

Les fonctions

Définition :

Une fonction est un bout de code **isolé** qui n'est pas interprété dans le flux de chargement de la page mais interprété uniquement lorsque la fonction est **appelée**. Une fonction peut admettre ou non des **paramètres**, typés ou non. On parle alors de la **signature** de la fonction.

Exemple :

```
function direBonjour($personne) {  
    $monTexte = "Bonjour, " . $personne . "<br>";  
    return $monTexte;  
}
```

Puis l'appel de cette même fonction :

```
echo direBonjour("Titouan");
```

Le périmètre (ou scope)

Nous avons dit plus haut qu'une fonction était un bout de code **isolé**. Il faut entendre par là qu'une fonction n'a pas accès aux variables déclarées en dehors de la fonction, et que depuis le reste du code nous n'avons pas accès aux variables déclarées par la fonction. Nous appelons ce principe le **périmètre** de la fonction, ou le **scope** en anglais. Cependant il existe des alternatives pour contourner ce problème.

Nous avons deux cas :

- [Accéder à des données extérieures depuis la fonction :](#)

Dans ce cas de figure, on va vouloir transmettre à notre fonction un accès à une donnée issue d'un autre endroit dans le code. La méthode pour faire ceci consiste à passer à notre fonction des **paramètres**.

Déclarés dans la **signature** de la fonction, ces paramètres vont donner à la donnée extérieure un **nouveau nom** qui sera utilisé uniquement dans le **scope** de la fonction. Par exemple :

```
function calculerMoyenne(array $listeNotes) {  
    $sum = array_sum($listeNotes);  
    $moyenne = $sum / count($listeNotes);  
    echo $moyenne;  
}  
  
calculerMoyenne($notes);
```

Ici, mon tableau **\$notes**, est ce que je vais passer, venu de l'extérieur, à ma fonction, mais à l'intérieur de ma fonction il s'appellera **\$listeNotes**

- Accéder à des données de la fonction depuis l'extérieur

Dans ce cas de figure, on va vouloir récupérer le résultat de ce qu'a fait la fonction pour s'en servir autre part, sauf que la fonction ne nous laisse pas accéder aux variables qu'elle a déclarées. Heureusement il existe une syntaxe précise pour « renvoyer » une variable ou plusieurs dans un tableau, la syntaxe **return**.

Exemple :

```
function calculerMoyenne($tableauDeNotes) {  
    $nbNotes = count($tableauDeNotes);  
    $sommeNotes = 0;  
    foreach($tableauDeNotes as $note) {  
        $sommeNotes = $sommeNotes + $note;  
    }  
  
    return round($sommeNotes / $nbNotes, 2);  
}  
echo calculerMoyenne($notes);
```

Ici on « renvoie » le résultat d'un calcul vers l'extérieur de la fonction. Ce résultat étant renvoyé, on peut simplement utiliser **echo** pour l'afficher.

Attention : Le code situé après un return dans la fonction n'est pas exécuté !