Les Médecins Maîtres-Toile

Publié le : 18 avril 2006
Publication antérieure :
22 décembre 2003

Imprimer cet article

Auteur :
Alain Sichel †

Page contact en PHP 1/3

1. Page contact classique

Cet article présente un formulaire pour une page contact en PHP. [1] Il s'agit d'une adaptation de l'article de Jean-Paul Mesters dans ".net" n° 79 novembre 2003 page 75, dont certaines explications sont reprises ici.

Sommaire

L'objectif est de recevoir un message venant d'un formulaire, afin de ne pas laisser traiter son email à disposition des robots spameurs. Un contrôle du formulaire est inclu, vérifiant que les champs obligatoires sont biens remplis et que l'adresse email indiquée a une syntaxe correcte.

Cet article a été modifié le 27 janvier 2005 et le 15 octobre 2005 pour éviter une faille de sécurité permettant le *Cross Site Scripting* [2].

Cette page peut être testée ici : contact, le code de l'ensemble du script peut être vu et copié ici : [3].

Explications du code :

```
<?php
function formulaire($Nom="",$Profession="",$Email="",$Message="") {
?>
```

▸ La fonction initialise les quatres zones du formulaire, l'accolade ouvrante est suivie du code HTML du formulaire. La fonction se termine à l'emplacement de l'accolade fermante après la balise de fermeture du formulaire `</form>`.

```
<form action="contact.php" method="post">
Nom : <input type="text" name="Nom" size="30" value="<?php echo
trim(htmlentities($Nom)) ?>" /><br />
Profession : <input type="text" name="Profession" value="<?php echo
trim(htmlentities($Profession)) ?>" size="30" /><br />
E-mail : <input type="text" name="Email" value="<?php echo
trim(htmlentities($Email)) ?>" size="30" /><br />
Message : <textarea name="Message" alt="Message" rows="10" cols="47"
wrap="virtual"><?php echo trim(htmlentities($Message)) ?></textarea><br />
<input type="submit" name="Submit" value="Envoi" alt="Envoi"  title="Envoi
de votre message" />
</form>
```

▸ Lorsqu'on enfonce le bouton **submit** du formulaire, le contenu les zones est transmis au même fichier php (ici : *"contact.php"*). Les minis scripts insérés au niveau de l'attribut **value** affichent la donnée introduite dans la zone elle-même. Le code présenté ici est simplifié par rapport au fichier de démonstration indiqué plus haut, où le formulaire est présenté dans un tableau et avec des effets de style pour le bouton **Envoi**.

```
<?php
}
if(!isset($Nom)) {
        formulaire();
}
```

▸ L'accolade met fin à la fonction PHP. Si la variable **$Nom** n'existe pas, cela signifie que le formulaire n'a pas encore été complété. La fonction **formulaire()** est alors appelée pour l'afficher. Si la variable **$Nom** existe, les données sont récupérées dans les variables **$vNom**, **$vProfession**, **$vEmail** et **$vMessage**. La fonction **trim** élimine les éventuels espaces avant et après la donnée [4] , la fonction **htmlentities()** évite les failles de sécurité en empêchant le *Cross Site Scripting* [2].

```
else {
        $vNom=trim(htmlentities($_POST["Nom"]));
        $vProfession=trim(htmlentities($_POST["Profession"]));
        $vMessage=trim(htmlentities($_POST["Message"]));
        $vEmail=trim(htmlentities($_POST["Email"]));
        $destinataire='stetho@domaine.com';
        $titre="Modèle de page Contact";
        $message="Provenance : $HTTP_REFERER\n";
        $message.="Adresse IP : $REMOTE_ADDR,\n";
        $message.="Navigateur : $HTTP_USER_AGENT\n";
        $message.="Nom : $vNom\n";
        $message.="Profession : $vProfession\n";
        $message.="E-mail : $vEmail\n";
        $message.="Message : $vMessage\n";
        if (empty($vNom) || empty($Message) || empty($Email)) {
                echo "<p class='red'>Vous n'avez pas complété toutes les
zones :</p>";
```

```
                $erreur=1;
        }
```

A la ligne `$destinataire='stetho@domaine.com';` il faudra remplacer `stetho@domaine.com` par votre email pour personnaliser ce script.

▸ Si l'une des zones du formulaire n'a pas été complétée, le message indiqué s'affiche et la variable **$erreur** prend la valeur **1**. Dans ce cas, la ligne *"Vous n'avez pas complété toutes les zones :"* sera affichée (ici en rouge en fonction de la feuille de style de la page). La variable **$message** prépare le contenu du message qui sera envoyé par Mail. Les lignes sur *"Provenance", "Adresse IP", "Navigateur"* peuvent être sautées si ces informations ne sont pas souhaitées.

```
if (!eregi("^[0-9a-z]([-_.]?[0-9a-z])*@[0-9a-z]([-_.]?[0-9a-z])*\\.[a-
z]{2,4}$",$vEmail) && ($erreur<>1)){
        echo "<p class='red'>L'adresse e-mail n'est pas correcte :</p>";
        $erreur=1;
        }
```

▸ L'adresse e-mail est testée par l'expression régulière **^[0-9a-z]** : le premier caractère doit être une lettre ou un chiffre. **([-_.] ?[0-9a-z])*** : il est suivi par un ou plusieurs caractères (lettres ou chiffres) précédés optionnellement d'un tiret, d'un soulignement ou d'un point. **@** : présence du caratère @. **[0-9a-z]** : un ou plusieurs caractères (lettres ou chiffres). **([-_.] ?[0-9a-z])*** : à nouveau un ou plusieurs caractères (lettres ou chiffres), précédés ou non de l'un des 3 signes entre les crochets. **\\.** : présence d'un point. **[a-z]{2,4}** : présence de 2, 3 ou 4 lettres. **$** : fin de donnée. Si la syntaxe n'est pas correcte, la phrase *"L'adresse e-mail n'est pas correcte :"* est affichée en rouge.

```
if ($erreur==1) {
        formulaire($Nom,$Profession,$Email,$Message);
        }
```

▸ Si l'un des tests précédents est vrai, la variable **$erreur** aura la valeur **1**. Dans ce cas, la fonction formulaire est à nouveau exécutée pour afficher le formulaire.

```
else {
        mail($destinataire,$titre,$message,"From: $vEmail") ;
        echo "<p class='vi4'>Votre message :</p>";
        echo "<ul><li>Nom : <span class='red'>" . $vNom . "</span><br />";
        echo "<li>Profession : <span class='red'>" . $vProfession.
"</span></li>";
        echo "<li>E-mail : <span class='red'>" . $vEmail. "</span></li>";
        echo "<li>Message : <span class='red'>" .
$vMessage.'</span></li></ul>';
        echo "<p class='vi4'>a bien été envoyé au webmestre du site. Nous
vous remercions.</p>\n" ;
        echo "<form name='boutons'><table border='0' cellspacing='0'
cellpadding='0' width='90%' align='center'><tr><td><center>" ;
       echo "<input type='button' name='accueil' alt='Accueil'
value=\"Retour à la page d'accueil\" onclick=\"location.href='/index.php'\"
```

```
/>\n";
       echo "</center></td><td><center><input type='button' name='plan'
alt='Plan du site' value='Vers le plan du
site'  onclick=\"location.href='plan.php'\" /></center></td></tr></table>";
        }
}
?>
</form>
```

- Si tout est bien rempli, le message est envoyé par la fonction **mail()** et le visiteur retrouve la page, cette fois-ci sans le formulaire, mais avec confirmation de l'envoi de son message qui lui est présenté (ici dans une liste à puces, et avec quelques effets de couleurs avec la feuille de style). On propose au visiteur de se rendre ailleurs dans le site (ici par deux boutons présentés dans un tableau, avec des liens vers la page d'accueil et vers le plan du site).

Conclusion

Voilà, tout y est pour une page contact classique, si par contre vous souhaitez des éléments supplémentaires, c'est abordé dans Page contact en PHP 2/3, et pour éviter le spam, l'ajout d'un captcha dans Page contact en PHP 3/3.

---

[1] Evidemment, pour que ce script fonctionne il faut le placer sur une page avec l'extension .php ou .php3, elle-même mise sur un serveur interprétant le PHP. Cette page a été mise au point pour le site cyes.info ; la page choix techniques présente aux webmestres les différentes techniques mises en œuvre pour ce site.

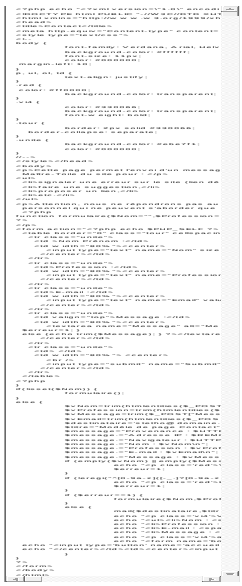
[2] Voir Pourquoi les failles de Cross Site Scripting se ramassent à la pelle et Le Cross Site Scripting.
Les données récupérées par le formulaire passent par la fonction `htmlentities()` qui bloque les attaques en tranformant tout caractère spécial en code HTML. L'ennui, c'est que les caractères accentués sont eux aussi transformés (é devient `&eacute;`), ce qui donne des messages peu lisibles. La nouvelle fonction proposée dans l'article pour le champ "Message" ne bloque que les caractères indésirables, pour ceux qui utilisent l'ancien script, il suffit de remplacer :

```
<?php echo trim(htmlentities($Message)) ?>
```

par :

```
<?php if (ereg("[]%~#`$&|}{^[><]",$Message)) { echo "Certains caractères
utilisés sont interdits";
 $erreur=1; }
 else {echo trim($Message); } ?>
```

[3] Code du script :



L'utilisation de `<?php echo $PHP_SELF ?>` permet de récupérer le nom de la page, sinon il faudrait indiquer "contact.php" dans notre exemple, et modifier ce nom si la page est renommée.

[4] Avant la version 4.12 de PHP, il faut utiliser $HTTP_POST_VARS à la place de $_POST.

Imprimer cet article