



## 9) Les fonctions utilisateurs

Nous venons de voir le chapitre sur les fonctions. Ces fonctions étaient toutes déjà existantes et proposées par PHP via la documentation officielle. Ce sont des fonctions internes à PHP.

Il faut savoir que l'on peut définir nous-même nos propres fonctions PHP. L'objectif c'est d'englober une certaine logique que vous allez pouvoir réutiliser dans vos applications. Ça permettra d'éviter de répéter plusieurs fois le même code.

Pour commencer, créez d'abord un fichier « **fonctionsUtilisateurs.php** » dans coursphp.

Pour **créer** une **fonction** en **PHP** rien de plus simple, vous devez commencer par écrire le mot clé « **function** » **suivi** du **nom** de votre **fonction**, un nom logique par exemple si vous écrivez une fonction qui calcule la somme de deux entiers, vous n'allez pas l'appeler fonction bonjour, mais plutôt par exemple fonction somme() ou fonction sommeDeDeuxNombres() ou addition() etc.

Pour le nom, c'est comme pour les variables, on a le droit d'utiliser que des caractères alphanumériques et des Under score.

Voici la syntaxe :

```
- function nomDeLaFonction(0 à plusieurs paramètres séparé par des virgules){  
    Instruction(s) ;  
}
```

On peut mettre de 0 à N paramètres.

**Attention** une fonction existante ne peut pas être remplacé, par exemple si je crée une fonction rand(), ça sortira une erreur parce qu'elle existe déjà. Pareil si je crée une fonction « **somme** » deux fois ça posera un problème.

Nous allons commencer par créer une fonction qui s'appellera `hello()`, qui affichera juste `hello world`.

```
fonctionsUtilisateurs.php > ...  
1  <?php  
2      function hello(){  
3          echo "Hello World !!!";  
4      }  
5      hello();  
6  ?>
```

Ce code on voit donc qu'on a une fonction qui ne retourne rien, elle se contente juste d'afficher une chaîne de caractères.

On appelle une fonction pour l'utiliser en donnant juste son nom suivi de ses parenthèses avec ou sans paramètres en fonction de la fonction. Ici dans notre cas, c'était une fonction sans paramètres donc parenthèse vide.

```
← → ↻ ⓘ localhost/coursphp/fonctionsUtilisateurs.php
```

Hello World !!!

Voici l'affichage.

On va essayer de la modifier un peu, pour qu'il arrive à dire Hello suivi d'un prénom. Par exemple Hello Julien.

```
fonctionsUtilisateurs.php > hello  
1  <?php  
2      function hello($nom){  
3          echo "Hello ". $nom . " !!!";  
4      }  
5      hello("Julien");
```

Cette fonction ne retourne toujours rien, mais cette fois-ci elle reçoit quelques choses en paramètre. On lui a donné une variable sans donner le type. Ensuite la fonction affiche Hello suivi de ce qu'on lui donnera en paramètre, dans ce cas si à la ligne 5, je lui ai donné un String (une chaîne de caractère) portant le prénom Julien.

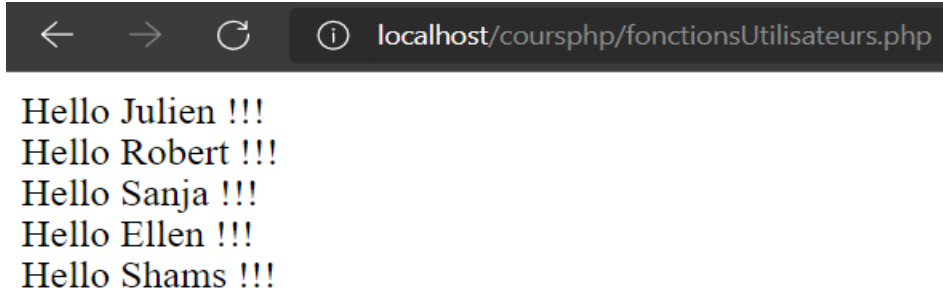
Il affiche donc logiquement :

```
PS C:\wamp64\www\coursphp> php fonctionsUtilisateurs.php  
Hello Julien !!!
```

J'aurai pu afficher plusieurs prénoms à la ligne juste en appelant à chaque fois la fonction et en changeant juste le prénom en paramètre.

### (Exos)

1) A votre avis en vous basant sur mon exemple comment obtenir cela :

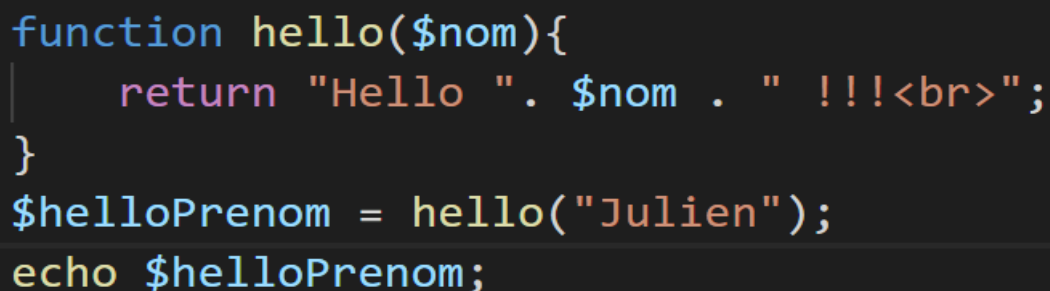


```
← → ↻ ⓘ localhost/coursphp/fonctionsUtilisateurs.php  
Hello Julien !!!  
Hello Robert !!!  
Hello Sanja !!!  
Hello Ellen !!!  
Hello Shams !!!
```

2) Comment à votre avis on pourrait améliorer le code précédent pour qu'on ne se répète pas plusieurs fois.

On a vu précédemment qu'une fonction pouvait retourner quelques choses aussi. Par exemple si je vais voir la fonction `rand()` dans la documentation officielle de PHP, on nous dit bien qu'elle **retourne un nombre**.

C'est donc grâce au mot clé « `return` » ce qui veut dire en français « retourne ». En gros retourne moi ça. Donc lorsque la fonction sera exécutée, elle va renvoyer le résultat qu'on souhaite. Reprenons notre exemple sur l'affichage de « Hello Julien !!! » et modifions-le comme cela :



```
function hello($nom){  
    return "Hello ". $nom . " !!!<br>";  
}  
$helloPrenom = hello("Julien");  
echo $helloPrenom;
```

Le résultat sera le même et donnera bien Hello Julien !!!.

On voit bien que notre fonction retourne quelques choses. On va récupérer ce quelques choses dans une variable que j'ai appelé « `helloPrenom` ». Ici on sait que c'est une chaîne de caractère qui est retournée mais on aurait bien pu retourner un entier ou un booléen ou un tableau etc.

A la fin je fais juste un `echo` de ce qu'il y a dans la variable « `helloPrenom` ».

**Attention** tout ce qui est en dessous du `return` ne sera pas pris en compte. En gros dès que la fonction rencontre un `return`, il sort automatiquement de la fonction.

Si vous tentez aussi d'appeler une fonction qui n'a pas le même nombre de paramètres. Par exemple ici j'ai appelé à la ligne 28 la fonction hello sans lui donner de paramètre/argument, il me ressort une erreur.

```
25     function hello($nom){
26         return "Hello ". $nom . " !!!<br>";
27     }
28     $helloPrenom = hello();
29     echo $helloPrenom;
```

PROBLÈMES 1 SORTIE TERMINAL CONSOLE DE DÉBOGAGE powershell + - [ ] [X] ^ X

PS C:\wamp64\www\coursphp> php fonctionsUtilisateurs.php

Fatal error: Uncaught ArgumentCountError: Too few arguments to function hello(), 0 passed in C:\wamp64\www\coursphp\fonctionsUtilisateurs.php on line 28 and exactly 1 expected in C:\wamp64\www\coursphp\fonctionsUtilisateurs.php:25

Stack trace:

#0 C:\wamp64\www\coursphp\fonctionsUtilisateurs.php(28): hello()  
#1 {main}  
thrown in C:\wamp64\www\coursphp\fonctionsUtilisateurs.php on line 25

Il faut savoir lire les erreurs, ici concrètement il dit que il y a trop peu d'argument dans la fonction hello(), zéro a été passé en argument à la fonction hello à la ligne 28 et exactement 1 seul argument est attendu à la fonction hello à la ligne 25 du fichier.

Voilà donc PHP vous donne tout ce qu'il vous faut pour réparer cette erreur.

On peut donner une valeur par défaut pour que si on ne met aucun argument il affiche Hello World.

```
29     function hello($nom = null ){
30         if ($nom != null){
31             return "Hello ". $nom . " !!!<br>";
32         }
33         return "Hello World !!!<br>";
34     }
35     echo hello();
36     echo hello("Julien");
```

Du coup ici ça règlera le problème de si on ne met pas d'argument car il affichera Hello World et si on ajoute un argument ('Julien') il mettra Hello Julien.

**A noter** que toutes les variables qui sont déclaré en dehors de la fonction, si on les appelle dedans, il va tout simplement les nier. Si je rajoute un prénom que je déclare en dehors de la fonction et que j'essaie de l'afficher dans la fonction, ça ne va rien faire.

(Exos)

- 3) Essayer d'améliorer le code juste en haut en demandant à l'utilisateur d'écrire au clavier un prénom et qu'il affiche Hello suivi du prénom mais si on ne met pas de prénom il affiche Hello World.

```
PS C:\wamp64\www\coursphp> php fonctionsUtilisateurs.php
Veuillez introduire un prénom : Julien
Hello Julien !!!
PS C:\wamp64\www\coursphp> php fonctionsUtilisateurs.php
Veuillez introduire un prénom :
Hello World !!!
```

- 4) Créez une fonction « bienvenue » qui permet d'afficher le nom, le prénom, l'email et l'âge. Le tout entré au clavier.

```
Veuillez introduire un prénom : Julien
Veuillez introduire un nom : Dunia
Veuillez introduire un email : juliendunia@hotmail.com
Veuillez introduire votre age : 18
Bienvenue Dunia Julien !!!
Vous avez 18 ans.
Voici votre mail juliendunia@hotmail.com
```

- 5) Créez une fonction « adulte » qui permet de vérifier si une personne est majeure ou mineure. Donc quand on introduit au clavier, il affichera comme message vous êtes majeur ou vous êtes mineur.
- 6) (Difficile) Créez une fonction « addition » qui retourne quelque chose et qui permet de faire la somme de 2 entiers, entré au clavier. Je veux exactement **LE MEME** type de résultat.

```
PS C:\wamp64\www\coursphp> php fonctionsUtilisateurs.php
Veuillez introduire un premier nombre : 13
Veuillez introduire un deuxieme nombre : 5
13 additionné à 5 donne 18
```

- 7) Je veux que vous me fassiez pareil que la question 6, sauf que ça doit être une fonction « soustraction » qui retourne une soustraction.
- 8) Je veux aussi la même chose pour la « multiplication ».
- 9) (Difficile) Je veux la même chose pour la « division », **ATTENTION** on ne sait pas diviser par 0. Faites en sortes qu'on vous renvoi un message comme cela et qu'on vous redemande d'introduire des nombres tant que la division ne se fait pas.

```
PS C:\wamp64\www\coursphp> php fonctionsUtilisateurs.php
Veuillez introduire un premier nombre : 45
Veuillez introduire un deuxieme nombre : 0
Vous avez tenter une division par 0, recommencer
Veuillez introduire un premier nombre : 45
Veuillez introduire un deuxieme nombre : 2
45 divisé par 2 donne 22.5
```

Sachez que vous pouvez utiliser une fonction dans une fonction, si vous copiez collez ce code

```
function addition($nb1,$nb2){
    return $nb1+ $nb2 ;
}
function moyenne2nombre($nb1,$nb2){
    return addition($nb1,$nb2)/2;
}
$nbr1 = readline("Veuillez introduire un premier nombre : ");
$nbr2 = readline("Veuillez introduire un deuxième nombre : ");
echo "La moyenne de " . $nbr1 . " et " . $nbr2 . " donne
". moyenne2nombre($nbr1,$nbr2);
```

Vous verrez qu'il affichera bien la moyenne de 2 nombres.

**A noter** Il faut savoir que depuis **php 7.1**, on peut typer les variables des paramètres dans les fonctions. Donc on peut définir son type :

```
function typage(int $entier, string $chaineDeCar, bool $booleen){}
function typages(float $nombreDecimal, array $tableau) : float {
    return 2.5;
}
```

On peut aussi préciser le type de retour d'une fonction, ici pour ma deuxième fonction on voit que j'ai mis « : float » ça veut dire que ma fonction va devoir retourner un nombre décimal. Une fonction qui ne retourne rien ce sera « : void ».

Type	Description	Version
Nom de classe/interface	La valeur doit être une <a href="#">instanceof</a> de la classe ou interface donnée.	
self	La valeur doit être une <a href="#">instanceof</a> de la même classe que celle dans laquelle la déclaration de type est utilisée. Peut uniquement être utilisé dans les classes.	
parent	La valeur doit être une <a href="#">instanceof</a> du parent de la classe dans laquelle la déclaration de type est utilisée. Peut uniquement être utilisé dans les classes.	
array	La valeur doit être un tableau (array).	
<a href="#">callable</a>	La valeur doit être un <a href="#">callable</a> valide. Ne peut être utilisé comme déclaration de type de propriété de classe.	
bool	La valeur doit être un booléen (bool).	
float	La valeur doit être un nombre flottant (float).	
int	La valeur doit être un entier (int).	
string	La valeur doit être une chaîne de caractères (string).	
<a href="#">iterable</a>	La valeur doit être soit un array ou une <a href="#">instanceof Traversable</a> .	PHP 7.1.0
object	La valeur doit être un objet (object).	PHP 7.2.0
<a href="#">mixed</a>	La valeur peut être de n'importe quelle valeur.	PHP 8.0.0

Ce tableau se trouve sur la documentation officiel de PHP. On y voit les types.

(Exos)

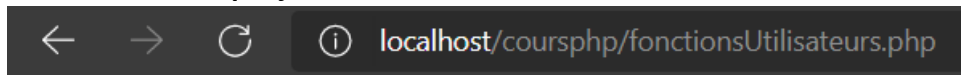
10) (difficile) Créez moi une première fonction « estFemme » qui vérifie si vous êtes un homme ou une femme qui **retourne** un **booléen**.

Créez une deuxième fonction « estAdulte » qui vérifié si vous êtes majeur ou mineur et qui **retourne** un **booléen**.

Créez tableau de tableau indexé/associatif de 4 utilisateurs, avec comme clé (prénom, sexe et âge). Je veux une femme et un homme mineur et je veux une femme et un homme majeur.

Créez une troisième fonction qui reçoit **un seul** tableau indexé en paramètres qui affiche un message en fonction de si c'est une femme ou un homme et si c'est majeur ou mineur. **ICI je veux que vous utilisiez les deux fonctions précédentes pour vérifier dans cette troisième fonction.**

Voici le résultat que je veux :



Sarah est une femme majeur(e) agé(e) de 35 ans.

Eric est un homme mineur(e) agé(e) de 15 ans.

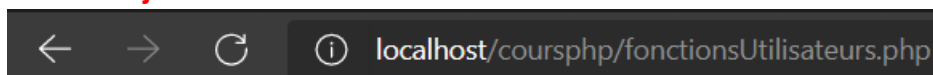
Elisabeth est une femme mineur(e) agé(e) de 17 ans.

Karim est un homme majeur(e) agé(e) de 45 ans.

11) (Extrêmement difficile pour les gens en fusée) Reprenez le même tableau d'utilisateurs et les 2 mêmes fonctions que l'exercice précédent. Modifier la troisième fonction pour qu'au final quand c'est :

- homme majeur il dit « Bonjour Monsieur Prenom, vous êtes un adulte âgé de Age ans »
- femme majeure il dira « Bonjour Madame Prenom, vous êtes une adulte âgée de Age ans »
- homme mineur il dira « Bonjour Prenom, vous êtes un mineur âgé de Age ans »
- femme mineure il dire « Bonjour **Mademoiselle** Prenom, vous êtes une mineure âgée de Age ans »

**Attention j'insiste sur les « e » .**



Bonjour Madame Sarah, vous êtes une adulte âgée de 35 ans.

Bonjour Eric, vous êtes un mineur agé de 15 ans.

Bonjour Mademoiselle Elisabeth, vous êtes une mineure âgée de 17 ans.

Bonjour Monsieur Karim, vous êtes un adulte agé de 45 ans.