X) Envoi d'emails

PHP avancé

Envoi d'emails

L'envoi d'emails en PHP est relativement simple et facile à mettre en oeuvre. C'est la fonction **mail** qui s'occupe de gérer ça.

Syntaxe

mail(destinataire, sujet, message,header)

Note : Il existe un 5ième paramètre optionnel que nous ne verrons pas

destinataire la ou les adresses destinataires
sujet le titre de votre email
message le corps du message à proprement parler
header paramètres optionnels permettant de donner des informations supplémentaires.

Il existe 2 formats d'emails : **texte** et **html**.

PHP avancé

Envoi d'emails

X.1) Email de type "texte"

Les plus faciles à mettre en place mais relativement limités en terme de rendu puisqu'il ne permettent d'afficher que du texte. Les seules mises en forme possibles sont des sauts de ligne (**\r\n**) ou des tabulations (**\t**).

Pour ce type d'email on peut très bien ne pas rajouter d'informations dans le header. (Le minimum est généré automatiquement).

Exemple :

```
$destinataire = 'toto@gmail.com';
$titre = 'Salut ceci est le titre';
$message = "Ici le message avec un saut de ligne\r\n";
$message.= "\tEt ici la suite du message avec une tabulation au début";
mail($destinataire, $titre, $message);
```

PHP avancé

Envoi d'emails

La fonction **mail** revoie une valeur booléenne **true** si l'email a été envoyé avec succès (ce qui ne veut pas forcément dire qu'il est arrivé à bon port), **false** dans le cas contraire. On prendra donc l'habitude de tester cette valeur de retour afin de s'assurer du bon déroulement.

```
// initialisation
$destinataire = 'toto@gmail.com';
$titre = 'Salut ceci est le titre';
$message = "Ici le message avec un saut de ligne\r\n";
$message.= "\tEt ici la suite du message avec une tabulation au début";

// envoi du courrier
$envoi = mail($destinataire, $titre, $message);

// vérification
if ($envoi == true) {
  echo 'courrier correctement envoyé';
} else {
  echo 'problème d'envoi...';
}
```

PHP avancé

Envoi d'emails

X.1.1) Informations complémentaires

Il sera cependant préférable de venir compléter les informations par défaut du header. Nous allons ainsi pouvoir préciser d'où part le courrier, mettre d'autres destinataires en copy caché et bien d'autres choses. Voyons en quelques unes classiques :

Cc: "Carbon Copy", ajoute des adresses en copie
Bcc: "Blind Carbon Copy", pareil mais en copie cachée
From: précise l'adresse de l'expéditeur
Reply-to: spécifie une adresse de réponse

Et d'autres plus spéciales :

X-Confirm-Reading-To: adresse de confirmation de réception
Priority: spécifie la priorité de l'email ("normal", "urgent" ou "non-urgent")

PHP avancé

Envoi d'emails

X.2) Email de type "html"

Il est bien évidemment plus intéressant d'envoyer un courrier dans ce format, les possibilités de mise en page étant beaucoup plus riche. Pour faire comprendre aux gestionnaires de messagerie que notre courrier est dans ce format, nous allons devoir passer des informations supplémentaires au "header".

```
// initialisation
$destinataire = 'toto@gmail.com';
$titre = 'Titre du courrier';
$message = '<strong>Message</strong> du courrier' . "\r\n";
$header = 'MIME-Version: 1.0' . "\r\n";
$header.= 'Content-type: text/html; charset=utf-8' . "\r\n";

// envoi du courrier
mail($destinataire, $titre, $message, $header);
```

Note : Certains logiciels de messagerie ne comprennent pas l'**utf8**, il faudra éventuellement utiliser l'**iso-8859-1**.

PHP avancé

Envoi d'emails

X.3) Email alternatif

Notre courrier va avoir ici 2 composantes : une partie **texte** et une partie **html**. Si l'utilisateur ne peux lire les emails contenant du HTML, c'est la version texte qui sera affichée.

X.3.1) Header

Commençons par préciser l'aspect "alternatif" de notre courrier dans le header grâce à l'attribut **Content-type** :

```
$header = 'From: "Toto" <toto@cheztoto.com>' . "\r\n";
$header = 'Reply-to: "Contact" <contact@cheztoto.com>' . "\r\n";
$header = 'MIME-Version: 1.0' . "\r\n";
$header.= 'Content-type: multipart/alternative;' . "\r\n";
```

PHP avancé

Envoi d'emails

X.3.2) Boundary

Nous allons devoir "séparer" les 2 aspects de notre courrier (texte et html) par ce qu'on appelle une frontière. (**boundary**)
Cette frontière doit être unique afin d'être sûr qu'elle isole bien les différentes partie de notre courrier. Elle est très couramment définie ainsi :

```
$frontiere = md5(rand());
```

Il nous faut absolument l'ajouter dans le **Content-type** du header :

```
$header = 'From: "Toto" <toto@cheztoto.com>' . "\r\n";
$header = 'Reply-to: "Contact" <contact@cheztoto.com>' . "\r\n";
$header = 'MIME-Version: 1.0' . "\r\n";
$header.= 'Content-type: multipart/alternative; boundary="'.$frontiere.'"'."\r\n";
```

PHP avancé
Envoi d'emails

Nous allons également devoir utiliser cette frontière dans le corps du message pour séparer les parties.

Une frontière est précédée de 2 tirets (**--**).
Sa fermeture doit avoir également 2 tirets à la fin.

Le contenu de notre message sera donc désormais structuré ainsi :

- frontière
- définition du type 'texte' / charset / encodage
- message **texte**
- frontière
- définition du type 'html' / charset / encodage
- message **html**
- fin de frontière

Voici page suivante un exemple complet simplifié :

PHP avancé

Envoi d'emails

```
$destinataire = 'cible@gmail.com';
$titre = 'Titre du courrier';
$frontiere = md5(rand());
// HEADER
$header = 'From: "Toto" <toto@cheztoto.com>' . "\r\n";
$header = 'Reply-to: "Contact" <contact@cheztoto.com>' . "\r\n";
$header = 'MIME-Version: 1.0' . "\r\n";
$header.= 'Content-type: multipart/alternative; boundary="'.$frontiere.'"'."\r\n";
// MESSAGE
$message = '--'.$frontiere."\r\n";
// version TEXTE
$message.= 'Content-Type: text/plain; charset="ISO-8859-1"'."\r\n";
$message.= 'Content-Transfer-Encoding: 8bit'."\r\n";
$message.= 'Message au format texte'."\r\n";
$message.= '--'.$frontiere."\r\n";
// version HTML
$message.= 'Content-Type: text/html; charset="ISO-8859-1"'."\r\n";
$message.= 'Content-Transfer-Encoding: 8bit'."\r\n";
$message.= 'Message au format <strong>html</strong>'."\r\n";
$message.= '--'.$frontiere.'--'."\r\n";
// ENVOI du COURRIER
mail($destinataire, $titre, $message, $header);
```

Content-Transfer-Encoding nous sert ici à spécifier un encodage sur 8bits, ce qui assure le bon encodage des caractères accentués.
(Par défaut l'encodage est de type anglosaxon, sur 7bits)
Cet attribu nous sera également utile lorsque nous allons attacher une pièce jointe à notre courrier.

X.4) Email avec pièce jointe

Dans ce cas nous allons avoir également 2 parties distinctes : un message à l'attention du destinataire et un fichier attaché...

Nous retrouvons donc une structure similaire avec des frontières mais nous allons devoir modifier les **Content-type** du header et de la partie du message contenant la pièce jointe ainsi que l'encodage et quelques autres petites choses...

Voyons cela dans le détail :

PHP avancé

Envoi d'emails

X.4.1) Header

Le type de l'email est maintenant de type **multipart/mixed**.

```
$header.= 'Content-type: multipart/mixed; boundary="'.$frontiere.'"'."\r\n";
```

X.4.2) Partie "pièce jointe" du message

L'encodage est désormais en **base64**. Nous devons préciser qu'il s'agit d'une pièce jointe avec un nouvel attribu : **Content-Disposition**. Il nous reste à préciser le type MIME.

```
$message.= 'Content-Type: application/octet-stream; name="nom_fichier"'."\r\n";
$message.= 'Content-Transfer-Encoding: base64'."\r\n";
$message.= 'Content-Disposition: attachment;'."\r\n";
```

Note : Plutôt que de préciser le type exact de fichier que nous envoyons, nous précisons juste que c'est du binaire.

PHP avancé

Envoi d'emails

X.4.3) "Attacher" la pièce jointe

Il nous reste à lire l'intégralité du fichier que nous souhaitons attacher et l'encoder correctement dans notre email. Quelques fonctions vont nous y aider :

```
$fichier = file_get_contents('fichier.ext');
$fichier = chunk_split(base64_encode($fichier));
```

file_get_contents : Envoie le contenu d'un fichier dans une chaîne
base64_encode : Encode une chaîne en type MIME base64
chunk_split : Formate une chaîne selon la norme RFC 2045

La partie du message pour la pièce jointe :

```
$message.= 'Content-Type: application/octet-stream; name="fichier.ext"'."\r\n";
$message.= 'Content-Transfer-Encoding: base64'."\r\n";
$message.= 'Content-Disposition: attachment; filename="fichier.ext"'."\r\n";
$message.= $fichier."\r\n";
```