

# Technologies du Web - TP 4 : Formulaires

Maintenant que vous savez créer et exécuter une page PHP, nous allons passer à l'étape suivante : traiter les entrées fournies par l'utilisateur, envoyer des emails (si l'UM nous y autorise), et utiliser les sessions PHP.

## Exercice 1 - formulaire de base

Vous créerez un formulaire simple dans une page nommée "formulaire.html", qui contiendra tous les éléments demandés au premier point du TD 4 (destinataire, nom, objet, contenu, format). Pour le champ destiné à recevoir le contenu de l'email, on utilisera <textarea>.

Voici pour vous aider une référence des éléments de formulaires (en anglais) :

[http://www.w3schools.com/html/html\\_forms.asp](http://www.w3schools.com/html/html_forms.asp)

## Exercice 2 - traitement de formulaire, envoi d'email

Nous allons maintenant créer la page PHP nommée "traitement.php", qui recevra les données du formulaire et enverra un email.

**AVERTISSEMENT** : normalement la fonction mail() est limitée sur le réseau de l'UM, vous ne devriez pas pouvoir envoyer réellement de message. Mais méfiez-vous, on ne sait jamais :-). Si les messages sont réellement envoyés, **vous êtes responsables de vos envois et du respect de la charte d'utilisation du réseau de l'UM** ! (ne spammez pas)

Dans la balise <form> de votre formulaire, choisissez une méthode d'envoi et indiquez comme action la page PHP nouvellement créée, qui effectuera le traitement (action="").

Écrivez maintenant cette page PHP. Récupérez dans des variables les données envoyées par le formulaire, et passez-les à la fonction mail() : <http://php.net/manual/fr/function.mail.php>  
Exemple : mail(\$destinataire, \$sujet, \$message)

Dans la documentation, le chapitre "valeurs de retour" vous informe de ce que renvoie la fonction mail(). Utilisez la valeur de retour pour afficher un message, qui indique si l'envoi d'email a réussi ou échoué.

Pour plus de clarté et pour tester votre récupération de valeurs, vous pourrez aussi afficher un résumé des informations envoyées par l'utilisateur.

## Exercice 3 - redirection

Nous voulons maintenant vérifier que les données du formulaire sont correctes avant d'envoyer l'email, et si elles sont incorrectes, rediriger l'utilisateur vers le formulaire.

Écrivez le code PHP qui vérifie que le destinataire, l'auteur, le sujet et le message ne sont pas vides; vous utiliserez des tests "if", en utilisant les opérateurs || ou && sur les conditions.

Si au moins un des champs ci-dessus est vide, nous voulons rediriger l'utilisateur vers le formulaire. Vous utiliserez pour cela un entête (header) de redirection. Voici la documentation de la fonction header : <http://php.net/manual/fr/function.header.php>  
Au chapitre "liste de paramètres" de cette page, inspirez-vous du deuxième exemple pour faire votre redirection.

Dans un tel cas, comment faire pour afficher un message d'erreur dans le formulaire, ou mieux : ne pas perdre ce qu'on a saisi ?

## Exercice 4 - session PHP

On veut interdire à un utilisateur d'envoyer plus de 5 emails. Pour cela, on va compter combien d'emails ont été envoyés. Comme on passe d'une page à une autre lorsqu'on valide le formulaire, on ne peut pas utiliser de variable simple pour compter, car cette variable serait réinitialisée à chaque appel de la page.

On va donc utiliser une session PHP : <http://fr2.php.net/manual/fr/function.session-start.php>

Au début de la page de traitement, utilisez session\_start(). Ceci va créer (la première fois) et rappeler (les fois suivantes) un espace sur le serveur, réservé au client (la machine) en cours, dans lequel vous pourrez stocker des valeurs et les conserver de page en page.

Le tableau \$\_SESSION est fait pour ça : vous pouvez y ajouter les "cases" que vous voulez. Exemples : \$\_SESSION["prenom"] = "pierre"; \$\_SESSION["age"] = 45;

La première fois que vous accédez à une case de ce tableau, vous devrez initialiser une valeur; les fois suivantes vous pourrez la modifier (incrémenter un compteur par exemple).

Rappel : pour savoir si une valeur existe à un indice de tableau donné, on utilise isset()

Mettez en place ce compteur. Lors de chaque validation de formulaire, affichez sa valeur; si la valeur dépasse 5, empêchez l'utilisateur d'envoyer l'email et affichez un message d'erreur.

Comment réinitialiser le compteur ?