

Développement web en PHP - Fonctions -

Groupe des étudiants : CIR2

Ayoub KARINE

E-Mail : ayoub.karine@isen-ouest.yncrea.fr

Numéro de bureau : A214

Déclaration et appel

- Déclaration de la fonction :

- Syntaxe

```
function Nom_fonction(argument1, argument2,...) {  
    Instructions;  
    return valeur; // valeur renvoyée par la fonction  
}
```

- Exemple

```
1 function increment($x, $y) {  
2     $z = $x + $y;  
3     return $z;  
4 }
```

- Appel de la fonction :

- Syntaxe

```
Nom_fonction(argument1, argument2,...);
```

- Exemple

```
1 $res = increment(8, 5); // Ici res vaut 13.
```

Arguments d'une fonction

- **Argument par défaut** : il est possible d'affecter des valeurs par défaut aux arguments

```
function test($var, $texte='Essai') {  
    Instructions...;  
    return xxx ;  
}
```

- **Argument typés** : il est possible de forcer le type d'argument

```
function test(string $var) {  
    $val = substr($var, 0, 2);  
    return $val ;  
}
```

Fonctions anonymes et variables

- **Fonctions anonymes** = fonctions sans noms

```
$salut = function($nom)
{
    echo "Hello $nom \n";
};

$salut('World');
```

- **Fonctions variables** = Un nom de variable suivi de parenthèses → recherche d'une fonction portant ce nom

```
function test($texte='Essai') {
    echo $texte ;
}

$func = 'test' ;
$func('ceci est mon texte') ;
```

Portée des variables

- La portée des variables à l'intérieure d'une fonction modifie la visibilité de cette variable dans l'ensemble du code :
 - **Locale (à la fonction)** : visible uniquement à l'intérieur de la fonction
 - **static** : persiste (dans la fonction) durant la durée du script appelant
 - **global** : visible à l'intérieur et à l'extérieur de la fonction

```
$chaine = "Nombre de camions : ";
```

```
function ajoute_camion() {  
    global $chaine;  
    static $nb = 0;  
    $nb++;  
    echo $chaine.$nb."<br>";  
}
```

```
ajoute_camion();// => donne 1  
ajoute_camion();// => donne 2  
ajoute_camion();// => donne 3  
ajoute_camion();// => donne 4
```

Passage des variables

- Le passage des variable se fait de 2 façons :
 - **Par copie** : la fonction n'impacte pas la valeur de la variable hors de la fonction

```
$nbTest = 5;

function ajoute_test($maVal) {
    $maVal = $maVal + 10;

    echo $maVal."<br>";
}

ajoute_test($nbTest); // => donne 15
echo $nbTest."<br>"; // => donne 5
```

Passage des variables

- Le passage des variable se fait de 2 façons :
 - **Par copie** : la fonction n'impacte pas la valeur de la variable hors de la fonction
 - **Par référence** : la fonction impacte la valeur de la variable

```
$nbTest = 5;

function ajoute_test(&$maVal) {
    $maVal = $maVal + 10;

    echo $maVal."<br>";
}

ajoute_test($nbTest);// => donne 15
echo $nbTest."<br>";// => donne 15
```

Bonne pratique

- Plusieurs fonctions sont déjà définies en PHP. Il faut apprendre à se référer, à chaque fois à la documentation officielle

<http://php.net/manual/fr/>



Fonctions d'affichage

- `echo` : s'utilise avec ou sans parenthèses et affiche ses paramètres ;
- `print` : identique à `echo`, sauf qu'un seul paramètre est permis ;
- `printf` : fonctionne comme la fonction C de même nom ;
- `print_r` : affiche la valeur d'une variable ;

```
1 $tab = array('a', 'b');  
2 print_r($tab); // Affiche : Array([0]=>a [1]=>b)
```

- `var_dump` : affiche des informations complètes à propos d'une variable.

```
1 $tab = array('a', 'b');  
2 var_dump($tab); // Affiche :  
3                // array(2){[0]=>string(1) "a"  
4                //                [1]=>string(1) "b"}
```

Fonctions sur les chaînes de caractères

- `strlen($string)` : renvoie la longueur de \$string ;
- `str_word_count($string)` : renvoie le nombre de mots dans \$string ;
- `strrev($string)` : renvoie la chaîne inverse de \$string ;
- `strpos($string, $substring)` : renvoie la position de \$substring dans \$string (false si non trouvé) ;
- `str_replace($word1, $word2, $string)` : renvoie la chaîne \$string où \$word1 à été remplacé par \$word2 ;
- `substr($string, $pos, $len)` : renvoie la sous-chaîne démarrant à la position \$pos et de longueur \$len ;
- `strtolower($string)` : renvoie la chaîne \$string en minuscules ;
- `strtoupper($string)` : renvoie la chaîne \$string en majuscules ;
- `intval($string)` : renvoie le nombre correspondant à la chaîne \$string.
- `intval(string $haystack , mixed $needle)` : trouve la première occurrence dans une chaîne

<http://php.net/manual/fr/book.strings.php>

Fonctions sur les tableaux

- **Fonctions de tri :**
 - `sort` : tri, selon les valeurs, un tableau indicé par des entiers ;
 - `asort` : tri, selon les valeurs, un tableau indicé par des clés ;
 - `ksort` : tri, selon les clés, un tableau indicé par des clés.
- **Fonctions de modification :**
 - `array_shift` : retourne le premier élément d'un tableau et le supprime
 - `array_pop` : retourne le dernier élément d'un tableau et le supprime ;
 - `array_push` : ajoute un (des) élément(s) à la fin d'un tableau.
- **Fonctions sur les clés et les valeurs :**
 - `array_keys` : renvoie sous forme de tableau l'ensemble des clés ;
 - `array_values` : renvoie sous forme de tableau l'ensemble des valeurs ;
 - `array_key_exists` : renvoie VRAI si une clé existe ;
 - `in_array` : renvoie VRAI si une valeur existe.