PHP Atelier 1 : Bibliothèque de fonctions

Imaginons que nous voulions utiliser une fonction `writeMessage()` dans toutes les pages d'un site : il faudrait mettre le code de cette fonction dans chacune des pages. Imaginez alors pour un site de 1000 pages : ce n'est clairement pas possible en termes de maintenabilité du code, car il faudrait reporter 1000 fois la moindre modification effectuée dans le code de la fonction `writeMessage()`.

Pour résoudre ce problème, PHP offre un mécanisme : l'**inclusion de fichiers**. On parle alors de *fichier externe*.

Inclusion de fichiers externes

Créez le fichier PHP suivant, appellons-le `index.php` :

```
<?php
// Fichier 'index.php'

function writeMessage($sText)
{
   $html = "<h1>".$sText."</h1>";
    echo $html;
}
?>
<!DOCTYPE html>
<html lang="fr">
<head>
    <meta charset="utf-8">
    <title>Inclusion de fichiers PHP</title>
    <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
    <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.0/css/bootstrap.min.css" integ
</head>
<body>
<?php
writeMessage($sMessage);
?>
<br>
<?php
writeMessage("Bonjour tout le monde !");
?>
<script src="https://code.jquery.com/jquery-3.5.1.slim.min.js" integrity="sha384-DfXdz2htPH0lsSSs5nCTpuj/zy4C+OG
<script src="https://stackpath.bootstrapcdn.com/bootstrap/4.5.0/js/bootstrap.min.js" integrity="sha384-OgVRvuATP
</body>
</html>
```

Tout d'abord, nous allons déplacer la fonction `writeMessage()` dans un second fichier PHP nommé *functions.php*. Ce fichier *functions.php* sera un fichier de bibliothèque (ou encore *librairie*) de code, avec pour seul contenu le code de la fonction `writeMessage()` :

```
<?php
// Fichier 'functions.php'

function writeMessage($sText)
{
   $sHtml = "<h1>".$sText."</h1>";
    echo $sHtml;
}
```

Il s'agit donc de factoriser à un seul emplacement le code des fonctions utilisées dans plusieurs pages, cela rejoint la définition même d'une fonction qui est d'être réutilisable.

> On pourra bien entendu par la suite ajouter autant de fonctions que nécessaire dans notre fichier *functions.php*.

Dans le fichier d'origine *index.php*, on peut maintenant supprimer le code de la fonction `writeMessage()` et le remplacer par l'inclusion (chargement ou appel) du fichier *functions.php* via la fonction PHP native `include()` qui prend en argument le chemin vers le fichier et son nom :

```
include("functions.php");
```

Cette fonction `include()` permet de recopier dans la page le contenu du fichier dont l'URL est passée en paramètre.

Il suffit donc de recopier cette ligne dans toutes les pages où nous voulons utiliser notre bibliothèque de fonctions personnelle.

> ATTENTION :

PHP permet au développeur de créer et de manipuler ses propres fonctions. Pour illustrer ceci, nous allons encore une fois modifier *index.php* pour définir une fonction d'écriture d'un titre :

```
<?php
 // Fichier 'index.php'

include("functions.php");
$sMessage = "Hello world !";
?>
 <!DOCTYPE html>
<html lang="fr">
<head>
    <meta charset="utf-8">
    <title>Inclusion de fichiers PHP</title>
    <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
    <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.0/css/bootstrap.min.css" integ
</head>
<body>
<?php
writeMessage($sMessage);
?>
<br>
<?php
writeMessage("Bonjour tout le monde !");
?>
<script src="https://code.jquery.com/jquery-3.5.1.slim.min.js" integrity="sha384-DfXdz2htPH0lsSSs5nCTpuj/zy4C+OG
<script src="https://stackpath.bootstrapcdn.com/bootstrap/4.5.0/js/bootstrap.min.js" integrity="sha384-OgVRvuATP
</body>
</html>
```

Découpage d'une page HTML

Non seulement vous allez trouver sur le web des bibliothèques de fonctions libres de droits à inclure dans vos programmes, mais vous allez pouvoir les utiliser pour découper du simple code HTML en plusieurs fichiers.

Par exemple, vous pourriez découper une page HTML de la façon suivante : en-tête, contenu principal et pied de page :

Fichier `index.php`

```
<?php
include("entete.php");
?>
<body>
 page de test
<?php
include("pieddepage.php");
?>
```

Fichier `entete.php`

```
<!DOCTYPE html>
<html lang="fr">
<head>
   <meta charset="utf-8">
   <title>Inclusion de fichiers PHP</title>
   <link rel="stylesheet" href="css/style.css">
</head>
```

fichier `pieddepage.php`

```
<script src="js/scripts.js"></script>
</body>
</html>
```

Les différentes fonctions d'inclusion

PHP fournit 4 fonctions d'inclusion de fichiers :

Fonction	Usage
`include()`	lève une erreur de type avertissement (_warning_), c'est-à-dire qui ne bloque pas l'exécution du code suivant l'appel de la fonction `include()`.
`require()`	lève une erreur dite _fatale_, le script s'arrête PHP là.
`include_once()`	pareil que pour `include()` mais le fichier n'est chargé qu'une seule fois, lors du premier appel dans le site.
`require_once()`	pareil que pour `require()` mais le fichier n'est chargé qu'une seule fois, lors du premier appel dans le site.

Compléments

- Différences entre `include()` et `require()`
- La gestion des erreurs en PHP

Vérification de l'existence d'un fichier

Dans le cadre d'une inclusion de fichier, il faut s'assurer que le fichier à inclure existe bien. Ceci se fait avec la fonction PHP `file_exists()`, qui retourne un booléen, à laquelle on passe le chemin du fichier dont on veut vérifier l'existence :

```
<?php
if (file_exists("functions.php") )
{
    include("functions.php");
}
else
{
     // Erreur (afficher un message ou redirection)
}
```

En argument de `file_exists()`, on peut mettre un chemin, relatif ou absolu) :

```
f (file_exists("../chemin/fichier.php") )
{
    include("../chemin/fichier.php");
}
else
{
     // Erreur (afficher un message ou redirection)
}
```

Exercice

Reprenez une page que vous avez réalisée avec Bootstrap (maquette Jarditou) et découpez-la comme ci-dessus.