

PHP

05 - Les fonctions

I. Définition

Une fonction est une partie de code PHP qui peut remplir n'importe quelle tâche. Dans ce code, peut figurer toute instruction PHP valide. Une fonction est destinée à effectuer une tâche spécifique prédéfinie.

On rencontre 2 types de fonctions :

- Les fonctions **natives** = celles proposées par défaut par le langage; en PHP il en existe [des centaines](#).
- Les fonctions **utilisateurs** = celles que le développeur écrit, et qui peuvent bien entendu intégrer des fonctions natives.

Dans l'exemple ci-dessous, nous utilisons la fonction `ucfirst()`. Cette fonction permettra de passer le premier caractère de la chaîne en majuscule :

```
$texte = ucfirst("Bonjour le monde");
```

La variable `$texte` contiendra alors la valeur : *Bonjour le monde*. Le premier caractère est passé en majuscule. L'utilisation des fonctions présente de très nombreux avantages parmi lesquels se trouve la non répétition d'une séquence de code car cette opération est souvent la cause d'importants dysfonctionnements.

Ces fonctions permettent également une meilleure lisibilité du code source, un gain de productivité ainsi qu'une maintenance facilitée.

1. Définir une fonction

Pour déclarer (= définir) une fonction, on utilisera le mot clé `function` suivi du nom de la fonction (à vous de lui donner un nom) et enfin de parenthèses (obligatoires, mêmes vides) pour accueillir d'éventuels paramètres. Exemple d'une fonction simple :

```
function bonjour()  
{  
    echo "Bonjour";  
}
```

On appellera donc la fonction de la manière suivante :

```
<?php  
bonjour();  
?>
```

Cette fonction affichera Bonjour. On pourrait également ajouter un nom ou prénom comme ceci :

```
function bonjour($prenom)
{
    echo "Bonjour ".$prenom;
}
```

On appellera la fonction avec l'instruction suivante :

```
<?php
bonjour("Dave");
```

Cette fonction affichera *Bonjour Dave*. On peut également utiliser plusieurs paramètres entre les parenthèses. Exemple :

```
function bonjour($prenom, $nom)
{
    echo "Bonjour ".$prenom." ".$nom;
}
```

On appellera la fonction avec l'instruction suivante :

```
<?php
bonjour("Dave", "LOPER");
```

Cette fonction affichera donc *Bonjour Dave LOPER*. En pratique, les valeurs indiquées entre les parenthèses sont les arguments de la fonction.

2. Les librairies de fonctions

Une librairie de fonctions est un ensemble de fonctions regroupées dans un seul fichier. En fait, les fonctions que nous créons ne sont disponibles que dans le programme dans lequel elles se trouvent. Ainsi, pour les utiliser dans un autre programme, il faudrait les recopier dans celui-ci. Mais, il existe bien sur un moyen de rendre les fonctions disponibles dans tous nos programmes. Il suffit pour cela, de regrouper vos fonctions dans un fichier et d'ajouter une ligne à tous vos programmes afin d'inclure ce fichier et ainsi rendre disponible toutes les fonctions pour ce programme. En pratique, il faut créer un fichier contenant toutes les fonctions.

Concrètement :

```
<?php
function 1...
...
function 2...
...
?>
```

Enregistrer le fichier sous un nom quelconque que vous choisirez. Par exemple : *lib.inc.php*

Vous pourrez ajouter à ce fichier autant de fonctions que vous aurez besoin pour vos applications. Ensuite, nous ajouterons l'instruction require dans chaque programme où nous aurons besoin des fonctions :

```
<?php
require('lib.inc.php');
...suite de l'application...
?>
```

II. Les grands groupes de fonctions PHP

1. Les fonctions arithmétiques

Le langage PHP offre un jeu de fonctions mathématiques assez complet qui couvre largement les besoins que l'on peut avoir pour réaliser une application. Composition :

- opérateurs mathématiques (addition, soustraction, multiplication, division, puissance...)
- arrondis, valeur absolue, minimum, maximum, ...
- cosinus, sinus, tangente, logarithmes...
- génération de nombres aléatoires
- conversions (binaire, décimal, hexadécimal...)
- formatage de nombres (en chaîne de caractères, ...)

2. La gestion des chaînes de caractères

PHP fournit un ensemble complet de fonctions pour gérer les chaînes de caractères :

- comparer des chaînes de caractères
- définir la longueur d'une chaîne de caractères
- découper une chaîne, extraire des parties
- crypter des chaînes/données
- remplacer des parties de chaînes
- rechercher des composants d'une chaîne de caractères
- convertir des caractères spécifiques en code HTML

3. La gestion des fichiers et répertoires

Les fonctions de gestion de fichiers et de répertoires fournies par PHP vous permettront :

- de créer, modifier, supprimer, renommer et déplacer des fichiers de tout type (texte, csv, images, PDF...)
- de créer, modifier (nom), supprimer, renommer et déplacer des répertoires
- de lire, ajouter, modifier et supprimer du contenu d'un fichier
- de lire et modifier les informations de métadonnées (dates de création/modification/accès, nom, type, poids, droits d'exécution...) d'un fichier ou d'un répertoire

4. Les fonctions de génération d'images dynamiques

Une librairie de fonctions de création et de manipulation d'images est fournie avec PHP. Cette librairie se nomme GD (pour *Gif Draw*).

GD permet de nombreuses possibilités :

- générer dynamiquement des images (formes géométriques, miniatures, graphiques...)
- gestion des couleurs dans une image
- transmettre des informations sur une image (largeur x hauteur),
- découper et faire pivoter une image (rotation selon un angle précis)

- incrustation de texte, superposition/fusion d'images (par exemple, ajouter un logo dans une image pour montrer qu'elle provient d'un site)
- manipuler les différents formats d'images (gif, png, jpeg, tiff, raw...)

[Les bases de GD et quelques exemples](#)

5. Les fonctions spécifiques à Internet

Le PHP, conçu pour Internet, propose bien sûr un groupe de fonctions spécifiques à celui-ci :

- fonctions de traitement d'URLs
- fonctions de lecture d'une page HTML (avec la librairie *CURL* notamment)
- la protection d'accès à une page web : sessions, cryptage de mots de passe que nous aborderons plus tard dans le programme
- envoi de mails (séquence complète)
- fonctions de gestion de cookies

III. Quelques fonctions utiles

Fonction	Description	Code PHP	Résultat
<code>addslashes()</code>	Ajoute des anti-slashes devant les caractères spéciaux	<code>\$s = addslashes("L'a");</code>	<code>L\'a</code>
<code>stripslashes()</code>	Retire les antislashes devant les caractères spéciaux	<code>\$s = stripslashes("L\'a");</code>	<code>L'a</code>
<code>ceil()</code>	Retourne le nombre entier supérieur ici (12,1)	<code>\$s = ceil("12.1");</code>	13
<code>chunk_split()</code>	Permet de scinder une chaîne en plusieurs morceaux	<code>\$s = chunk_split("DGDFEF", "2", "-");</code>	<code>DG-DF-EF-</code>
<code>htmlentities()</code>	Remplace les caractères par leur équivalent HTML (si ils existent).	<code>\$s = htmlentities("&");</code>	<code>&</code>
<code>strstr()</code>	Recherche le premier caractère 'p' dans la chaîne et affiche le reste de la chaîne y compris le 'p'	<code>\$s = strstr("webmaster@php.cdsi.fr.st", "p");</code>	<code>php.cdsi.fr.st</code>

Fonction	Description	Code PHP	Résultat
<code>strlen()</code>	Retourne la longueur de la chaîne	<code>\$s = strlen("lachainedecaracteres");</code>	20
<code>strtolower()</code>	Passe tout les caractères en minuscules	<code>\$s = strtolower("LA CHAINE dE caRActERes");</code>	<i>la chaine de caracteres</i>
<code>strtoupper()</code>	Passe tout les caractères en majuscules	<code>\$s = strtoupper("LA CHAINE dE caRActERes");</code>	LA CHAINE DE CARACTERES
<code>str_replace()</code>	Remplace un caractère par un autre dans une chaîne (tient compte de la casse)	<code>\$s = str_replace("a", "o", "Lalala");</code>	<i>Lololo</i>
<code>trim()</code>	Efface les espaces dits blancs (espaces, sauts et retours de lignes au début et à la fin d'une chaîne)	<code>\$s = trim(" Bonjour le monde ");</code>	<i>Bonjour le monde</i>
<code>ucfirst()</code>	Met la première lettre d'une chaîne en majuscule	<code>\$s = ucfirst("bonjour le monde");</code>	<i>Bonjour le monde</i>
<code>ucwords()</code>	Met la première lettre de chaque mot d'une chaîne en majuscule	<code>\$s = ucwords("bonjour le monde");</code>	<i>Bonjour Le Monde</i>
<code>strpos()</code>	Recherche la position du premier caractère trouvé. Retourne le nombre de caractères placés avant lui (ici 4).	<code>\$s = strpos("abcdef", "e");</code>	4
<code>preg_match()</code>	Recherche si une chaîne de caractère est contenue dans une autre (ex. recherche si <i>ABCDE</i> contient <i>BCD</i>). Peut rechercher via des expressions régulières	<code>if (preg_match ("BCD", "ABCDE")) { echo "Trouvé"; }</code>	Renvoie 1 (TRUE) si trouvé, 0 (FALSE) sinon