

Envoi du contenu d'un formulaire vers un email

Le formulaire de la page Contact est devenu l'incontournable des sites vitrines. Voici quelques solutions pour recevoir des messages.

Contact

Une remarque ? une suggestion ? N'hésitez-pas à m'écrire.

Votre nom (obligatoire)

Votre e-mail (obligatoire)

Sujet

Votre message

Le "mailto"

La formule la plus simple consiste à publier l'email du responsable à contacter. En HTML, on crée un lien vers ce mail avec le protocole *mailto*, comme montré ci-dessous :

```
<a href="mailto:jules@free.fr">Ecrivez-moi</a>
```

Avec l'usage répandu des webmails, la plupart des internautes ne paramètrent plus leurs clients mails comme *Outlook* ou *Thunderbird*, ou pire, certains utilisateurs ne savent pas copier l'email de la rubrique contact dans leur webmail. Il est évident que cela pose problème si le propriétaire du site veut être facilement contacté.

Le formulaire de contact

Une solution efficace est d'expédier le contenu du formulaire vers un email à l'aide de la fonction PHP `mail()`. Autre avantage, votre email n'étant pas directement publié, c'est un atout contre le spam.

Voici un exemple simple pour l'utiliser :

contact.html

```
<!doctype html>
<html>
<head>
  <meta charset="utf-8">
  <title>Envoi d'un message par formulaire</title>
</head>
<body>
  <form method="post" action="mail.php">
    <label>Tape ton message ici pour m'écrire</label>
    <textarea rows="10" cols="30" name="message"></textarea>
    <input type="submit">
  </form>
</body>
</html>
```

Dans ce formulaire, la balise `<textarea>` permet de saisir plusieurs lignes de texte, là où la traditionnelle balise `<input type='text'>` n'autorise qu'une seule ligne de saisie.

mail.php

```
<!doctype html>
<html>
<head>
  <meta charset="utf-8">
  <title>Envoi d'un message par formulaire</title>
</head>
<body>
  <?php
    $retour = mail('jules@free.fr', 'Envoi depuis la page Contact', $_POST['message'], 'From :
webmaster@monsite.fr');
    if ($retour) {
      echo '<p>Votre message a bien été envoyé.</p>';
    }
  ?>
</body>
</html>
```

Dans cet exemple tout simple, le contenu du formulaire sera expédié vers l'email `jules@free.fr` grâce à la fonction `mail()`.

Syntaxe de la fonction `mail()` :

```
mail(chaine_email_destinataire, chaine_Sujet, chaine_corps_du_message, chaine_options);
```

On retrouvera dans la chaîne options, le mot *From*: suivi de l'email de l'expéditeur. Ce qui est amusant, c'est qu'on peut inventer l'email de l'expéditeur. Il peut être réel ou pas. S'il n'existe pas, le destinataire ne pourra évidemment pas répondre sur cet email.

Page contact "tout en un"

Le code ci-dessous fusionne les deux pages précédentes grâce à la fonction `isset()`. Il ajoute également la collecte de l'email de l'internaute pour s'en servir comme adresse d'expédition puis adresse de réponse. Les attributs *required* affectés aux champs de texte forceront l'internaute à renseigner ces champs. Si le navigateur ne gère pas l'attribut *required*, le code PHP vérifiera la présence de l'arobase avec la fonction `strpos()`.

contact.php

```
<!doctype html>
<html>
<head>
    <meta charset="utf-8">
    <title>Contact</title>
</head>

<body>
    <h1>Contact</h1>
    <form method="post">
        <label>Email</label>
        <input type="email" name="email" required><br>
        <label>Message</label>
        <textarea name="message" required></textarea><br>
        <input type="submit">
    </form>
    <?php
    if (isset($_POST['message'])) {
        $position_arobase = strpos($_POST['email'], '@');
        if ($position_arobase === false)
            echo '<p>Votre email doit comporter un arobase.</p>';
        else {
            $retour = mail('jules@free.fr', 'Envoi depuis la page Contact', $_POST['message'],
'From: ' . $_POST['email']);
            if($retour)
                echo '<p>Votre message a été envoyé.</p>';
            else
                echo '<p>Erreur.</p>';
        }
    }
    ?>
</body>
</html>
```

Envoi d'un email au format HTML

Pour envoyer un email au format HTML, il faut mettre l'entête de l'email au format *MIME (Multipurpose Internet Mail Extensions)*.

Attention : les guillemets doubles autour des "`\r\n`" sont indispensables. Il permettent de générer un caractère de retour à la ligne dans l'entête de l'email.

Ici l'intérêt est de pouvoir mettre des balises HTML dans le message. On pourra par exemple mettre des caractères gras, des liens, des couleurs et autres formatages. Il est également possible de mettre du code CSS, interne ou en ligne, qui sera, la plupart du temps, bien interprété par les clients mails.

```
<!doctype html>
<html>

<head>
  <meta charset="utf-8">
  <title>Contact</title>
</head>

<body>
  <h1>Contact</h1>
  <form method="post">
    <label>Nom</label>
    <input type="text" name="nom" required>
    <label>Email</label>
    <input type="email" name="email" required>
    <label>Message</label>
    <textarea name="message" required></textarea>
    <input type="submit">
  </form>
  <?php
  if(isset($_POST['message'])){
    $entete = 'MIME-Version: 1.0' . "\r\n";
    $entete .= 'Content-type: text/html; charset=utf-8' . "\r\n";
    $entete .= 'From: ' . $_POST['email'] . "\r\n";

    $message = '<h1>Message envoyé depuis la page Contact de monsite.fr</h1>
    <p><b>Nom : </b>' . $_POST['nom'] . '<br>
    <b>Email : </b>' . $_POST['email'] . '<br>
    <b>Message : </b>' . $_POST['message'] . '</p>';

    $retour = mail('destinataire@free.fr', 'Envoi depuis page Contact', $message,
$entete);
    if($retour) {
      echo '<p>Votre message a bien été envoyé.</p>';
    }
  }
  ?>
</body>
</html>
```