

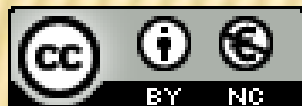
LA GESTION DES ERREURS EN PHP



LES ERREURS EN PHP



- ✗ Auteur → Ramus Ledorf
- ✗ Contributeur → Wassim, Romain, Jocelyn
- ✗ Scénariste → Wassim, Romain, Jocelyn
- ✗ Pilotage → Wayz, Weylin, Fury (WWF)
- ✗ Licence → Creative Commons



CITATION

**L'expérience, c'est le nom que chacun
donne à ses erreurs.**

Oscar Wilde

DÉFINITION : BUG, ERREUR, DEBUG

Bug ➔ un bogue est un dysfonctionnement d'un programme informatique, au moment du bogue l'état du programme est soit le résultat faux, soit l'arrêt, soit la boucle infinie

L'erreur ➔ plusieurs types d'erreur (compilation, exécution, logique, ...) peuvent être la cause du dysfonctionnement d'un programme

Debug ➔ activité importante des concepteurs (et des testeurs) qui consiste en la recherche et la correction des erreurs

TYPES D'ERREURS

✗ Erreurs de compilation

- Erreur de syntaxe
- Erreur d'orthographe

✗ Erreurs d'exécution

- Opérations impossible
- Boucles infinies

✗ Erreurs de logique

- Résultat faux
- Résultat inattendu

```
if $score > 0 :  
  
echo strtoupret('lol');  
  
$result = $score / 0;
```

L'AFFICHAGE DES ERREURS PAR PHP

(!) Warning: Division by zero in C:\wamp64\www\Cour_Erreurs_PHP\test.php on line 4				
Call Stack				
#	Time	Memory	Function	Location
1	0.0000	235160	{main} ()	... \test.php:0

(!) Warning: Division by zero in C:\wamp64\www\Cour_Erreurs_PHP\test.php on line 4				
1	2	3	4	

1 - Degré d'importance 2 – Opération concernée 3 – Le programme 4 – La ligne

(!) Notice: Undefined variable: fsfg in C:\wamp64\www\Cour_Erreurs_PHP\index.php on line 107				
1	2	3	4	

RÉGLAGES DE L’AFFICHAGE D'ERREUR

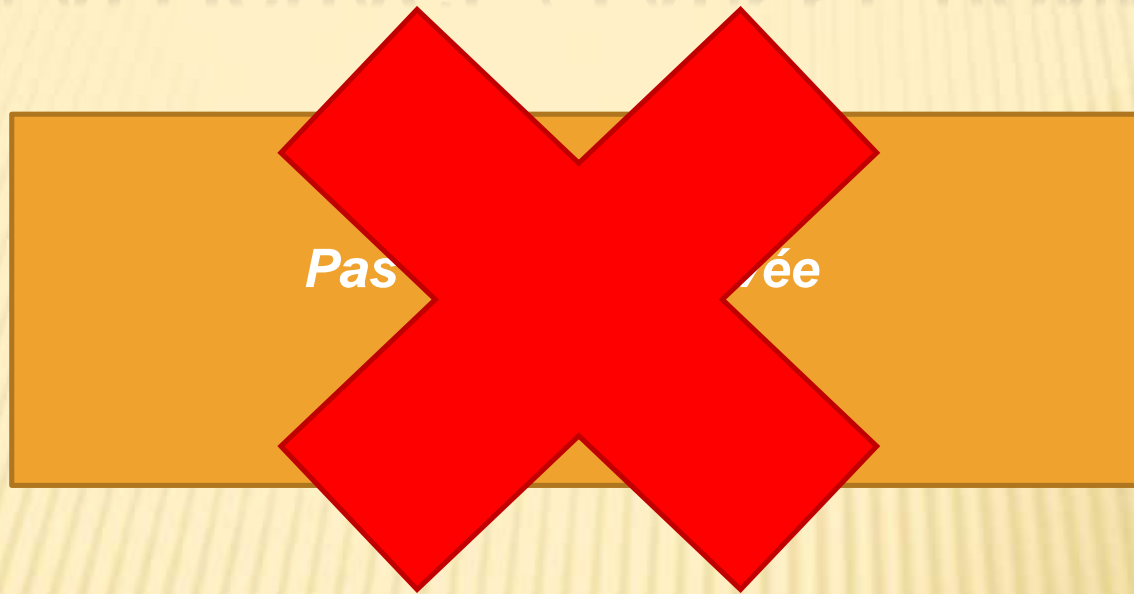
✖ Dans le fichier « **php.ini** »

- + Des variables de configuration
- + Exemples : `display_errors`, `html_errors`, `error_log`, ...

✖ La fonction **error_reporting()**

- + `// Rapporte les erreurs passées en paramètre`
- + `error_reporting(E_ERROR | E_WARNING | E_PARSE | E_NOTICE);`
- + `// Désactive l’affichage d’erreur`
- + `error_reporting(0);`

LE NON AFFICHAGE (FAUSSE BONNE IDÉE)



Il est parfois utile d'empêcher une fonction de déclencher une erreur si son code de retour est correctement géré par ailleurs. L'alias « @ » devant une fonction n'affichera pas l'erreur lorsqu'elle se produira.

```
$var = @strtolower($novar);
```


LES LOGS

Qu'est ce qu'un log ?

Un journal d'événement qui trace tout les comportements anormaux ainsi que les connexions de l'application

Comment l'appeler ?

Directement dans votre code PHP via la fonction `Error_log()` ou dans le fichier de log du serveur

PERSONNALISATION D’AFFICHAGE

Try {} Catch {} un atout dans la personnalisation !!

Le Try Catch sert à tester un morceau de code et à retourner un code d'erreur personnalisé si il y a un problème ...

Les headers (entête HTTP) une arme universelle ... !

L'erreur HTTP 404 est un header qui permet de personnaliser le message d'affichage de votre erreur pour l'utilisateur

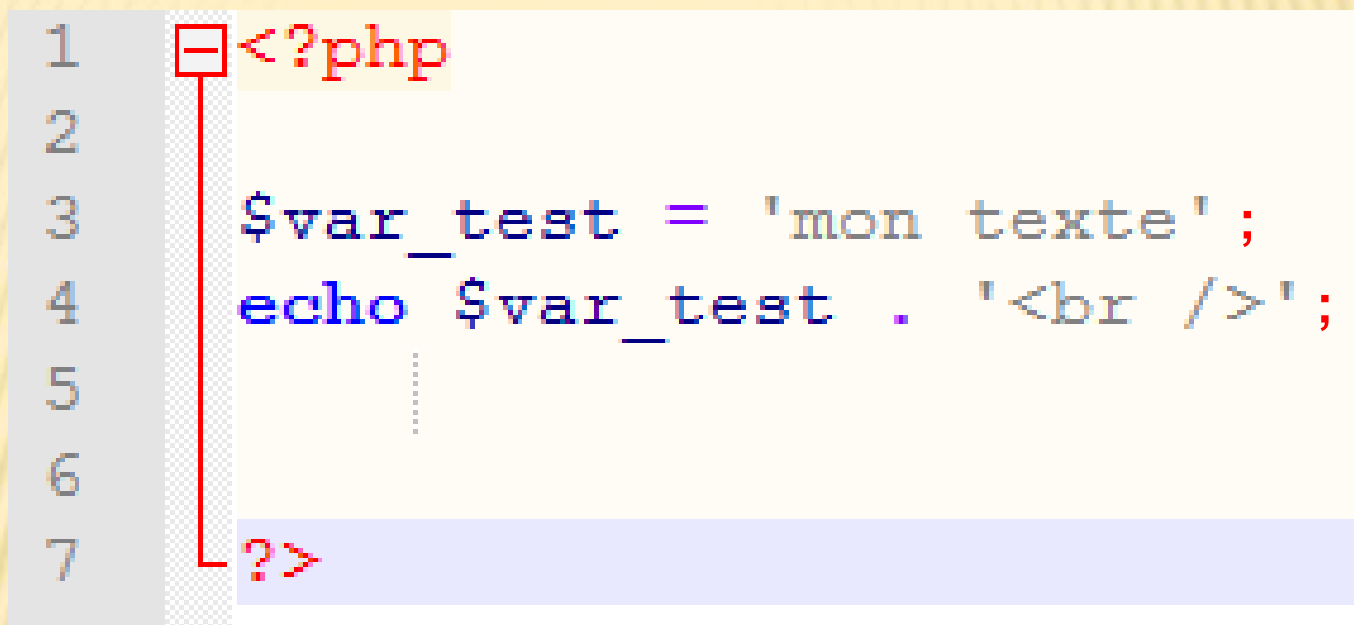
LE DÉBOGAGE

- Prendre le temps de lire le message d'erreur
- Le debug s'effectue pas à pas, la documentation technique à porter de main
- Beaucoup de patience et de sang froid, 1/3 de développement pour 2/3 de débogage
- Le `die();` ou `exit();` sont vos meilleures armes
- Utiliser `var_dump();` affiche le contenu des données
- Les commentaires `//` ou `/* */` pour le reste du code

ERREUR COURANTE

- Erreur ligne 52 vérifier si la ligne 51 manque surement un ;
- Les variables non initialisées !!!
- Les boucles sans fin
- Les erreurs dans les paramètres de connexion
- Utiliser un echo pour afficher un tableau
- Etre certain de ce que l'on a fait !!!
- Faire confiance à son éditeur !

L'ERREUR LA PLUS COURANTE



```
1  <?php
2
3  $var_test = 'mon texte';
4  echo $var_test . '<br />';
5  ...
6
7  ?>
```

L'erreur la plus courante c'est l'oubli
Alors penser bien a fermer votre code !

QUIZ ERREUR PHP



- ✗ Doit-on afficher les erreurs en PHP (Oui/Non) ?
- ✗ Que faut-il faire pour rendre « muette » une erreur ?
- ✗ L'erreur 301 de redirection, a t-elle une incidence sur le PHP ? (Oui/Non)
- ✗ Remettez dans l'ordre le contenu d'affichage de l'erreur en PHP (le fichier, l'opération concernée, le degré d'importance, la ligne concernée)
- ✗ Quelle sont les tâches principales du débogage : (Chercher les erreurs, Éliminer les erreurs, Sauvegarder les erreurs, Résoudre les erreurs)

EXERCICES



- ✗ Faire une division par 0 puis désactiver l'affichage d'erreur.
- ✗ Inclure un fichier inexistant et désactiver l'affichage (sur la même ligne)
- ✗ Intercepter une exception à l'aide de l'instruction try catch
- ✗ Inscrire une erreur dans le log et la retrouver
- ✗ Faire une fonction de personnalisation de l'affichage
- ✗ Déboguer the Orange Screen Of the Death (OSOD)

(!) Notice: Undefined variable: fsfg in C:\wamp64\www\Cour_Erreurs_PHP\index.php on line 107

CORRIGE

```
/*---- Exercice 1 ----*/

error_reporting(0);

$nb = 8 ;
$user_age = 0;

$result = $nb / $user_age;

/*---- Exercice 2 ----*/

@require('fichier_inexistant.php');

/*---- Exercice 3 ----*/

try{
    $bdd = new PDO('tutu', 'tata', 'toto');
} catch(Exception $e){
    echo 'Erreur : '.$e->getMessage().' | Ligne : '.$e->getLine();
} finally{
    echo 'Bonjour à vous!';
}

/*---- Exercice 4 ----*/

$fausseVar = "1656";

// On test si la variable est de type entier
if (!is_int($fausseVar)){
    error_log('Mauvais type de variable!');
}
```


CORRIGE

```
/*---- Exercice 5 ----*/

function my_error_handler($no, $str, $file, $line){
    switch($no){
        // Erreur fatale
        case E_USER_ERROR:
            echo '<p><strong>Erreur fatale</strong> : '.$str.'</p>';
            exit; // On arrête le script
            break;

        // Avertissement
        case E_USER_WARNING:
            echo '<p><strong>Avertissement</strong> : '.$str.'</p>';
            break;

        // Note
        case E_USER_NOTICE:
            echo '<p><strong>Note</strong> : '.$str.'</p>';
            break;

        // Erreur générée par PHP
        default:
            echo '<p><strong>Erreur inconnue</strong> ['. $no .'] : '.$str.'<br/>';
            echo 'Dans le fichier : "'.$file.'", à la ligne '.$line.'</p>';
            break;
    }
}

set_error_handler('my_error_handler');

echo [8,7];
```