

PHP : Fonctions - Exercices

Il est possible d'attribuer une **valeur par défaut** à un ou plusieurs paramètres d'une fonction. Lorsqu'un paramètre a une valeur par défaut, il devient optionnel dans l'appel de la fonction. Si aucun argument n'est fourni pour ce paramètre, la valeur par défaut est utilisée.

Exercices pour apprendre à utiliser les fonctions en PHP

Exercice 1 – Centraliser un affichage répétitif

Objectif : comprendre l'intérêt de créer une fonction simple sans paramètre. **Situation** : Tu affiches souvent ce message dans tes pages :

```
1 | echo "Bienvenue sur mon site !<br>";
```

Consigne :

1. Copie ce code 3 fois.
2. Puis, remplace-le par un appel à une fonction `afficherBienvenue()`. su
3. Appelle la fonction 3 fois.

Exercice 2 – Réutiliser une logique

Objectif : structurer du code répétitif.

Situation : On affiche l'heure courante dans plusieurs pages.

Code à convertir :

```
1 | echo "Il est " . date("H:i");
```

Consigne : Crée une fonction `afficherHeure()` qui affiche cette information.

Appelle-la dans deux fichiers (ou deux blocs `echo`) pour simuler la réutilisation.

Exercice 3 – Rendre le message dynamique

Objectif : apprendre à créer une fonction avec paramètre.

Consigne :

1. Crée une fonction `direBonjour($prenom)` qui affiche :
Bonjour, [prénom] !
2. Appelle la fonction avec "Alice", "Bob", puis ton propre prénom.

Exercice 4 – Extraire un calcul

Objectif : isoler un calcul dans une fonction pour le réutiliser ailleurs.

Consigne :

1. Crée une fonction `additionner($a, $b)` qui retourne la somme.
2. Affiche plusieurs additions dans la même page :

- 3 + 5
- 10 + 4
- 15 + 7



Exercice 5 – Transformer du code long en fonction

Objectif : découper un traitement en fonction.

Situation :

```
1 | $prixHT = 100;
2 | $tauxTVA = 21;
3 | $prixTTC = $prixHT + ($prixHT * $tauxTVA / 100);
4 | echo "Prix TTC : $prixTTC €";
```

Consigne :

- Transforme ce code en une fonction `calculerPrixTTC($ht, $taux)`.

* Puis affiche le prix TTC de 100€, 250€, 420€ (taux 21%).



Exercice 6 – Utiliser un paramètre par défaut

Objectif : apprendre à gérer les cas où un paramètre peut être optionnel.

Consigne :

1. Crée une fonction `saluer($prenom = "Invité")` qui affiche : `Bonjour, Alice !` ou `Bonjour, Invité !`
2. Appelle-la avec et sans argument.



Exercice 7 – Générer une structure HTML

Objectif : créer une fonction réutilisable pour générer du HTML.

Consigne :

1. Crée une fonction `genererTitre($texte, $niveau = 2)` qui retourne :

```
1 | <h2>Titre</h2> // ou <h1>... selon $niveau
```

2. Utilise-la pour afficher :
 - Un titre principal (`<h1>Accueil`)
 - Un sous-titre (`<h2>Derniers articles`)



Exercice 8 – Analyser du code existant et l'optimiser

Objectif : identifier un besoin de fonction dans un code répétitif.

Code de départ :

```
1 | $prixHT = 50;  
2 | echo "Prix TTC : " . ($prixHT + ($prixHT * 21 / 100)) . " €<br>";  
3 |  
4 | $prixHT = 120;  
5 | echo "Prix TTC : " . ($prixHT + ($prixHT * 21 / 100)) . " €<br>";  
6 |  
7 | $prixHT = 300;  
8 | echo "Prix TTC : " . ($prixHT + ($prixHT * 21 / 100)) . " €<br>";
```

Consigne :

1. Crée une fonction `afficherPrixTTC(\$ht)` avec un taux par défaut de 21. 2. Remplace les 3 blocs par des appels à la fonction.



Exercice 9 – Améliorer la lisibilité du code

Objectif : apprendre à encapsuler une condition dans une fonction.

Situation : Tu veux afficher pour chaque nombre entre 1 et 10 s'il est pair ou impair.

Consigne :

1. Crée une fonction `estPair(\$n)` qui retourne `true` ou `false`. 2. Affiche :

```
1 est impair  
2 est pair  
...
```



Exercice 10 – Construire une bibliothèque de fonctions utiles

Objectif : apprendre à structurer son code dans plusieurs fonctions.

Consigne : Crée un fichier `fonctions.php` avec les fonctions suivantes :

- `direBonjour($nom)`
- `afficherDate()`
- `calculerTVA($montant, $taux = 21)`

Puis crée une page `test.php` qui :

1. inclut `fonctions.php`
2. appelle chaque fonction avec des valeurs de test