types=1);  
  
function diviser(int $dividende, int $diviseur): float {  
    if ($diviseur === 0) {  
        throw new InvalidArgumentException("Division par zéro !");  
    }  
    return $dividende / $diviseur;  
}  
  
try {  
    $resultat = diviser(10, 2);  
    echo "Le résultat de la division est : $resultat";  
} catch (InvalidArgumentException $e) {  
    echo "Erreur : " . $e->getMessage();  
}
```

Exercice 6 : Fonction de Rappel

```
function appliquerOperation($a, $b, callable $operation) {  
    return $operation($a, $b);  
}  
  
function soustraire($a, $b) {  
    return $a - $b;  
}  
  
$resultat = appliquerOperation(8, 3, 'soustraire');  
echo "La soustraction donne : $resultat";
```

Conclusion

Les fonctions sont des éléments fondamentaux de la programmation en PHP, offrant la possibilité de réutiliser du code et de le rendre plus modulaire. Explorez davantage les fonctions prédéfinies de PHP et expérimentez

avec la création de vos propres fonctions pour améliorer vos compétences en programmation. N'hésitez pas à consulter la [documentation officielle de PHP](#) pour en savoir plus sur les fonctions disponibles.