# Passage de paramètres par valeur et par référence

Considérez une fonction qui intervertit les valeurs de deux variables envoyées en paramètre:

function echanger ($valX,$valY){

    $temp=$valX;

    $valX=$valY;

    $valY=$temp;

}

Si vous exécutez ce code dans un script:

$x=5;

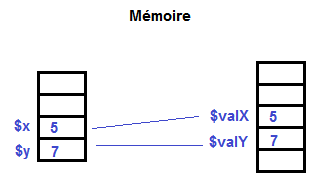
$y=7;

echanger($x,$y);

echo "<br>".$x." ".$y; // 5 et 7

vous observerez que les valeurs de $x et $y n'ont pas été intervertis. Pourquoi?

Les $valX et $valY **sont de copies de $x et $y, mais ne sont pas les variables originales**.



Ça a comme conséquence que les modifications de $valX et $valY n'affectent pas $x et $y (les variables dont on a copie les valeurs):

function echanger ($valX,$valY){

    $temp=$valX;

    $valX=$valY;        // je modifie $valX mais pas $x

    $valY=$temp;        // je modifie $valY mais pas $y

    // les variables meurent ici

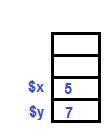
}

## Considérons le code principal:

1. Au début (code principal) il existe que $x et $y

$x=5;

$y=7;



1. Quand on appelle **echanger($x,$y);**

On fait une copie de $x et $y dans $valX et $valY. $x et $y sont indépendantes de $valX et $valY

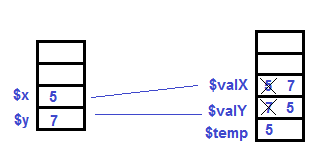


1. On fait les opérations dans la fonction:

$temp = $valX;

    $valX = $valY;         // je modifie $valX mais pas $x

    $valY = $temp;        // je modifie $valX mais pas $x

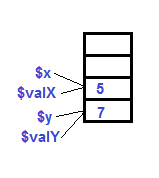


1. La fonction termine et $x et $y contiennent les valeurs originales… seulement $valX et $valY ont échangé ses valeurs et elles n'existent plus

## Solution

On a besoin d'indiquer à PHP d'utiliser de références aux valeurs originales au lieu de réaliser de copies des valeurs. Il suffit de rajouter un "**&**" **devant chaque paramètre dans la déclaration de la fonction pour que PHP ne fasse pas de copies des valeurs mas de références**. Tant la référence comme la variable original pointent vers la même valeur dans la mémoire. La modification d’une implique la modification de l’autre et vice versa :

Si je fais $x=100, $valX vaudra aussi 100…



function echanger (&$valX, &$valY){

    $temp=$valX;

    $valX=$valY;        // je modifie $valX et aussi $x

    $valY=$temp;        // je modifie $valY et aussi $y

}

Les variables $x et $y sont créés dans le script principal. Elles ne sont **pas accessibles depuis la fonction**. Pareillement, les paramètres $valX et $valY ne sont pas accessibles depuis le script principal.

$valX et $valY sont cette fois de références de $x et $y, un changement de valeur dans une (ex: $valX) entraine le changement dans l'autre (ex: $x) et vice-versa. A la fin de la fonction, les références $valX et $valY sont détruites, mais on conserve $x et $y.