

# Développement PHP

## Partie 2 : Bases de PHP

DENIS LOEUILLET – IFA - 2017

# Partie 2 : Bases de PHP

- Variables, types de variables
- Les instructions conditionnelles
- Les boucles
- les types de données composés : tableaux et objets
- Les fonctions
- Les constantes

# Partie 2 : Bases de PHP

## Les fonctions

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

- En PHP, on n'aime pas avoir à répéter le même code :
  - Le développeur est fainéant!!!!
- Pour pallier ce problème, nous avons découvert les boucles, qui permettent d'exécuter des instructions un certain nombre de fois.
- Ici nous allons découvrir un autre type de structure indispensable pour la suite :
  - les fonctions.
- Comme les boucles, les fonctions permettent d'éviter d'avoir à répéter du code PHP que l'on utilise souvent.
- Mais alors que les boucles sont de bêtes machines tout juste capables de répéter deux cents fois la même chose :
  - les fonctions sont des robots « intelligents » :
    - ✓ qui s'adaptent en fonction de ce que vous voulez faire
    - ✓ qui automatisent grandement la plupart des tâches courantes.

# Partie 2 : Bases de PHP

## Les fonctions

1. Qu'est-ce qu'une fonction?
2. Les fonctions prêtes à l'emploi de PHP
3. Créer ses propres fonctions

# Partie 2 : Bases de PHP

## Les fonctions

### 1. Qu'est-ce qu'une fonction?

- C'est une série d'instructions :
  - qui effectue des actions
  - et qui retourne une valeur.

En général, dès que vous avez besoin d'effectuer des opérations un peu longues dont vous aurez à nouveau besoin plus tard, il est conseillé de vérifier s'il n'existe pas déjà une fonction qui fait cela pour vous.

Et si la fonction n'existe pas, vous avez la possibilité de la créer.



# Partie 2 : Bases de PHP

## Les fonctions : qu'est-ce qu'une fonction?

- Dialogue avec une fonction
  - Voici le genre de dialogue qu'on peut avoir avec une fonction :
    - ✓ Toi, la fonction « calculCube », donne-moi le volume d'un cube dont l'arête mesure 4 cm.
    - ✓ La fonction effectue les calculs demandés et répond :
      - ❖ Ce cube a un volume de 64 cm<sup>3</sup>
  - On donne en entrée à la fonction un paramètre sur lequel elle va faire des calculs (ici, la longueur de l'arête : 4)
  - La fonction nous retourne en sortie le résultat : 64.
  - Grâce à la fonction, vous n'avez pas eu besoin de vous souvenir de la manière dont on calcule le volume d'un cube.
  - Si vous aviez eu à déterminer le volume du cube une seule fois, vous auriez pu écrire le calcul à la main. Mais si vous aviez à le faire 5 fois ? 10 fois ? 100 fois ?

# Partie 2 : Bases de PHP

## Les fonctions : qu'est-ce qu'une fonction?

- Appeler une fonction

- Comment ?

- ✓ Par son nom

```
1 <?php
2 calculCube();
3 ?>
```

- La fonction « calculCube » est une fonction imaginaire : elle n'existe pas (à moins qu'on la crée nous-mêmes). Par conséquent, n'essayez pas d'exécuter ce code PHP il ne fonctionnera pas

- Syntaxe

- ✓ On écrit simplement écrit le nom de la fonction

- ✓ Suivi de parenthèses vides

- ✓ Puis de l'inévitable point-virgule.

- En faisant cela, j'appelle la fonction « calculCube » mais je ne lui envoie aucune information, aucun paramètre.

- Certaines fonctions peuvent fonctionner sans paramètres, mais elles sont assez rares.

- Dans le cas de « calculCube », ça n'a pas de sens de l'appeler sans lui donner la longueur de l'arête du cube pour faire le calcul !

# Partie 2 : Bases de PHP

## Les fonctions : qu'est-ce qu'une fonction?

- Appeler une fonction

- Syntaxe

- ✓ Si on veut lui envoyer un paramètre (un nombre, une chaîne de caractères, un booléen...), il faut l'écrire entre les parenthèses :

```
1 <?php
2 calculCube(4);
3 ?>
```

- ✓ « calculCube » saura qu'elle doit travailler avec le nombre 4
- ✓ Souvent, les fonctions acceptent plusieurs paramètres. Vous devez dans ce cas les séparer par des virgules :

```
1 <?php
2 fonctionImaginaire(17, 'Vert', true, 41.7);
3 ?>
```

- ✓ Cette fonction recevra quatre paramètres :

- 17
- le texte « Vert »
- le booléen vrai
- et le nombre 41,7.



# Partie 2 : Bases de PHP

## Les fonctions : qu'est-ce qu'une fonction?

- Récupérer la valeur de retour de la fonction
  - Nous savons :
    - Appeler une fonction
    - Lui envoyer plusieurs paramètres
  - Il faut maintenant récupérer ce que la fonction nous retourne si toutefois elle retourne quelque chose
    - ✓ Il y a en effet deux types de fonctions :
      - ❖ celles qui ne retournent aucune valeur (ça ne les empêche pas d'effectuer des actions) ;
      - ❖ celles qui retournent une valeur.
    - ✓ Si la fonction ne retourne aucune valeur, il n'y a rien de plus à faire que dans les codes précédents. La fonction est appelée, fait son travail, et on ne lui demande rien de plus.
    - ✓ En revanche, si la fonction retourne une valeur (comme ça devrait être le cas pour « calculCube »), on la récupère dans une variable, comme ceci :

```
1 <?php
2 $volume = calculCube(4);
3 ?>
```

## Partie 2 : Bases de PHP

### Les fonctions : qu'est-ce qu'une fonction?

- Récupérer la valeur de retour de la fonction
  - Il faut maintenant récupérer ce que la fonction nous retourne si toutefois elle retourne quelque chose

```
1 <?php
2 $volume = calculCube(4);
3 ?>
```

- ✓ Sur une ligne comme celle-ci, il se passe en fait les deux choses suivantes (dans l'ordre, et de droite à gauche) :
  1. la fonction « calculCube » est appelée avec le paramètre 4 ;
  2. le résultat renvoyé par la fonction (lorsqu'elle a terminé) est stocké dans la variable \$volume
- La variable \$volume aura donc pour valeur 64 après l'exécution de cette ligne de code !

#### Remarque :

- Il est possible d'envoyer en entrée plusieurs paramètres à une fonction ;
- en revanche cette dernière ne peut retourner qu'une seule valeur.
- Il existe un moyen de contourner cette limitation en combinant des variables au sein d'un array.

# Partie 2 : Bases de PHP

## Les fonctions

### 2. Les fonctions prêtes à l'emploi de PHP

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

PHP propose des centaines de fonctions prêtes à l'emploi

Ces fonctions sont très pratiques et très nombreuses.

C'est en partie là que réside la force de PHP :

- ses fonctions sont vraiment excellentes car elles couvrent la quasi-totalité de nos besoins.

# Partie 2 : Bases de PHP

## Les fonctions : les prêtes à l'emploi de PHP

- Voici un petit aperçu des fonctions qui existent :
  - Une fonction qui permet de rechercher et de remplacer des mots dans une variable.
  - Une fonction qui envoie un fichier sur un serveur.
  - Une fonction qui permet de créer des images miniatures (aussi appelées « thumbnails »).
  - Une fonction qui envoie un mail avec PHP
  - Une fonction qui permet de modifier des images, y écrire du texte, tracer des lignes, des rectangles, etc.
  - Une fonction qui crypte des mots de passe.
  - Une fonction qui renvoie l'heure, la date...
  - Etc...

Dans la plupart des cas, il faudra indiquer des paramètres à la fonction pour qu'elle sache sur quoi travailler.

# Partie 2 : Bases de PHP

## Les fonctions : les prêtes à l'emploi de PHP

- Exemples : Traitement des chaînes de caractères

- strlen : longueur d'une chaîne

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

- ✓ Cette fonction retourne la longueur d'une chaîne de caractères, c'est-à-dire le nombre de lettres et de chiffres dont elle est constituée (espaces compris).

```
1 <?php
2 $phrase = 'Bonjour tout le monde ! Je suis une phrase !';
3 $longueur = strlen($phrase);
4
5
6 echo 'La phrase ci-dessous comporte ' . $longueur . ' caractères :<br />' . $phrase;
7 ?>
```



# Partie 2 : Bases de PHP

## Les fonctions : les prêtes à l'emploi de PHP

- Exemples : Traitement des chaînes de caractères

- `str_replace` : rechercher et remplacer

- <http://php.net/manual/fr/function.str-replace.php>

- ✓ « `str_replace` » remplace une chaîne de caractères par une autre

```
1 <?php
2 $ma_variable = str_replace('b', 'p', 'bim bam boum');
3
4 echo $ma_variable;
5 ?>
```

- On a besoin d'indiquer trois paramètres :

1. la chaîne qu'on recherche (ici, les « b » (on aurait pu rechercher un mot aussi))
2. la chaîne qu'on veut mettre à la place (ici, on met des « p » à la place des « b ») ;
3. la chaîne dans laquelle on doit faire la recherche.

# Partie 2 : Bases de PHP

## Les fonctions : les prêtes à l'emploi de PHP

- Exemples : Traitement des chaînes de caractères

- `str_shuffle` : mélanger les lettres

- <http://php.net/manual/fr/function.str-shuffle.php>

- ✓ « `str_shuffle` » mélange aléatoirement les caractères de votre chaîne

```
1 <?php
2 $chaine = 'Cette chaîne va être mélangée !';
3 $chaine = str_shuffle($chaine);
4
5 echo $chaine;
6 ?>
```

# Partie 2 : Bases de PHP

## Les fonctions : les prêtes à l'emploi de PHP

- Exemples : Traitement des chaînes de caractères
  - strtolower : écrire en minuscules  
<http://php.net/manual/en/function strtolower.php>
  - ✓ « strtolower » met tous les caractères d'une chaîne en minuscules

```
1 <?php
2 $chaine = 'COMMENT CA JE CRIE TROP FORT ???';
3 $chaine = strtolower($chaine);
4
5 echo $chaine;
6 ?>
```

# Partie 2 : Bases de PHP

## Les fonctions : les prêtes à l'emploi de PHP

- Exemples : Récupérer la date
  - Nous allons découvrir la fonction qui renvoie l'heure et la date.
  - Il s'agit de « date »  
<http://php.net/manual/en/function.date.php>
  - Cette fonction peut donner beaucoup d'informations.
  - Voici les principaux paramètres à connaître :

Paramètre	Description
H	Heure
i	Minute
d	Jour
m	Mois
Y	Année

# Partie 2 : Bases de PHP

## Les fonctions : les prêtes à l'emploi de PHP

- Exemples : Récupérer la date

- La fonction « date »

- ✓ Attention ! Respectez les majuscules/minuscules, c'est important !
    - ✓ Si vous voulez afficher l'année, il faut donc envoyer le paramètre Y à la fonction :

```
1 <?php
2 $annee = date('Y');
3 echo $annee;
4 ?>
```



# Partie 2 : Bases de PHP

## Les fonctions : les prêtes à l'emploi de PHP

- Exemples : Récupérer la date
  - La fonction « date »
    - ✓ Si vous voulez afficher la date complète et l'heure :

```
1 <?php
2 // Enregistrons les informations de date dans des variables
3
4 $jour = date('d');
5 $mois = date('m');
6 $annee = date('Y');
7
8 $heure = date('H');
9 $minute = date('i');
10
11 // Maintenant on peut afficher ce qu'on a recueilli
12 echo 'Bonjour ! Nous sommes le ' . $jour . '/' . $mois . '/' . $annee . 'et il est ' . $heure .
    ' h ' . $minute;
13 ?>
```

# Partie 2 : Bases de PHP

## Les fonctions : les prêtes à l'emploi de PHP

- Exemples : Récupérer la date

- La fonction « date »

Remarque :

Si l'heure n'était pas bonne, sachez que c'est le serveur qui donne l'heure, et le serveur peut se situer sur un autre fuseau horaire,

# Partie 2 : Bases de PHP

## Les fonctions

### 3. Créer ses propres fonctions

Bien que PHP propose des centaines et des centaines de fonctions parfois il n'y aura pas ce que vous cherchez et il faudra écrire vous-mêmes la fonction. C'est une façon pratique d'étendre les possibilités offertes par PHP.

Quand écrire une fonction ?

- En général, si vous effectuez des opérations un peu complexes que vous pensez avoir besoin de refaire régulièrement, il est conseillé de créer une fonction

Nous allons découvrir la création de fonctions à travers deux exemples :

- l'affichage d'un message de bienvenue en fonction du nom ;
- le calcul du volume d'un cône.

# Partie 2 : Bases de PHP

## Les fonctions : créer ses propres fonctions

- Premier exemple : afficher un message de bienvenue en fonction du nom

C'est peut-être un peu fatigant de dire bonjour à chacun de ses visiteurs, non ? Ça serait bien de le faire automatiquement... Les fonctions sont justement là pour nous aider !

Regardez le code ci-dessous :

```
1 <?php
2 $nom = 'Sandra';
3 echo 'Bonjour, ' . $nom . ' !<br />';
4
5 $nom = 'Patrick';
6 echo 'Bonjour, ' . $nom . ' !<br />';
7
8 $nom = 'Claude';
9 echo 'Bonjour, ' . $nom . ' !<br />';
10 ?>
```

# Partie 2 : Bases de PHP

## Les fonctions : créer ses propres fonctions

- Premier exemple : afficher un message de bienvenue en fonction du nom  
La même chose avec une fonction :

```
1 <?php
2 function DireBonjour($nom)
3 {
4     echo 'Bonjour ' . $nom . ' !<br />';
5 }
6
7 DireBonjour('Marie');
8 DireBonjour('Patrice');
9 DireBonjour('Edouard');
10 DireBonjour('Pascale');
11 DireBonjour('François');
12 DireBonjour('Benoît');
13 DireBonjour('Père Noël');
14 ?>
```



# Partie 2 : Bases de PHP

## Les fonctions : créer ses propres fonctions

- Premier exemple : afficher un message de bienvenue en fonction du nom  
La même chose avec une fonction :
  - les lignes en haut permettent de définir la fonction :
    - ✓ son nom
    - ✓ ce qu'elle est capable de faire, etc.
    - ✓ Elles ne font rien de particulier, mais elles disent à PHP : « Une fonction « DireBonjour » existe maintenant ».

# Partie 2 : Bases de PHP

## Les fonctions : créer ses propres fonctions

- Premier exemple : afficher un message de bienvenue en fonction du nom

### ➤ Syntaxe :

- ✓ Pour créer une fonction, vous devez :

- ❖ Taper fonction (en français, ça veut dire « fonction »).
- ❖ Ensuite, donnez-lui un nom. Par exemple, celle-ci s'appelle « DireBonjour ».
- ❖ Après, entre parenthèses : il y a une variable.
  - C'est le paramètre dont a besoin la fonction pour travailler, afin qu'elle sache à qui elle doit dire bonjour dans notre cas.
  - Notre fonction doit forcément être appelée avec un paramètre (le nom) sans quoi elle ne pourra pas travailler.
- ❖ Ensuite les accolades :
  - Elles permettent de marquer les limites de la fonction.
  - Cette dernière commence dès qu'il y a un « { » et se termine lorsqu'il y a un « } ».
  - Entre les deux, il y a son contenu.

# Partie 2 : Bases de PHP

## Les fonctions : créer ses propres fonctions

- Premier exemple : afficher un message de bienvenue en fonction du nom

### ➤ Syntaxe :

- ✓ Pour appeler la fonction :

- ❖ Il suffit d'indiquer son nom, et de préciser ses paramètres entre parenthèses.
- ❖ Enfin, il ne faut pas oublier le fameux point-virgule ( ; ) car il s'agit d'une instruction.

Par exemple :

```
1 <?php
2 DireBonjour('Marie');
3 ?>
```

# Partie 2 : Bases de PHP

## Les fonctions : créer ses propres fonctions

- Premier exemple : afficher un message de bienvenue en fonction du nom

### ➤ Exercice :

- ✓ On veut écrire une fonction pour dire bonjour à tout le monde en même temps.
- ✓ Récrire la fonction afin de ne pas avoir à écrire plusieurs fois l'appel de la fonction :

```
DireBonjour ('Marie');  
DireBonjour ('Patrice');
```

...

- ✓ Utiliser un array
- ✓ Utiliser les boucles
- ✓ L'appel de fonction ne doit être écrit qu'une fois

# Partie 2 : Bases de PHP

## Les fonctions : créer ses propres fonctions

- Premier exemple : afficher un message de bienvenue en fonction du nom

➤ Exercice :

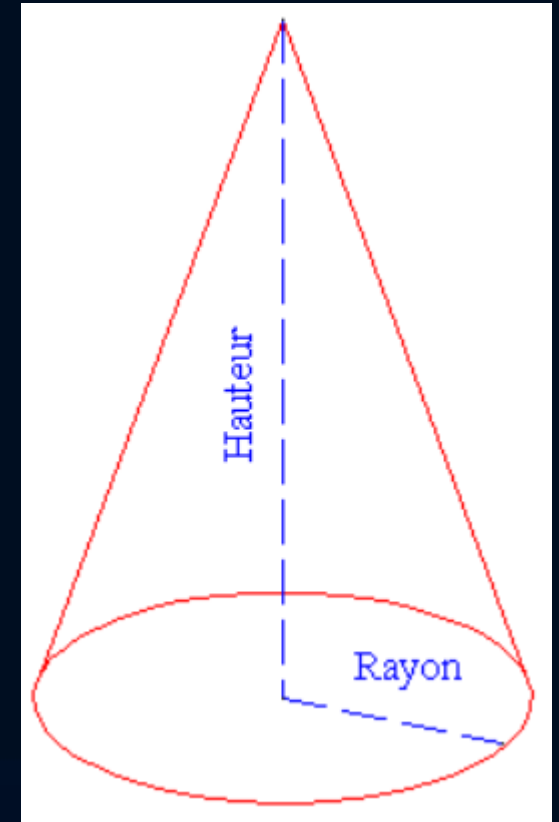
```
function DireBonjour ($array_prenoms) {  
  
    foreach ($array_prenoms as $prenom) {  
  
        echo "Bonkour " . $prenom . " !<br/>";  
  
    }  
}  
  
$prenoms = array ("Marie", "Patrice", "Edouard", "Pascale");  
DireBonjour ($prenoms);
```



# Partie 2 : Bases de PHP

## Les fonctions : créer ses propres fonctions

- Deuxième exemple : calculer le volume d'un cône
  - La fonction précédente ne renvoyait aucune valeur
  - Elle se contentait d'effectuer des actions
  - Nous allons créer une fonction qui renvoie une valeur :
    - ✓ Notre fonction va servir à faire un calcul : le calcul du volume du cône
    - ✓ Le principe est le suivant :
      - ❖ vous donnez à la fonction le rayon et la hauteur du cône
      - ❖ elle travaille et vous renvoie le volume que vous cherchez.
    - ✓ la formule à utiliser est la suivante :
      - $\text{Volume cône} = \text{rayon}^2 \times \pi \times \text{hauteur} \times 1/3$



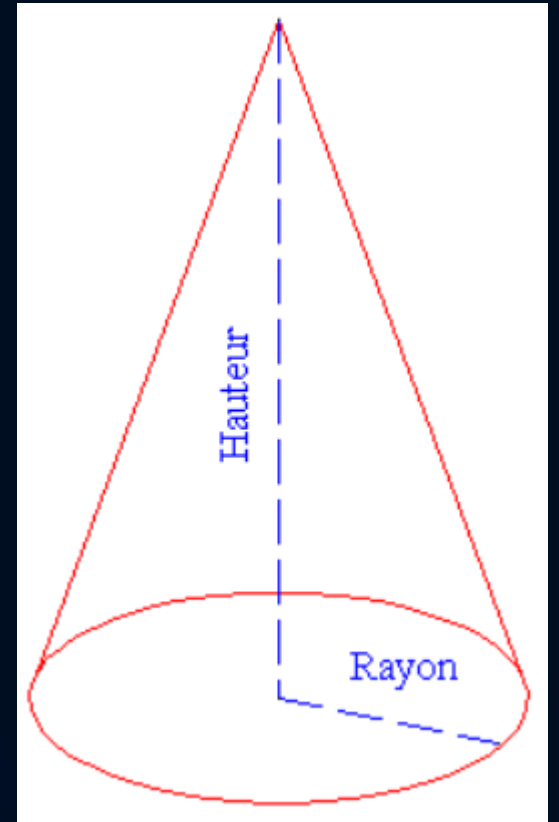
# Partie 2 : Bases de PHP

## Les fonctions : créer ses propres fonctions

- Deuxième exemple : calculer le volume d'un cône
  - Exemple sans fonction :

```
1 <?php
2 // Calcul du volume d'un cône de rayon 5 et de hauteur 2
3 $volume = 5 * 5 * 3.14 * 2 * (1/3);
4 echo 'Le volume du cône de rayon 5 et de hauteur 2 est : ' . $volume . ' cm<sup>3</sup><br>
   />';
5
6 // Calcul du volume d'un cône de rayon 3 et de hauteur 4
7 $volume = 3 * 3 * 3.14 * 4 * (1/3);
8 echo 'Le volume du cône de rayon 3 et de hauteur 4 est : ' . $volume . ' cm<sup>3</sup><br>
   />';
9 ?>
```

✓ Inconvénient de cette méthode ?

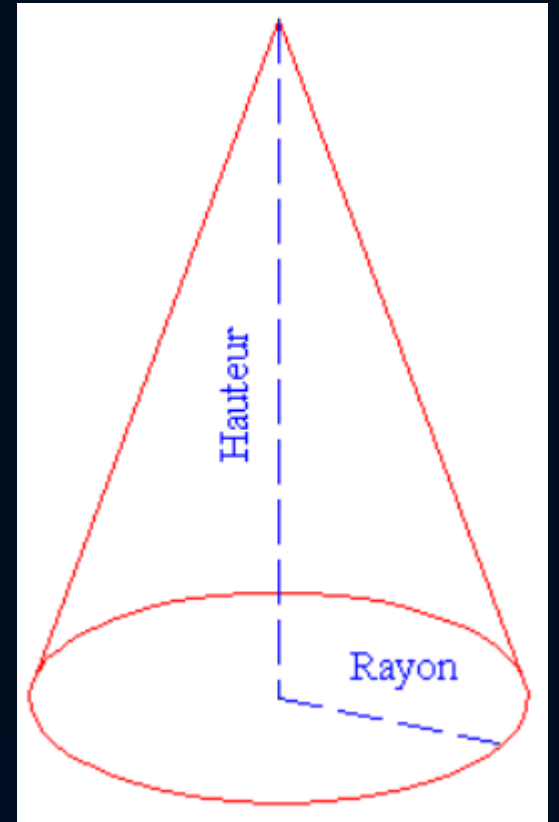


# Partie 2 : Bases de PHP

## Les fonctions : créer ses propres fonctions

- Deuxième exemple : calculer le volume d'un cône
  - Exemple avec fonction :
    - ✓ Son nom : « VolumeCone »
    - ✓ Que fait-elle ?
      - Elle calcul le volume d'un cône
      - Elle n'affiche rien
      - Elle renvoie le volume calculé
    - ✓ Paramètres :
      - Le rayon du cône
      - La hauteur du cône

A vous de jouer!!



## Partie 2 : Bases de PHP

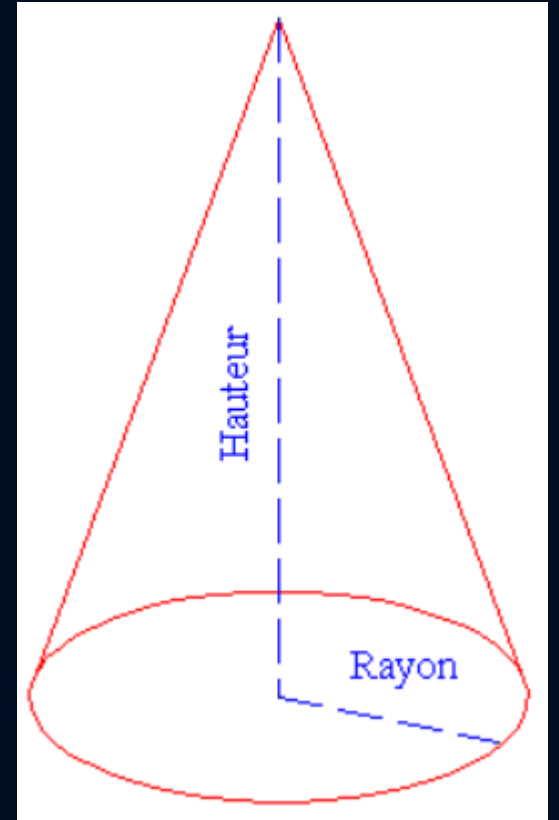
### Les fonctions : créer ses propres fonctions

- Deuxième exemple : calculer le volume d'un cône

➤ Exemple avec fonction :

```
1 <?php
2 // Ci-dessous, la fonction qui calcule le volume du cône
3 function VolumeCone($rayon, $hauteur)
4 {
5     $volume = $rayon * $rayon * 3.14 * $hauteur * (1/3); // calcul du volume
6     return $volume; // indique la valeur à renvoyer, ici le volume
7 }
8
9 $volume = VolumeCone(3, 1);
10 echo 'Le volume d\'un cône de rayon 3 et de hauteur 1 est de ' . $volume;
11 ?>
```

- ✓ Nouveauté : l'instruction « return \$volume »
- ✓ Cette instruction indique ce que doit renvoyer la fonction.
  - Ici, elle renvoie le volume.
- ✓ Si vous aviez tapé return 15, ça aurait à chaque fois affiché un volume de 15
- ✓ Le volume calculé renvoyé est enregistré dans la variable \$volume



# Partie 2 : Bases de PHP

## Les fonctions : créer ses propres fonctions

- Deuxième exemple : calculer le volume d'un cône

### ➤ Exemple avec fonction :

#### ✓ Modifiez la fonction pour :

##### ❖ contrôler les paramètres d'entrée :

###### ○ Contrôle \$rayon :

- \$rayon = 0
- \$rayon pas un int ou pas un float

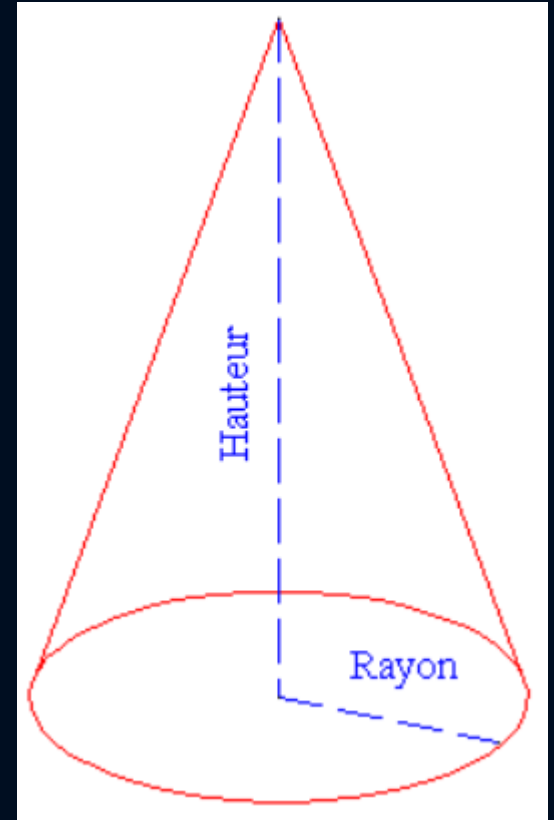
###### ○ Contrôle \$hauteur :

- \$hauteur = 0
- \$hauteur pas int ou pas float

##### ❖ Que se passe-t-il si on ne transmet :

- Aucun paramètre
- Un seul paramètre :
  - Le premier
  - Le deuxième

##### ❖ Modifier le message en conséquence



## Partie 2 : Bases de PHP

### Les fonctions : créer ses propres fonctions

- Deuxième exemple : calculer le volume d'un cône

➤ Exemple avec fonction :

```
function VolumeCone ($rayon, $hauteur) {  
    if (is_numeric($rayon) && $rayon != 0 && is_numeric($hauteur) && $hauteur != 0) {  
  
        $volume = $rayon * $rayon * 3.14 * $hauteur * 1/3;  
  
        return "le volume est de " . $volume . " cm";  
  
    } else {  
        return "erreur dans les paramètres!!!";  
    }  
}  
  
echo VolumeCone (4, 2);
```



# Partie 2 : Bases de PHP

## Les fonctions

- Résumé
  - Les fonctions sont des blocs de code qui exécutent des instructions en fonction de certains paramètres.
  - Les fonctions ont généralement une entrée et une sortie. Par exemple, si on donne la valeur 4 à la fonction de calcul du cube, celle-ci renvoie 64 en sortie.
  - PHP propose des centaines et des centaines de fonctions prêtes à l'emploi pour tous types de tâches : envoyer un e-mail, récupérer l'heure, crypter des mots de passe, etc.
  - Si PHP ne propose pas la fonction dont on a besoin, il est possible de la créer avec le mot-clé « function ».

# Partie 2 : Bases de PHP

## Exercices

- Exercice 1 :  
Ecrire un script qui inverse une chaîne de caractères.  
`$chaine = 'Bonjour tout le monde';`

# Partie 2 : Bases de PHP

## Exercices

- Exercice 1 : solutions

- 2 solutions

- ❖ La première :

```
$chaine = "bonjour tout le monde";  
for ($i=strlen($chaine)-1; $i >= 0 ; $i--) {  
    $chaine_inverse .= $chaine[$i];  
}  
echo $chaine_inverse;
```

- ❖ La seconde :

```
echo strrev($chaine)."<br/>";
```

# Partie 2 : Bases de PHP

## Exercices

- Exercice 2 :  
Rédiger une expression conditionnelle pour tester si un nombre est à la fois un multiple de 3 et de 5.
  - Utiliser les expressions conditionnelles multiples

# Partie 2 : Bases de PHP

## Exercices

- Exercice 2 : solution

```
<?php
$x=1245;
if($x%3==0 && $x%5==0)
{
    echo "$x est multiple de 3 et de 5 <br />";
}
else
{
    echo "$x n'est pas multiple de 3 et de 5 <br />";
}
?>
```

N.B : En pratique un multiple de 3 et de 5 est un multiple de 15 !

# Partie 2 : Bases de PHP

## Exercices

- Exercice 3 :  
Écrire une expression conditionnelle utilisant les variables `$age` et `$sexe` dans une instruction `if` pour sélectionner une personne de sexe féminin dont l'âge est compris entre 21 et 40 ans et afficher un message de bienvenue approprié.
  - Utiliser les expressions conditionnelles multiples



# Partie 2 : Bases de PHP

## Exercices

- Exercice 3 : solution

```
<?php
$sexe="F";
$age=43;
if($sexe=="F" && $age>=21 && $age <= 40)
{
    echo "Bonjour Madame vous avez entre 21 et 40 ans <br />";
}
else
{
    echo "Désolé, vous ne remplissez pas les conditions <br />";
}
?>
```

# Partie 2 : Bases de PHP

## Exercices

- Exercice 4 :  
Effectuer une suite de tirages de nombres aléatoires jusqu'à obtenir une suite composée d'un nombre pair suivi de deux nombres impairs.
  - Utiliser la boucle do ... while
  - Utiliser les expressions conditionnelles multiples

# Partie 2 : Bases de PHP

## Exercices

- Exercice 4 : solution

```
<?php
$compteur=0;
do
{
    $x=rand(0,1000);
    $y=rand(0,1000);
    $z=rand(0,1000);
    $compteur++;
    echo $x, ", ", $y, ", ", $z,"<br />";
}
while($x%2==1 || $y%2==0 || $z%2==0);

echo "Résultat obtenu en " . $compteur. " coups";
?>
```

# Partie 2 : Bases de PHP

## Exercices

- Exercice 5 :  
Choisir un nombre de trois chiffres.  
Effectuer ensuite des tirages aléatoires et compter le nombre de tirages nécessaire pour obtenir le nombre initial.  
Arrêter les tirages et afficher le nombre de coups réalisés.  
Réaliser ce script d'abord :
  - avec l'instruction while
  - puis avec l'instruction for

# Partie 2 : Bases de PHP

## Exercices

- Exercice 5 : solution boucle while

```
<?php
//Nombre à trouver
$nb=789;
//compteur
$coup=0;
//boucle de tirage
while($x!=$nb)
{
    $x=rand(0,1000);
    $coup++;
    //echo $x,"<br />";//pour afficher tous les tirages
}
echo "$nb trouvé en $coup coups ";
?>
```

# Partie 2 : Bases de PHP

## Exercices

- Exercice 5 : solution boucle for

```
<?php
//Nombre à trouver
$nb=789;
//boucle de tirage
for($coup=1;$x!=$nb;$coup++)
{
    $x=rand(0,1000);
    //echo $x,"<br />";//pour afficher tous les tirages
}
echo "$nb trouvé en $coup coups ";
?>
```



# Partie 2 : Bases de PHP

## Exercices

- Exercice 6 :  
Utiliser une boucle while pour déterminer le premier entier obtenu par tirage aléatoire qui soit un multiple d'un nombre donné.  
Écrire la variante utilisant la boucle do...while

# Partie 2 : Bases de PHP

## Exercices

- Exercice 6 : solution boucle while

```
<?php
    $nb=57;
    $compteur=0;
    $x=rand(0,1000);//Boucle while
    while($x%$nb!=0)
    {
        $x=rand(0,1000);
        $compteur++;
    }
    echo $x . " est multiple de " . $nb . ": Résultat obtenu en " . $compteur . " coups";
?>
```

- Avec une boucle while, notez qu'il faut initialiser \$x avant la boucle.

# Partie 2 : Bases de PHP

## Exercices

- Exercice 5 : solution boucle for

```
<?php
    $nb=57;
    $compteur=0;
    do
    {
        $x=rand(0,1000);
        $compteur++;
    }
    while($x%$nb!=0);
    echo $x . " est multiple de " . $nb . ": Résultat obtenu en " . $compteur . " coups";
?>
```

➤ Avec une boucle do...while il n'est pas nécessaire d'initialiser \$x.

# Partie 2 : Bases de PHP

## Exercices

- Exercice 7 :  
Écrivez un tableau multidimensionnel associatif dont les clés sont des noms de personne et les valeurs des tableaux indicés contenant le prénom, la ville de résidence et l'âge de la personne.

# Partie 2 : Bases de PHP

## Exercices

- Exercice 7 : solution

```
<?php
    $tab=array(
        "Dupont">array("Paul","Paris",27),
        "Schmoll">array("Kirk","Berlin",35),
        "Smith">array("Stan","Londres",45)

    );
    print_r($tab);

?>
```

# Partie 2 : Bases de PHP

## Exercices

- Exercice 8 :  
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.



# Partie 2 : Bases de PHP

## Exercices

- Exercice 8 : solution

```
<?php
    $tab=array(
        "Dupont">array(
            "prenom">"Paul",
            "ville">"Paris",
            "age">27),
        "Schmoll">array(
            "prenom">"Kirk",
            "ville">"Berlin",
            "age">35),
        "Smith">array(
            "prenom">"Stan",
            "ville">"Londres",
            "age">45)
    );
    print_r($tab);
?>
```

# Partie 2 : Bases de PHP

## Exercices

- Exercice 9 :  
Utilisez une boucle foreach pour lire les tableaux des exercices 7 et 8

# Partie 2 : Bases de PHP

## Exercices

- Exercice 9 : solution avec tableau exercice 7

```
<?php
$tab=array(
    "Dupont"=>array("Paul","Paris",27),
    "Schmoll"=>array("Kirk","Berlin",35),
    "Smith"=>array("Stan","Londres",45)
);
foreach($tab as $cle =>$valeur)
{
    echo "<b>Elément $cle :</b><br />";
    foreach($valeur as $ind =>$val)
    {
        echo "élément $ind :", $val, "<br />";
    }
}
?>
```

# Partie 2 : Bases de PHP

## Exercices

- Exercice 9 : solution avec tableau exercice 8

```
<?php
$tab=array(
    "Dupont"=>array("prenom"=>"Paul","ville"=>"Paris","age"=>27),
    "Schmoll"=>array("prenom"=>"Kirk","ville"=>"Berlin","age"=>35),
    "Smith"=>array("prenom"=>"Stan","ville"=>"Londres","age"=>45)
);
foreach($tab as $cle=>$valeur)
{
    echo "<b>Element $cle :</b><br />";
    foreach($valeur as $cle2=>$val)
    {
        echo " $cle2 :", $val, "<br />";
    }
} ?>
```