


**LES PARAMÈTRES ET  
ARGUMENTS DES FONCTIONS,  
LE PASSAGE DES ARGUMENTS  
ET LES VALEURS DE RETOUR  
D'UNE FONCTION**

A thick, wavy yellow line runs vertically along the left side of the image, starting from the top and extending to the bottom. It has a slightly irregular, hand-drawn appearance.

# **LES PARAMÈTRES ET ARGUMENTS DES FONCTIONS**

# I/A QUOI ÇA SERT ?

*Les paramètres et arguments sont utilisés pour spécifier à la fonction une ou plusieurs valeurs (variable ou non) dont elle a besoin à son bon fonctionnement.*

```
$var1 = 1;

function maFonction(){
    echo $var1;
}

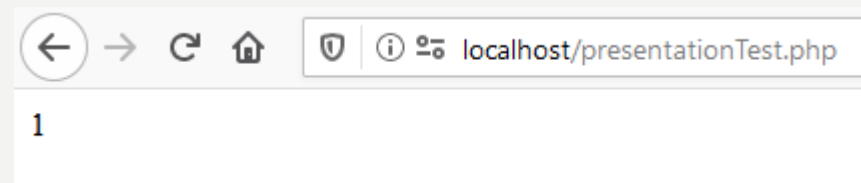
maFonction();
```

```
$var1 = 1;

function maFonction($monParametre){
    echo $monParametre;
}

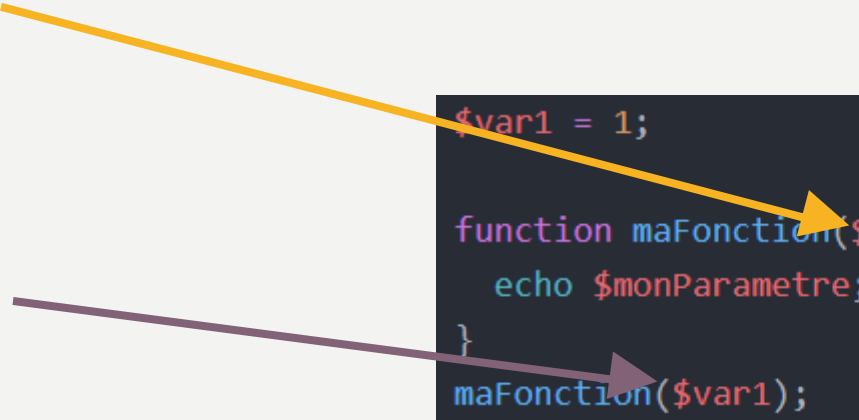
maFonction($var1);
```

(!) Notice: Undefined variable: var1 in C:\wamp64\www\presentationTest.php on line 17				
Call Stack				
#	Time	Memory	Function	Location
1	0.0008	406528	{main}()	...\presentationTest.php:0
2	0.0008	406528	maFonction()	...\presentationTest.php:19



# C'EST QUOI UN PARAMÈTRE OU UN ARGUMENT ?

- Les paramètres d'une fonction sont les noms listés dans la définition de la fonction.
- Les arguments d'une fonction sont les valeurs réelles passées à la fonction.
- Les paramètres sont initialisés avec les valeurs des arguments fournis.



```
$var1 = 1;  
  
function maFonction($monParametre){  
    echo $monParametre;  
}  
  
maFonction($var1);
```

The diagram illustrates the relationship between function parameters and arguments. A yellow arrow points from the text 'paramètres' in the first bullet point to the parameter '\$monParametre' in the function definition. A purple arrow points from the text 'arguments' in the second bullet point to the argument '\$var1' in the function call.

# II/COMMENT ÇA MARCHE ?

```
$var1 = 1;
$var2 = 'Bonjour';
$var3 = array(1,2,3);
$var4 = true;

function maFonction($monParametre, $monParametre2, $monParametre3,
$monParametre4){
    echo $monParametre.'</br>';
    echo $monParametre2.'</br>';
    print_r($monParametre3);
    echo '</br>'. $monParametre4;
}

maFonction($var1,$var2,$var3,$var4);
```



Changer la valeur d'un paramètre ne modifie pas la valeur de l'argument l'extérieure de la fonction.

## Type et nombre

Type de variable : string, int, float, bool, array, const, Objet

Nbr de paramètre : 1 ou +

```
localhost/presentationTest.php

1
Bonjour
Array ( [0] => 1 [1] => 2 [2] => 3 )
1
```

```
$var1 = 1;

function maFonction($monParametre){
    $monParametre -= 100;
    echo $monParametre; // Resultat = -99
}

maFonction($var1);
echo $var1; // Resultat = 1
```

# III/ALLER PLUS LOIN

- Déclarations de type

*Permet d'obliger un paramètre à accepter un type en particulier.*

```
function maFonction(array $monParametre){  
    print_r($monParametre);  
}
```



## Typage strict

Par défaut, PHP va convertir les mauvais types vers le type attendu tant que possible.

- Valeur par défaut des arguments

Permet de définir des valeurs par défaut pour les arguments.

```
$var1 = 1;

function maFonction($monParametre=2){
    print_r($monParametre);
}

maFonction($var1); //Resultat = 1
maFonction(); //Resultat = 2
```

Obligatoirement une constante.

❌ variable, classe, appel de fonction.

Si plusieurs paramètres sont utilisés, le ou les paramètres par défaut doivent être écrit après:

```
function maFonction($monParametre,$monParametre2=2){
    //...
}
```

- Passage d'arguments par référence

Permet de changer la valeur des arguments, avec & avant le paramètre.



```
$var1 = 1;

function maFonction($monParametre){
    $monParametre -= 100;
    echo $monParametre; // Resultat = -99
}

maFonction($var1);
echo $var1; // Resultat = 1
```



```
$var1 = 1;

function maFonction(&$monParametre){
    $monParametre -= 100;
    echo $monParametre; // Resultat = -99
}

maFonction($var1);
echo $var1; // Resultat = -99
```



A thick, wavy yellow line runs vertically along the left side of the image, starting from the top and ending at the bottom. It has a slightly irregular, hand-drawn appearance.

# **LES VALEURS DE RETOUR D'UNE FONCTION**

# I/A QUOI ÇA SERT ?

Permet à une fonction de renvoyer une valeurs.

Exemple: « is\_dir() » « isset() »

## II/Comment ça marche ?

```
$var1 = 1;

function maFonction(&$monParametre){
    return $monParametre -= 1;
}

echo maFonction($var1); // echo 0
$result = maFonction($var1); // $result = 0
```

***Merci de votre attention***

