

# Les fonctions en PHP

Les fonctions en PHP permettent de regrouper un ensemble d'instructions sous un nom unique, facilitant leur réutilisation et leur organisation dans le code.

## 1 - Définir une fonction

Une fonction en PHP se déclare à l'aide du mot clé **function** suivi du nom de la fonction et des parenthèses.

*Exemple :*

Voici une fonction simple qui affiche un texte :

```
function direBonjour() {  
    echo "Bonjour !";  
}  
direBonjour(); // Affiche "Bonjour !"
```

php

### 1.1 - Paramètres d'une fonction

Les fonctions peuvent accepter des paramètres pour personnaliser leur comportement.

*Exemple :*

Une fonction avec un paramètre :

```
function saluer($nom) {  
    echo "Bonjour, $nom !";  
}  
saluer("Alice"); // Affiche "Bonjour, Alice !"
```

php

### 1.2 - Valeurs par défaut des paramètres

Il est possible de définir des valeurs par défaut pour les paramètres lors de la déclaration de la fonction.

*Exemple :*

```
function saluer($nom = "inconnu") {  
    echo "Bonjour, $nom !";  
}  
saluer(); // Affiche "Bonjour, inconnu !"  
saluer("Bob"); // Affiche "Bonjour, Bob !"
```

php

## Retourner une valeur

Une fonction peut retourner une valeur avec le mot clé **return**.

*Exemple :*

Fonction qui additionne deux nombres :

```
function additionner($a, $b) {
    return $a + $b;
}
$resultat = additionner(3, 4);
echo $resultat; // Affiche "7"
```

php

## Fonctions avec nombre variable d'arguments

PHP permet de créer des fonctions acceptant un nombre illimité d'arguments avec la syntaxe `...`.

*Exemple :*

Additionner tous les arguments passés :

```
function additionnerTout(...$nombres) {
    return array_sum($nombres);
}
echo additionnerTout(1, 2, 3, 4); // Affiche "10"
```

php

## Fonctions anonymes

Une fonction anonyme est une fonction sans nom, souvent utilisée comme valeur d'une variable ou dans des callbacks.

*Exemple :*

```
$saluer = function($nom) {
    return "Bonjour, $nom !";
};
echo $saluer("Claire"); // Affiche "Bonjour, Claire !"
```

php

## Fonctions fléchées (PHP 7.4)

Introduites dans PHP 7.4, les fonctions fléchées fournissent une version simplifiée des fonctions anonymes. Elles utilisent la syntaxe `fn` et permettent d'écrire des fonctions courtes en une seule ligne.

### 5.1 - Caractéristiques principales

- Utilisent `fn` au lieu de `function` .
- Ne nécessitent pas le mot clé `return` .
- Héritent automatiquement des variables du contexte parent.

*Exemple :*

Addition de deux nombres avec une fonction fléchée :

```
$addition = fn($a, $b) => $a + $b;
echo $addition(3, 4); // Affiche "7"
```

php

## 5.2 - Comparaison avec une fonction anonyme

Fonction anonyme classique :

```
$multiplication = function($a, $b) {
    return $a * $b;
};
echo $multiplication(3, 4); // Affiche "12"
```

php

Fonction fléchée équivalente :

```
$multiplication = fn($a, $b) => $a * $b;
echo $multiplication(3, 4); // Affiche "12"
```

php

## Nouveautés PHP 8 pour les fonctions

PHP 8 introduit plusieurs améliorations utiles concernant les fonctions.

### 6.1 - Types unis (Union Types)

Une fonction peut maintenant accepter ou retourner plusieurs types de données.

Exemple :

```
function calculer($a, int|float $b): int|float {
    return $a + $b;
}
echo calculer(3, 4.5); // Affiche "7.5"
```

php

### 6.2 - Arguments nommés

Les arguments nommés permettent de préciser explicitement quel paramètre on souhaite donner, en particulier utile lorsque des paramètres ont des valeurs par défaut.

Exemple :

```
function afficher($prenom, $nom = "inconnu", $age = 0) {
    echo "$prenom $nom a $age ans.";
}
afficher(prenom: "Marie", age: 25); // Affiche "Marie inconnu a 25 ans."
```

php

## Fonctions intégrées courantes

PHP propose de nombreuses fonctions intégrées pour différentes tâches. En voici quelques-unes utiles : – **isset** : Vérifie si une variable est définie. – **empty** :

Vérifie si une variable est vide. – **var\_dump** : Affiche les informations détaillées d'une variable (type et contenu). – **strlen** : Renvoie la longueur d'une chaîne. –

**array\_merge** : Fusionne plusieurs tableaux.

## À RETENIR

- Déclaration de fonctions :

- Exemple :

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

php

- Paramètres et valeurs par défaut :

- Exemple :

```
function saluer($nom = "inconnu") {
    echo "Bonjour, $nom !";
}
```

php

- Retour de valeur : Utilisez le mot clé `return` .

- Exemple :

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

php

- Arguments multiples : Utilisez `...` pour accepter plusieurs arguments.

- Exemple :

```
function additionnerTout(...$nombres) {
    return array_sum($nombres);
}
```

php

- Fonctions fléchées (PHP 7.4) :

- Syntaxe concise pour des fonctions courtes.
- Exemple :

```
$addition = fn($a, $b) => $a + $b;
echo $addition(3, 4); // Affiche "7"
```

php

- Nouveautés PHP 8 :

- **Types unis (Union Types)** : Plusieurs types dans les arguments ou retour.

```
function calculer(int|float $a): int|float {
    return $a;
}
```

php

- **Arguments nommés** : Spécifiez les paramètres explicitement.

```
afficher(prenom: "Jean", age: 30);
```

php

- Fonctions intégrées :

- `isset` : Vérifie si une variable existe.
- `empty` : Vérifie si une variable est vide.
- `var_dump` : Affiche le type et la valeur.
- `strlen` : Retourne la longueur d'une chaîne.
- `array_merge` : Combine plusieurs tableaux.