

Exception Handling

H S Rana

CIT,UPES

April 16, 2014

Exception

- Exceptions are used to change the normal flow of a script if a specified error occurs.
- PHP 5 came a new object oriented way of dealing with errors.
- Exceptions are important and provides a better control over error handling.
- Exception handling is used to change the normal flow of the code execution if a specified error (exceptional) condition occurs. This condition is called an exception.

- When an exception is thrown, the code following it will not be executed, and PHP will try to find the matching "catch" block.
- If an exception is not caught, a fatal error will be issued with an "Uncaught Exception" message.
- To handle exception we use try and catch block

try a function call of a function that throw exception should be in try block.If the exception does not trigger, the code will continue as normal. However if the exception triggers, an exception is "thrown".

throw by using throw program can throw exception

catch A "catch" block retrieves an exception and creates an object containing the exception information.

Example

Let us consider example

```
<?php
//create function with an exception
function checkNum($number){
    if($number>1){
        throw new Exception("Value must be 1 or below");}
    return true;}
//trigger exception in a "try" block
try{
    checkNum(2);
    //If the exception is thrown, this text will not be shown
    echo 'If you see this, the number is 1 or below';}
//catch exception
catch(Exception $e)
{
    echo 'Message: ' . $e->getMessage();
}
```

Creating a Custom Exception Class

You can create custom exception class by extending php Exception class and you can add your own custom function to it for example:

Creating a Custom Exception Class

You can create custom exception class by extending php Exception class and you can add your own custom function to it for example:

```
<?php
class customException extends Exception
{
    public function errorMessage()
    {
        //error message
        $errorMsg = 'Error on line '.$this->getLine().' in '
.$this->getFile()
.': <b>'.$this->getMessage().'</b> is not a positive number';
        return $errorMsg;
    }
}
```

Custum Exception example

Let us consider the following example

```
<?php
$e = -5;
try {
    if($e<0){
        throw new customException($e);
    }
}
catch (customException $e)
{
    //display custom message
    echo $e->errorMessage();
}
?>
```


Multiple Exceptions

It is possible for a script to use multiple exceptions to check for multiple conditions.

```
<?php
class customException extends Exception
{
    public function errorMessage()
    {
        //error message
        $errorMsg = 'Error on line '.$this->getLine().' in '
.$this->getFile()
.': <b>'.$this->getMessage().'</b> ';
        return $errorMsg;
    }
}
```

?>

Multiple Exception Example

Let us consider following example

```
<?php
$e = -5;$a=0;
try{
    if($e<0){
        throw new customException("number is not positive"); }
    if($a==0){
        throw new Exception("Divisible by zero Exception");}
    echo $e/$a; }
catch (customException $e){
    //display custom message
    echo $e->errorMessage(); }
catch (Exception $e)
{ echo $e->errorMessage();}
?>
```

- Re-throwing Exception
- Set a Top Level Exception Handler

www.w3school.com