

Ca marche pas ?

"Ca marche pas !!!" vous connaissez ? Tout ces petits bugs qui nous rendent fous, peuvent être facilement repérés si on utilise une méthode organisée pour les détecter.

Vu les questions qui reviennent en permanence sur le chat et le forum ne me dites pas non :) et comme visiblement un seul tuto sur le déboguage ne suffit pas, voici un tuto complémentaire "ca marche pas".

Les messages d'erreur.

Première chose importante, il faut comprendre le message d'erreur correctement et ne pas tenter une interprétation personnelle de l'erreur. Partez du principe que PHP et Mysql ne se trompent pas ! Pour vous aider, voici une liste des erreurs les plus fréquemment rencontrées. Bien sûr, cette liste ne contient pas TOUS les messages d'erreur :).

Message	Signification
Parse error: parse error in xxxx.php on line y	Typique d'une erreur de syntaxe. Vérifiez si vous n'avez pas oublié un ; (point virgule) en fin de ligne. Ou encore un \$ (dollar) devant le nom d'une variable. N'hésitez pas à contrôler les lignes avoisinantes. L'erreur se trouve souvent juste au-dessus.
Warning: Oops, php_SetCookie called after header has been sent in xxxx.php on line y	Vous avez tenté d'initialiser un cookie après que l'entête HTTP soit envoyé au client. Vérifiez si une sortie (echo, print, message d'erreur, ligne blanche avant les tags php) ne se fait quelques temps avant.
Warning: MySQL Connection Failed: Access denied for user:	Erreur de connexion à la base MySQL. Vérifiez si host, user et password sont corrects.
Warning: Unable to create [chemin] No such file or directory in your script on line [numero]	Le chemin vers le répertoire censé contenir le fichier ou bien le chemin du répertoire dans lequel le fichier doit être créé n'est pas bon (n'existe pas)
Warning: 0 is not a MySQL result index in xxxx.php on line y	Erreur probable au niveau de la requête SQL. Vérifiez votre requête SQL, en particulier les champs manipulés, le nom de ou des tables impliquées, etc. Un petit truc : il est souvent pratique de stocker ses requêtes dans une chaîne.
Warning: Variable \$zzzz is not an array or string in xxxx.php on line y	Vous tentez de manipuler une valeur numérique avec une fonction dédiée aux chaînes ou aux tableaux.
Warning: Variable \$zzzz is not an array or object in xxxx.php on line y	Vous tentez de manipuler une valeur numérique avec une fonction dédiée aux tableaux ou aux objets.
Warning: Cannot add header information – headers already sent by (output started at ****.php:21) in ***.php on line 54	Vous avez tenté d'effectuer un header après que l'entête HTTP ait été envoyé au client. Le message est clair : Dans notre exemple votre header est en ligne 54 et l'envoi a commencé ligne 21. Vérifiez si une sortie (echo, print, message d'erreur) ne se fait dans les lignes 21 et suivantes.
Fatal error: Maximum execution time exceeded in xxxx.php on line y	PHP dispose d'un mécanisme permettant de se prémunir des scripts susceptibles d'engendrer un temps d'exécution trop important pouvant saturer un serveur.

	Par défaut, ce temps est de 30 secondes.
Fatal error: Allowed memory size of 8388608 bytes exhausted (tried to allocate x bytes) in yyyy.php on line z	PHP dispose d'un mécanisme permettant de se prémunir des scripts susceptibles d'engendrer une consommation mémoire trop importante pouvant saturer un serveur. Par défaut, une limite est fixée à environ 8 Mo (8388608 octets).
Fatal Error: Call to undefined function: xxxx() in yyy.php on line z	La fonction que vous appelez n'existe pas. Ce peut-être une fonction liée à une librairie externe (GD, Zlib, PDF, etc.). Dans ce cas, un simple <code>phpinfo()</code> vous renseignera sur les paramètres de compilation de votre version de PHP. Peut-être s'agit-il sinon d'une de vos propres fonctions. Vérifiez alors qu'elle existe. Et dans tous les cas, contrôlez de plus près le nom de la fonction appelée (orthographe, etc.). Une erreur de frappe est vite arrivée.
Fatal Error: Cannot redeclare xxxx() in yyy.php on line z	Vous avez certainement déclaré plusieurs fois la même fonction. Contrôlez à nouveau l'ensemble des fonctions que vous avez créées. Et n'hésitez pas à vérifier également dans les éventuels fichiers inclus. C'est souvent dans un script secondaire que vous trouverez le doublon. Veillez aussi à ne pas utiliser le nom d'une fonction propre à PHP ou à l'une de ses librairies.
Fatal error: Input in flex scanner failed in xxxx on line y	Vérifiez vos <code>include</code> et <code>require</code> . Il y a fort à croire que vous avez indiqué un chemin incomplet (genre <code>/usr/local/</code> sans préciser de fichier).
Fatal error: Input in flex scanner failed in xxxx on line y	Vérifiez vos <code>include</code> et <code>require</code> . Il y a fort à croire que vous avez indiqué un chemin incomplet (genre <code>/usr/local/</code> sans préciser de fichier).
Failed opening '%s' for inclusion (include_path='%'s')	Le fichier n'a pas pu être inclus dans votre script, car PHP n'a pas pu y accéder : vérifiez les droits (utilisateur PHP, droits du fichier), les noms et chemins du fichier inclus.
file("%s") – Bad file descriptor	Problème d'accès à un fichier avec la fonction <code>file()</code> . Vérifiez bien que l'URL est valide. (l'URL " <code>http://www.super.php</code> ") est invalide alors qu'une erreur de type 404 sera valide.
Wrong parameter count for %s()	La fonction est appelée avec un nombre insuffisant de paramètres, ou bien avec trop de paramètres. Certaines fonctions ont besoin d'un minimum de paramètres (<code>array()</code>), et généralement d'un maximum.
stat failed for %s (errno=%d – %s)	Impossible d'accéder au fichier (problème de droits ou de chemin d'accès).

A noter que si vous avez une parse error à la ligne 30, l'erreur peut se trouver à la ligne de code précédente (par exemple il peut manquer un ; à la fin de la ligne)

Vous pouvez aussi obtenir une parse error "bizarre" en fin de page. Vérifiez bien si vous avez fermé toutes vos accolades. Vous devez avoir autant d'accolades ouvrantes { que d'accolades fermantes }. Sinon PHP patiente jusqu'à la fin du script et signale l'erreur en dernière ligne.

Problèmes de variables

Parfois, vous pouvez avoir certains problèmes à cause de variables extérieures au script, elles ne semblent pas passer. D'abord assurez vous de ce qui passe vraiment, en imprimant vos variables en début de script, vous verrez ainsi d'un coup d'oeil, si certaines (ou toutes les) variables sont vides. Vous gagnerez au moins 30 minutes, plutôt que de chercher ce qui ne va pas et qui devrait marcher :). N'hésitez pas non plus à faire un echo ou un print chaque fois que vous avez manipulé une variable. Vous l'enlèverez après. Il vaut mieux régler les problèmes au fur et à mesure !

Rappelez vous notre chapitre sur les variables globales (register_globals) et vérifiez bien que vous récupérez les variables par les bons tableaux (d'après la version de PHP sur le serveur).

Vous pouvez utiliser var_dump() ou print_r() pour afficher d'un coup le contenu des tableaux.

Problèmes avec mysql

Ca ne sert à rien de tourner en rond et de jurer qu'une DB ou une table existe, si PHP vous dit le contraire. Avec Mysql il y a quelques bonnes habitudes à prendre.

die()

Cette fonction permet d'afficher un message et d'arrêter le script. Vous devriez toujours l'utiliser.

```
mysql_connect('localhost','root','');
or die('Problème de connection à mysql');
```

En cas d'erreur, le script s'arrêtera et affichera 'Problème de connection à mysql', de cette façon vous saurez exactement où se trouve le problème dans le script.

```
mysql_connect('localhost','root','');
or die('Problème de connection à mysql');

$link = mysql_select_db('tes')
or die('Problème de DB');
```

Construire votre requête dans une variable.

Prenez l'habitude de construire votre requête dans une variable à part, afin de pouvoir afficher votre requête et débogger facilement. En effet votre requête sera affichée avec le contenu de vos variables, ce qui vous permettra de repérer par exemple une variable qui n'est pas passée, une variable avec un ' non échappé etc ...

```
$select = "select nom, prenom from aches where race='".$race' ";
$result = mysql_query($select,$link)
or die('requete =>'.$select.'<br>');
```

Utiliser mysql_error()

Cette fonction retournera une erreur mysql sous la forme d'un texte, vous pouvez l'utiliser avec la fonction die()

```
$select = "select nom, prenom from aches where race='".$race' ";
$result = mysql_query($select,$link)
or die('requete =>'.$select.'<br>
error->'.mysql_error());
```

Résultat

```
requete =>"select nom, prenom from aches where race='sallers' ";
error->Table 'test.aches' doesn't exist // Bon sang mais c'est bien sûr !
```

Eviter l'affichage si pas de résultat

Si vous tentez d'afficher le résultat d'un select, alors que celui ci ne retourne pas de résultat, vous serez confronté à un message d'erreur. Il vaut mieux, donc, prendre l'habitude de vérifier si il y a un résultat à afficher. Pour cela vous pouvez utiliser la fonction mysql_num_rows()

```
$select = "select nom, prenom from vaches where race='\$race' ";
$result = mysql_query($select,$link)
or die('requete =>'.$select.'<br> error->'.mysql_error());

$compte = mysql_num_rows($result);

if($compte>0) {
    while($row = mysql_fetch_row($result)) {
        echo $row[0]. '<br>';
    }
}
else {
    echo 'aucun résultat';
}
```

Instruction conditionnelles

If...else

Votre script n'a pas le comportement attendu, il passe dans un if alors qu'il ne devrait pas, ou il passe dans un else alors que vous vous attendiez à ce qu'il passe dans le if.

Ne tournez pas autour du pot ! Affichez les variables utilisées dans le test avant le if afin d'être certain qu'elles ont les valeurs attendues.

Etes-vous sûr d'avoir utilisé l'opérateur de comparaison (==) et non l'opérateur d'assignation (=) ?

Les espaces dans les chaînes peuvent être une source de perte de temps également. Imaginez que vous comparez 2 variables de type string dans votre if, mais que l'une d'elle, pour l'une ou l'autre raison, ait un espace en début ou en fin de chaîne. Ce qui la rend non égale à l'autre chaîne.

Afin de vérifier ce genre de petites erreurs bêtes, faites un écho de vos 2 variables et entourez-les avec des *, de cette façon, du premier coup d'oeil, vous verrez si un espace s'est glissé dans votre chaîne.

```
$var1 = 'lala';
$var2 = 'lala ';

echo '*' . $var1 . '*' . <br>;
echo '*' . $var2 . '*' . <br>

if($var1 == $var2) {
    echo 'identiques';
}
else {
    echo 'pas identiques';
}
```

Les boucles

Une boucle n'est jamais exécutée ? Vérifiez la valeur de départ de la variable censée déclencher la boucle.

Mon script devient fou et ne s'arrête pas ! Vous avez sans doute un problème avec la condition d'arrêt de la boucle (elle n'est jamais atteinte). Dans le cas d'une boucle while, êtes vous certain de ne pas avoir oublié l'itération de votre variable ?

Pour voir le nombre de passages dans la boucle (afin de contrôler un peu) vous pouvez ajouter un echo '.'; cela vous permettra de voir combien de fois la boucle est exécutée et voir si c'est normal. Rien ne vous empêche d'utiliser une variable \$debug servant "d'interrupteur" au debug. Exemple :

```
$i = 0;
$debug = 1;
while($i < 100) {
    if($debug) {
        echo '.';
    }
    $i = $i + 1;
}
```

Si vous passez \$debug à 0, les echo de debug ne seront pas affichés, si vous avez besoin d'afficher des variables afin d'en contrôler la valeur, vous n'avez plus qu'à réactiver votre interrupteur.

Les fichiers

Une erreur d'include, de require ou de fopen peut avoir plusieurs causes.

- Le chemin vers le fichier n'est pas correct.
- Les permissions sur le fichier ne sont pas suffisantes.
- Certaines fonctions d'information sur les fichiers ne fonctionnent pas sous windows (ce qui est en général est indiqué dans la doc).

La fonction mail()

Vous devez bien vous douter que les mails ne s'envoient pas par vaches volantes (non ? même pas un petit peu ?).

Pour envoyer un mail, vous avez besoin ou bien :

- d'avoir un serveur SMTP sur le serveur qui héberge votre script
- ou bien de configurer votre php.ini avec un serveur SMTP que vous pouvez utiliser (celui de votre FAI par exemple)
- Et surtout d'avoir configuré le php.ini en conséquence

```
[mail function]
; pour windows
SMTP = smtp.monfai.fr

; pour windows
sendmail_from = test@moi.fr

; pour unix.
;sendmail_path =
```

Si vous obtenez cette erreur en utilisant la fonction mail(), rappelez vous de mon histoire de vaches volantes et vérifiez la configuration de PHP. N'oubliez pas de redémarrer Apache pour que les infos soient prises en compte (merci oreil)

```
Warning: Failed to Connect in [chemin] on line [nombre]
```

Pas à pas

Rien ne fonctionne comme vous voulez ? vous devenez fou à essayer de corriger votre script ? :) Restez Zen ! Debogez

votre script partie par partie, prenez le temps de tracker vos variables en les affichant avant et après certains traitements.

Si ça ne va toujours pas, allez prendre l'air, ou reprenez votre travail le lendemain, vous verrez qu'une bonne nuit de sommeil peut faire des miracles :)

Conclusion

Avec un peu d'expérience et de patience, vous allez développer des automatismes de déboggage. Et bientôt ces parses error qui vous faisaient perdre un temps fou n'auront plus de secret pour vous.

Oh capitaine, mon capitaine... [flyingcow](#)

www.phpdebutant.org © 2006 – L'équipe de phpDebutant