



# Inclusion de fichiers

Les inclusions sont très pratiques pour :

- ➡ Ne pas dupliquer de code
- ➡ Réutiliser des fonctions PHP déjà écrites dans d'autres fichiers
- ➡ Par convention, les noms des fichiers à inclure ont un suffixe **.inc**

Exemple: **header.inc.php** ou **header.inc.html**



# Inclusion de fichiers: include

➡ `include "header.inc.php";`

Inclus le fichier `header.inc.php`.

Si le fichier n'existe pas, il y a génération d'un warning.

➡ `include_once "header.inc.php";`

Comme `include`, à la différence que si le fichier est déjà inclus, il ne l'ajoutera pas de nouveau.



# Inclusion de fichiers: require

➡ `require "header.inc.php";`

Inclus le fichier `header.inc.php`.

Si le fichier n'existe pas, il y a génération d'une erreur fatale.

➡ `require_once "header.inc.php";`

Comme `require`, à la différence que si le fichier est déjà inclus, il ne l'ajoutera pas de nouveau.



# Inclusion de fichiers

## Configuration de PHP relative aux erreurs

L'affichage des erreurs est paramétrable en PHP.

En phase de développement il est souhaitable d'afficher toutes les erreurs.

2 possibilités:

- Dans le fichier php.ini
  - Régler les paramètres `display_error` et `error_reporting`  
`display_errors=on`  
`error_reporting=E_ALL`
- Dans le code en début de section PHP  
`error_reporting(E_ALL);`  
`ini_set("display_errors", 1);`



# Remarques sur les erreurs, exceptions, warnings

Php permet d'intercepter les exceptions avec un mécanisme de try /catch semblable à celui d'autres langages.

Ex:

```
try{
    fonction_a_risque();
} catch (Exception $e) {
    echo 'Exception : ', $e >getMessage();
}
```

Les défauts d'inclusion de fichier générés par include et require sont des warnings ou des erreurs fatales qui ne sont pas des exceptions et ne sont pas gérées par ce mécanisme



# Remarques sur les erreurs, exceptions, warnings

## Gestion des Warnings:

```
<?php
//fonction permettant de générer une Exception en cas de Warning
function exception_error_handler($severity, $message, $file, $line) {
    if (!(error_reporting() & $severity)) {
        return;
    }
    throw new RuntimeException($message, 0, $severity, $file, $line);
}

//initialization du error_handler avec la fonction "exception_error_handler"
set_error_handler("exception_error_handler");

//inclusion de fichier avec try/catch
try{
    include "file_include.inc.php";
}catch(RuntimeException $e){
    //traitement du warning
}

//restauration du error_handler
restore_error_handler();
"""
```



# Remarques sur les erreurs, exceptions, warnings

## Gestion des Errors:

```
<?php
// fonction executée en cas de shutdown après une erreur
function shutdown()
{
    if (($error = error_get_last())) {
        //nettoyage du tampon de sortie
        ob_clean();
        header('http/1.1 303 see other');
        header("Location:gestion_fatal_error.php");
    }
}
//demarrage du tampon de sortie
ob_start();
//initialisation de la fonction à exécuter en cas de shutdown après une erreur
register_shutdown_function("shutdown");

//début page WEB
require_once "file_require.inc.php";
'''
```



# Compléments

Lorsque les fichiers sont inclus (require ou include) ils sont immédiatement analysés par le moteur PHP

Si ces fichiers ne contiennent pas de code PHP mais uniquement du html, cette analyse devient inutile, il est possible alors d'utiliser à la place de include/require:

Ex:

```
<?php
```

```
    $data = file_get_contents("fichierhtml.inc.html");  
    echo $data;
```

```
?>
```