

b. **round (nombre [, précision])**: permet d'arrondir des nombres.

```
<?php
    echo round(3.4); // Affiche 3
    echo round(3.5); // Affiche 4
    echo round(1.95583, 2); // Affiche 1.96
?>
```

c. **ceil (nombre)**: arrondi supérieur.

```
echo ceil(7.2); // Affiche 8
echo ceil(99.999); // Affiche 100
```

d. **floor (nombre)**: arrondi inférieur.

```
echo floor(7.2); // Affiche 7
echo floor(99.999); // Affiche 99
```

e. **rand([min,max])**: retourne un nombre aléatoire entre min et max

```
rand(5,15) //retourne un nombre entre 5 et 15
```

f. **sqrt(nombre)**: retourne la racine carré du nombre.

g. **pow(nombre1,nombre2)**: retourne nombre1 à la puissance nombre2

Application:

Effectuer une suite de tirages de nombres aléatoires jusqu'à obtenir une suite composée d'un nombre pair suivi de deux nombres impairs.

4. Fonctions sur les chaînes de caractères:

a. **strlen(\$str)**: retourne la taille de la chaîne \$str en caractères.

b. **strtolower(\$str)**: conversion en minuscules.

c. **strtoupper(\$str)**: conversion en majuscules.

d. **ucfirst(\$str)**: convertit la première lettre en majuscule

e. **trim(\$str)**: suppression des espaces de début et de fin de chaîne

f. **substr(\$str,\$i,\$j)**: retourne une sous chaîne de \$str de taille \$j et débutant à la position \$i.

g. **strcmp (\$str1, \$str2)**: comparaison de 2 chaînes.

h. **ord(\$char)**: retourne la valeur ASCII du caractère \$char

i. **chr(nbr)**: retourne le caractère qui correspond au code ASCII du nbr

j. **strpos(\$str1,\$str2)**: retourne la position de \$str2 dans \$str1. La valeur FALSE est renvoyée si la chaîne n'est pas trouvée.

Application:

Créer deux variables \$nom et \$adresse, la première initialisée par votre nom et prénom et la deuxième par votre adresse mail, transformer ensuite \$nom dans des casses différentes afin que chaque mot ait une initiale en majuscule.

Afficher chaque lettre de la variable \$nom sur une ligne différente.

Vérifier à la fin que \$adresse contient une adresse mail valide.

TP4: FONCTIONS ET FONCTIONS PREDEFINIS

OBJECTIFS :

1. Comprendre l'utilisation des fonctions et des procédures.
2. Les fonctions prédéfinies en PHP.

EXERCICE 1: CREATION DES FONCTIONS ET DES PROCEDURES.

```
//par défaut le mode de passage est par valeur
function NomFonction($arg1, $arg2,...)
{
    //Liste des instructions;
    [return ($valeur_de_retour);]
}
Pour le passage par référence (la valeur initiale peut être
modifiée dans la fonction), il faut utiliser le symbole &.
```

Application:

1. En utilisant une fonction **Factoriel** qui retourne le factoriel d'un entier créer une deuxième fonction **Cardinal** qui calcule le cardinal (n, p)

$$C_n^p = \frac{n!}{p!(n-p)!}$$

2. Ecrire et tester une fonction **Somme** qui calcule et retourne la somme des éléments d'un tableau d'entiers.
3. Ecrire une fonction **MinMax** qui retourne la position de la valeur min et max d'un tableau d'entiers.

EXERCICE 2: LES FONCTIONS PREDEFINIS

1. **include(fichier)**: permet d'inclure un fichier (une autre page) dans la page (L'instruction include sera remplacée par le contenu de la page demandée).
2. **include_once**: inclut et évalue le fichier spécifié durant l'exécution du script une seule fois. Si dans la page on trouve deux commandes **include_once** du même fichier alors la deuxième commande sera ignorée.

Application:

Créer un fichier **Entete.php** composé par un traitement d'affichage de votre nom et prénom, et en utilisant les commandes include et include_once, afficher votre nom et prénom dans un deuxième fichier **MaPage.php**

3. Fonctions numériques:

- a. **max (arg1,arg2[, ...])**: retourne la plus grande valeur numérique.

```
echo max(1, 3, 5, 6, 7); // 7
echo max(array(2, 4, 5)); // 5
```

```
//Tableau des adresses mail
$tab=array("php5@free.com","jean5@fiscali.fr","abc@waladoo.fr",
,"saleh@waladoo.fr","paul@fiscali.fr","darine@fiscali.fr");
```

Le résultat obtenu ici est le suivant :

```
Fournisseur d'accès : free.com = 16.67 %
Fournisseur d'accès : fiscali.fr = 50 %
Fournisseur d'accès : waladoo.fr = 33.33 %
```

6. Fonctions sur les dates:

a. **date (format)** : retourne une date sous forme d'une chaîne, au format donné.

Caractère	Description	Exemple
d	Jour du mois, sur deux chiffres (avec un zéro initial)	01 à 31
g	Heure, au format 12h, sans les zéros initiaux	1 à 12
G	Heure, au format 24h, sans les zéros initiaux	0 à 23
h	Heure, au format 12h, avec les zéros initiaux	01 à 12
H	Heure, au format 24h, avec les zéros initiaux	00 à 23
i	Minute avec les zéros initiaux	00 à 59
j	Jour du mois sans les zéros initiaux	1 à 31
m	Mois au format numérique, avec zéros initiaux	01 à 12
M	Mois, en trois lettres, en anglais	Jan à Dec
n	Mois sans les zéros initiaux	1 à 12
s	Seconde avec zéros initiaux	00 à 59
W	Numéro de la semaine dans l'année	42
Y	Année à quatre chiffres	2004
y	Année à deux chiffres	04
Z	Jour de l'année	342

```
// Aujourd'hui, le 12 April 2010, 10:16:18 am
$aujourdhui = date("m.d.y"); // 04.12.10
$aujourdhui = date("j, m, Y"); // 12, 4, 2010
$aujourdhui = date("H:i:s"); // 10:16:18
$aujourdhui = date("d/m/y"); // 12/04/10
$aujourdhui = date("d/m/Y"); // 12/04/2010
```

- b. **checkdate (mois, jour, année)** : renvoie TRUE si la date représentée par le jour, le mois et l'année donnés en paramètres est valide, FALSE sinon.
- c. **time()** : retourne le timestamp actuel. Un timestamp représente le nombre de secondes écoulées entre la date actuel et le premier janvier 1970).
- d. **strtotime (date)** .prend en argument une chaîne de caractères représentant la date et retourne le timestamp équivalent.

```
echo strtotime("2008/01/12");--> 1200092400
$date1 = "2008/3/7";
$date2 = "2008/12/7";
if ($date1 < $date2)--> Résultat erroné comparaison avec code ASCII
if (strtotime($date1) < strtotime($date2)) Résultat valide
```

5. Fonctions sur les tableaux:

- a. **count (\$T)** et **size (\$T)** : retourne le nombre des éléments du tableau **\$T**.
- b. **print_r (\$T)** : permet d'afficher un tableau d'une manière lisible.
- c. **in_array (\$var, \$T)** : teste si **\$var** existe dans le tableau **\$T**
- d. **sort (\$T)** : trie alphanumérique les éléments du tableau et réaffecte les indices du tableau (Si le tableau est associatif, les clés sont alors supprimées).
- e. **asort (\$T)** : Le tri porte sur les valeurs du tableau et permet de conserver l'association clé/valeur à l'issue du tri..
- f. **ksort (\$T)** : Le tri s'effectue cette fois sur les clés, tout en maintenant l'association avec les valeurs.
- g. **unset (\$var)** : permet de supprimer des cellules d'un tableau (sert aussi à supprimer des variables).

```
$tab = array("a","b","c");
unset($tab[1]); // -> tab = "a","c"
```

NB: il existe des fonctions permettant de réaliser les mêmes tris mais en ordre inverse, il suffit d'ajouter "r" dans leur nom : **rsort()**, **krsort()**, **arsort()**.

- h. **array_shift()** : retourne le premier élément du tableau et le supprime
- i. **array_pop()** : retourne le dernier élément du tableau et le supprime
- j. **array_unshift()** : ajoute un ou plusieurs éléments au début du tableau
- k. **array_push()** : ajoute un ou plusieurs éléments à la fin du tableau

```
$tab = array("a","b","c");
$elem = array_shift($tab); // --> b,c //elem = a
$elem = array_pop($tab); // --> b // elem = c
array_unshift($tab,"d"); // --> d,b
array_push($tab,"e","f"); // --> d,b,e,f
```

- l. **explode()** : coupe une chaîne en fonction d'un séparateur, elle retourne un tableau de chaînes, chacune d'elle étant une sous-chaîne du paramètre string extraite en utilisant un séparateur.

```
$pizza = "piece1 piece2 piece3 piece4 piece5 piece6";
$pieces = explode(" ", $pizza);
echo $pieces[0]; // piece1
echo $pieces[1]; // piece2
```

Application:

1. Créer et remplir un tableau T de 25 entiers aléatoires compris entre 0 et 20, et pour chaque entier k de [0, 20], compte et affiche le nombre d'occurrences de k dans T.

Voici le contenu du tableau :

```
17 3 9 11 6 2 4 6 7 6 19 0 5 14 4 11 12 1 9 15 0 11 20 5 8
Nombre d'occurrences de chaque entier entre 0 et 20 :
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20
2 1 1 1 2 2 3 1 1 2 0 3 1 0 1 1 0 1 0 1 1
```

2. Créez un tableau contenant une liste d'adresses e-mail. Extrayez le nom de serveur de ces données, puis réalisez des statistiques sur les occurrences de chaque fournisseur d'accès.

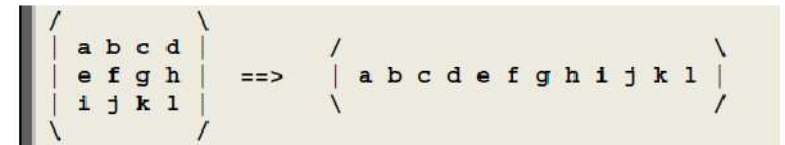
7. Tableaux multidimensionnels :

Application 1:

1. Créer et afficher une matrice carrée binaire aléatoire.
2. Calculer et afficher le pourcentage des 1 et des 0.

Application 2:

1. Ecrire un programme qui transfère un tableau M à deux dimensions L et C dans un tableau V à une dimension L*C.



Application 3:

1. Écrivez un tableau multidimensionnel associatif dont les clés sont des noms de personne et les valeurs des tableaux associatifs dont les clés sont le prénom, la ville de résidence et l'âge de la personne avec une série de valeurs associées.
2. Utilisez une boucle **foreach** pour afficher le contenu du tableau.

Le résultat obtenu est le suivant :

```

Element Dupont :
prenom :Paul
ville :Paris
age :27
Element Schmoll :
prenom :Kirk
ville :Berlin
age :35
Element Smith :
prenom :Stan
ville :Londres
age :45

```