

Error Handling

H S Rana

CIT,UPES

April 14, 2014

Error handling is the process of catching errors raised by your program and then taking appropriate action. If you would handle errors properly then it may lead to many unforeseen consequences.

There are different methods to handle error

- Using die() function
- Custom errors and error triggers
- Error reporting

Let us consider following example

```
<?php  
$file=fopen("welcome.txt","r");  
?>
```

Let us consider following example

```
<?php  
$file=fopen("welcome.txt","r");  
?>
```

If the file does not exist you might get an error like this:

Warning: fopen(welcome.txt) [function.fopen]: failed to open stream: No such file or directory in C:\webfolder\test.php on line 2

die function

To prevent the user from getting an error message like the one above, we test whether the file exist before we try to access it:

```
<?php
if(!file_exists("welcome.txt"))
{
    die("File not found");
}
else
{
    $file=fopen("welcome.txt","r");
}
?>
```

Creating a Custom Error Handler

simply create a special function that can be called when an error occurs in PHP.

Creating a Custom Error Handler

simply create a special function that can be called when an error occurs in PHP.

Syntax of error function

```
error_function_name(error_level,error_message,  
error_file,error_line,error_context)
```

Creating a Custom Error Handler

simply create a special function that can be called when an error occurs in PHP.

Syntax of error function

*error_function_name(error_level,error_message,
error_file,error_line,error_context)*

error_level Required. Specifies the error report level for the user-defined error. Must be a value number.

error_message Required. Specifies the error message for the user-defined error.

error_file Optional. Specifies the filename in which the error occurred

error_line Optional. Specifies the line number in which the error occurred

error_context Optional. Specifies an array containing every variable, and their values, in use when the error occurred

Error Report level

value	Error level
4	E_PARSE
8	E_NOTICE
2	E_WARNING
1	E_ERROR
1024	E_USER_NOTICE
512	E_USER_WARNING
256	E_USER_ERROR
2048	E_STRICT
8191	E_ALL

Error Handler Function

Let us create error handler function

```
<?php
//error handler function
function customError($errno, $errstr)
{
    echo "<b>Error:</b> [$errno] $errstr";
}

//set error handler
set_error_handler("customError");

//trigger error
echo $test;
?>
```

Error Handler Function

Let us create error handler function

```
<?php
//error handler function
function customError($errno, $errstr)
{
    echo "<b>Error:</b> [$errno] $errstr";
}

//set error handler
set_error_handler("customError");

//trigger error
echo $test;
?>
```

Error: [8] Undefined variable: test

Trigger an Error

In a script where users can input data it is useful to trigger errors when an illegal input occurs. In PHP, this is done by the `trigger_error()` function.

Trigger an Error

In a script where users can input data it is useful to trigger errors when an illegal input occurs. In PHP, this is done by the `trigger_error()` function.

```
<?php
$test=2;
if ($test>1)
{
trigger_error("Value must be 1 or below");
}
?>
```

Trigger an Error

In a script where users can input data it is useful to trigger errors when an illegal input occurs. In PHP, this is done by the `trigger_error()` function.

```
<?php
$test=2;
if ($test>1)
{
trigger_error("Value must be 1 or below");
}
?>
```

Notice: Value must be 1 or below in C:\webfolder\test.php on line 6

Example

Let us consider following example

```
<?php
//error handler function
function customError($errno, $errstr)
{ echo "<b>Error:</b> [$errno] $errstr<br>";
  echo "Ending Script";
  die();}
//set error handler
set_error_handler("customError",E_USER_WARNING);
//trigger error
$test=2;
if ($test>1){
    trigger_error("Value must be 1 or below",E_USER_WARNING);}
?>
```

Error: [512] Value must be 1 or below
Ending Script

Home Assignment

www.w3school.com