COURS 06 : DEFINIR DES FONCTIONS

Introduction

Les fonctions sont mises au point par des développeurs, bénévoles dans le cas de projets libres et open source, ou payés par des éditeurs. Certaines sociétés sont spécialisées dans la conception de bibliothèques de fonctions hautement réutilisables (communication téléphonique, interface utilisateur, accès aux bases de données ...)

Toutefois les fonctions disponibles résolvent des problèmes généraux, et le développeur est très souvent, appelé à résoudre des problèmes particuliers. Certains problèmes sont même récurrents dans l'entreprise. Le développeur d'entreprise conçoit alors des bibliothèques de fonctions maisons.

Définition d'une fonction

En php, une fonction peut être définie :

- Dans le même fichier que le programme principal. Dans ce cas, on place la définition de la ou des fonctions en tête du script.
- Dans un autre fichier. Dans ce cas, il faut faire apparaître le lien vers cet autre fichier en tête de script à l'aide de l'instruction **include.** Remarque : le nom du fichier n'a aucune importance. Mais par simplicité, s'il ne contient qu'une seule fonction, donnez lui le nom de la fonction. Cette solution permet d'avoir une fonction indépendante de tout programme et ainsi pouvoir l'utiliser autant de fois que l'on désire. Elle permet également d'alléger le code source du programme principal.

RÈGLES DE NOMMAGE D'UNE FONCTION

Ce sont les mêmes que pour une variable. N'utilisez que des caractères simples.
Bannissez les espaces, les _ / ; @ # -...
Commencez par une minuscule et mettez une majuscule pour chaque nouveau mot.
Exemple : rechercheMax
Donnez des noms significatifs à vos fonctions.
Vous n'avez pas le droit d'utiliser le nom d'une fonction déjà définie. Exemple : max

COMMENT DÉFINIR UNE FONCTION

En PHP, la définition d'une fonction débute avec le mot clef : **function**. Les instructions définissant le corps de la fonction sont délimitées par les accolades ouvrantes et fermantes{}.

L'instruction de retour d'un résultat est **return**. PHP n'étant pas un langage typé, vous n'avez pas à préciser dans l'entête de la fonction, le type des paramètres attendus et le type du résultat. Mais prenez l'habitude, quand vous définissez une fonction, d'écrire son interface (signature) avec les types sous forme de commentaire, reproduisant ainsi l'aide en ligne de PHP. Les règles de construction d'une fonction ou procédure déjà énoncées (variables locales, passage de paramètre...) sont valables quel que soit le langage de programmation.

Exemple 1 :

```
// double rechercheMax ( double val1, double val2)
// affiche le plus grand nombre des 2 variables double passées en paramètres
   function rechercheMax ($val1, $val2)
    {
      if ($val1 > $val2)
            $max = $val1;
     else
            $max = $val2;
     return $max ;
    }
```

Utilisation

```
print ( max(10,15) );       // affiche 15
```

Exemple 2 :

```
// afficheMax ( double val1, double val2)
    function afficheMax ($val1, $val2)
    {
      if ($val1 > $val2)
           $max = $val1;
     else
           $max = $val2;
      print( $max ) ;
    }
```

Utilisation

```
afficheMax(10,15) ;       // affiche 15
```

Remarque : redéfinir des fonctions existantes peut s'avérer être un bon exercice, mais il ne faut surtout pas que vous préfériez l'utilisation de vos propres fonctions à celles déjà définies.
Pourquoi ? car la fonction a été testée par de nombreux autres développeurs avant vous, et qu'elle a fait l'objet de multitude améliorations tant en terme de justesse que de performance.

LES PARAMÈTRES

PARAMÈTRE PAR DEFAUT

Vous pouvez définir des paramètres par défaut, ainsi ils deviendront optionnels lors de l'utilisation de la fonction.

Exemple :

```
<?php
 function affichebonjour ($nom = "inconnu")
 {
   echo (« bonjour » . $ nom) ;
 }
                                    Utilisation
 afficheBonjour(« jojo » ) ; // affiche bonjour jojo
 afficheBonjour( ) ; // affiche bonjour inconnu
?>
```

PASSAGE DE PARAMÈTRE

Le passage de paramètre est un passage par valeur mais vous pouvez intervenir en demandant un passage par référence. Pour cela, il vous suffit de placer le symbole & devant vos paramètres.
Exemple :

```
<?PHP
// ajouteUn ( double val1) ajoute 1 à la variale double passée en paramètre
      function ajouteUn (&$val)
      {
        $val = $val + 1 ;
      }

      $a = 3 ;
      print( "Avant ajout, a =" . $a ); // a = 3
      ajouteUn ($a);
      print( "Apres ajout, a =" . $a ); // a = 4
    ?>
```

EXERCICE 1 : CALCUL DES FRAIS DE PORT

Le programme **commandeVin.php** permet de calculer les frais de port en fonctions des règles suivantes :
Un marchand de vin expédie une quantité Q de bouteilles de prix unitaire P. Si le total de la commande est de plus de 200 €, le port est gratuit. Sinon, il est facturé 10% de la commande avec un minimum de 8€ de port. Autrement dit, si les frais de port que vous avez calculé sont inférieurs à 8 €, vous devez facturer 8 €.

Écrivez le programme PHP qui demandera à l'utilisateur de saisir quantité et prix unitaire afin d'afficher le montant du colis, le montant des frais de port et le total de la somme à payer.

```
<?php

 $p = readline("entrer un prix unitaire : \n");
 $q = readline ("entrer une quantite : \n");

 $montant= $p*$q;
 if ($montant>200)
   $port=0;
 else {
         $port=10/100*$montant;
         if ($port<8)
            $port=8;
 }

 print("montant du colis : $montant \n");
 print("frais de port : $port \n");
 print("total à payer : ".($montant+$port)." \n");
?>
```

Écrire une fonction port, qui recevra en paramètre le montant de la commande et qui retournera les frais de port correspondant. Utilisez cette fonction dans votre programme commandeVin.php

EXERCICE 2 : CALCUL DE PRIME

La société "La Généreuse" prévoit de donner à chacun de ses employés une prime de fin d'année.
Les employés ayant plus de 5 années d'ancienneté percevront 10 % de leur salaire, s'ils sont cadres sinon ils percevront 8 % de leur salaire.
Pour satisfaire tout le personnel, la société décide d'accorder une prime de 50 € aux employés ayant moins de 5 ans d'ancienneté.
Le programme **prime.php** permet de calculer le montant de la prime d'un employé.

```
<?php

 $anciennete = readline( "Entrez l'ancienneté du salarié") ;
 $salaire = readline( "Entrez le salaire du salarié") ;
 $statut = readline( "Entrez le statut du salarié (C pour cadre, N pour non cadre)") ;

 if (($anciennete >= 5 ) && ($statut == "C"))
   $prime =0.05 * $salaire ;
 if (($anciennete >= 5 ) && ($statut != "C"))
   $prime =0.03 * $salaire ;
 if ($anciennete < 5 )
   $prime = 50 ;
 print("la prime est de ". $prime."€ \n");
?>
```

Écrire une fonction prime, qui recevra en paramètre le salaire, l'ancienneté et le statut du salarié et qui retournera la prime correspondante. Utilisez cette fonction dans votre programme prime.php

EXERCICE 3 : NUMÉRO DE SECURITE SOCIALE

Concevez une fonction : isNumSecuOk qui reçoit en paramètre une chaîne de caractères représentant un numéro de sécurité sociale. Cette fonction retournera un booléen :

- TRUE si le paramètre contient exactement 13 caractères et commence soit par 1, soit par un 2.
- FALSE sinon

1. Définissez la fonction en pensant à donner sa signature (interface) en commentaire.

Remarques :

- True et False sont des mots clefs, ne mettez jamais de « » !
- vous pouvez utiliser les fonctions strlen et substr :

int strlen (string string) strlen retourne la taille de la chaîne string .
string substr (string string , int start , int length) substr retourne le segment de string défini par start et length .

2. Utilisez votre fonction dans un programme principal afin de la tester.
Demandez un numéro à l'utilisateur puis utilisez votre fonction afin d'indiquer

- Si le « numéro est correct »
- Si le « numéro est incorrect »

3. Dans votre programme principal, demandez à l'utilisateur un numéro tant que celui qu'il a saisi n'est pas correct.