

# 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 débiter l'exécution. ce qui permet donc d'utiliser des fonctions définies plus bas.

Lorsque php rencontre une nouvelle définition de fonction, celle-ci est définie au niveau global

```
function f () {  
    function g() { echo'fonction g'; }  
    echo'fonction f';  
}  
g();          // produit une erreur : g pas définie  
f();  
g();          // fonctionne correctement
```

## appel de la fonction

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

## Créer une fonction std

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

## Créer une fonction anonyme

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

## Redéfinition et surcharge

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

## Cas particuliers

### Valeurs par défaut

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

### Passage par référence

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

### 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
```

## 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)

## 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;
```