

Les fonctions utilisateurs

PHP propose de nombreuses fonctions, mais un autre avantage est de pouvoir créer les siennes à l'aide de `function()`. Ceci est vraiment très utile pour ne pas avoir à retaper des parties de code en entier.

Le code PHP de fonction.php

(ne copier/coller pas ce code dans votre éditeur, retapez-le ou gare aux erreurs...)

```
<?
function Arial($size,$color,$texte)
{
    print("<font face=Arial size=".$size." color=".$color.">".$texte."</font>") ;
}
?>
```

Le code PHP de index.php

```
<?
Require("fonction.php");      // on appelle la page contenant la fonction
// affichage -----
Arial("2","red","Ici le texte ...");
Arial("3","#0F74A3","Le second texte ...");
?>
```

On appelle `index.php` dans le navigateur, ce qui donne à l'écran

Ici le texte ...Le second texte ...

Reprenons depuis le début. On distingue donc deux étapes lors de l'utilisation de fonctions. La première consiste à créer votre fonction en lui donnant un nom, en choisissant les paramètres dont elle aura besoin et déterminer ce qu'elle doit faire. Une fois cela fait, vous pouvez appeler votre fonction. Plus précisément : entre parenthèses ce sont les arguments que vous entrerez lors de l'utilisation de cette fonction. Ces arguments correspondent bien sûr à ceux qui se trouvent entre parenthèses à la création de la fonction. Ici, il s'agit juste de donner la taille "`$size`", la couleur "`$color`" et le contenu du texte "`$texte`". L'utilisation de vos fonctions se font de la même manière que les fonctions intégrées de PHP (`print` et `echo` en l'occurrence). Dans `Index.php`, vous voyez que l'on appelle la fonction `Arial()` et qu'on lui donne les valeurs que l'on veut, pratique non ?

Pratique : Je vous conseille de créer un fichier `fonction.php` qui va contenir toutes vos fonctions de texte, tableau, etc. Il suffit juste ensuite de mettre un `require()` en tête de chacun de vos autres fichiers php pour pouvoir utiliser toutes les fonctions.

Je pense que vous voyez de suite l'intérêt de créer ses propres fonctions. Cela permet une bien meilleure lecture du code et un gain de temps à la programmation.

Dans un premier temps nous en resterons là, mais nous reviendrons à des fonctions plus complètes au cours des exercices futurs.