Chapitre 4 : Les fonctions

- Pas de vérification de type mais (contrairement à Javascript,) il faut passer **au moins** le nombre d'arguments de la définition!
- Les noms de fonction ne sont pas sensibles à la case mais respectez le cleanCode
- Le php lit les définition de fonctions situées au niveau globale avant de débuter l'exécution. ce qui permets donc d'utiliser des fonctions définies plus bas.
- Lorsque php rencontre une nouvelle définition de fonction, celle-ci est définie au niveau global

4.1 Appel de la fonction

```php
nom_de_fonction($param1, $param2);
```

4.2 Créer une fonction std

```php
function nom ($var[,$var]){
return expr ;
}
```

4.3 Créer une fonction anonyme

```php
$double = function ($x) {return $x*2}
```

4.4 Redéfinition et surcharge

ATTENTION : **PAS DE REDEFINITION** sinon erreur fatale

4.5 Cas particuliers

4.5.1 Valeurs par défaut

Les valeurs dont les arguments ont des valeurs par défaut doivent êtres placées en **bout de liste** des arguments.

```
function maFonct($var, $var =  20){
 // ...
}
```

4.5.2 Passage par référence

Les objets sont automatiquement passés par référence. Pour les autres types, il faut utiliser la notation &

```
mafonction($var, &$var)
```

4.5.3 Renvoyer plusieurs valeurs

On utilisera un tableau

```
function test(){
return array($somme,$produit); // mets les variables dans un tableau
}
list($somme,$produit) = test(); //décompose le tableau et les places dans les 2 variables
```

4.5.4 Fonctions a nombre d'arguments variables

au sein d'une fonction :

- ***func_num_args()*** : nombre d'arguments transmis à la fonction
- ***func_get_arg(i)*** : le ieargument (en comptant à partir de 0)
- ***func_get_args***: tous les arguments (tableau)

4.6 Le mot clé static

Si il est utilisé à l'intérieur d'une fonction, la variable déclarée ne sera initialisée que la première fois

```
static $nbModif = 0; // dans une fonction, 0 ne sera juste utilisé que quand la variable sera init
```

> utile pour sauvegarder (ex : nombres de passages sur un site web)

4.7 Les Fonctions Magiques

Le php possède des fonctions dites 'magiques', ces fonctions sont des raccourcis qui facilitent l'écriture. Il en existe dans le cadre de la création d'objet, leur destruction, et bien d'autres. en voici quelques une qui pourraient être utiles :

- *__toString*() : Affiche un objet (appelé automatiquement lors d'une utilisation comme **string** (ex : concaténation ou echo))
- *__invoke($arg)* : L'objet comme fonction. Il est appelé automatiquement si on utilise l'objet comme une **fonction.**
- *__set(nomProp,valeur)* : Appel automatique si crée une propriété inexistante / inaccessible à un objet
- *__get($nom)* : Appel auto quand on essaye de lire une propriété inaccessible d'un objet
- *__unset($propriete)* : Appel auto lors de la suppression d'une propriété d'un objet.

> les Fonctions magiques sont reconnaissables avec leur _ _ devant le nom de fonctions

4.8 Portée des variables

Les variables globales ne sont pas forcément accessibles en locales

```
function maFonction($var){
    globale $var_globale1,$va_globale2;
}
```

Il est aussi possible de passer les variables globales par :

- passage comme argument (si on veut juste accéder à la valeur sans la modifier)
- référence à la fonction (si on veut pouvoir la modifier)