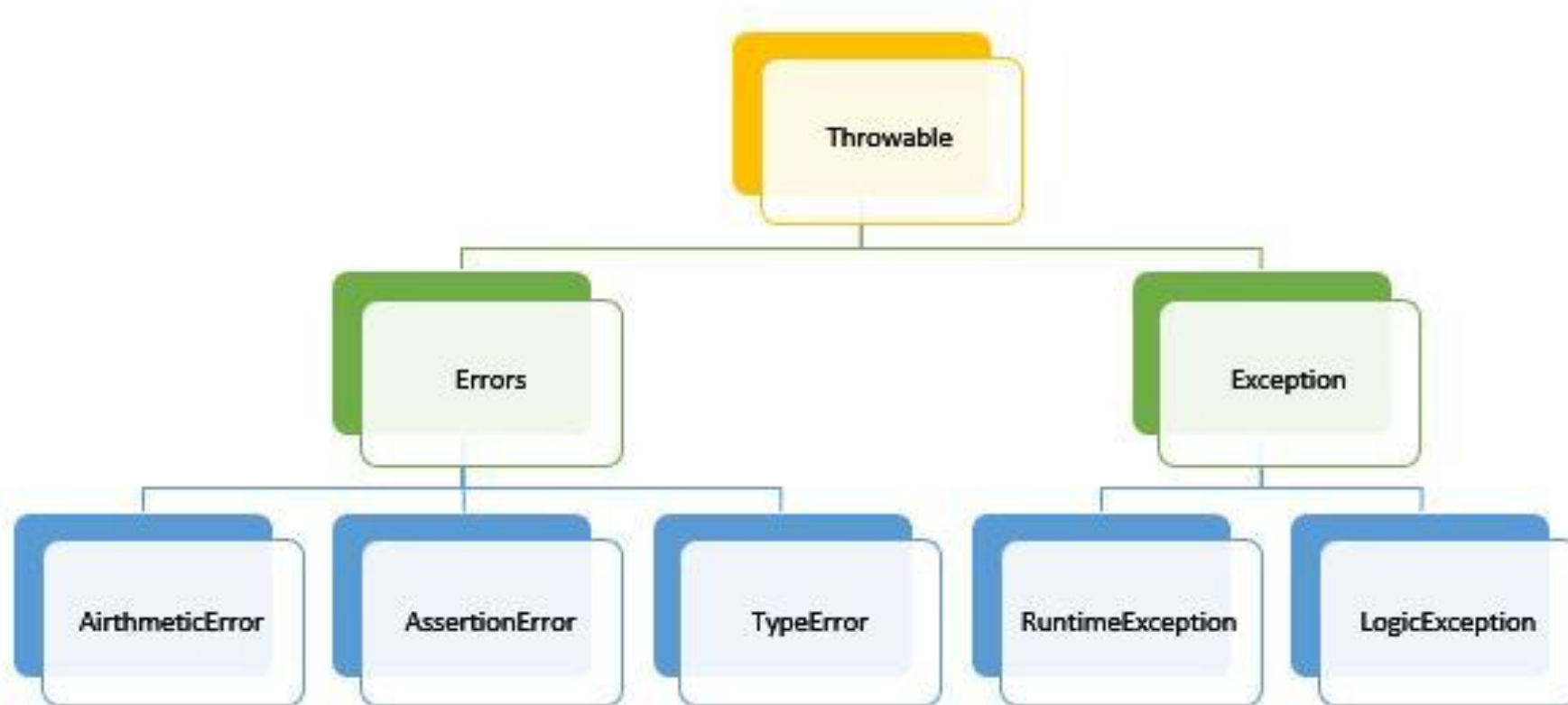


PHP:Errors i Excepcions

Throwable es la interfaz base para cualquier objeto que pueda ser lanzado mediante una sentencia throw en PHP 7, incluyendo Error y Exception.



	Errors	Excepcions
1.	Totes les excepcions són errors.	La inversa no és certa.
2	No es poden solucionar tots els errors.	Es poden gestionar totes les excepcions.
3.	No es poden controlar tots els errors en temps d'execució.	Les excepcions es poden gestionar en temps d'execució.
4.	Els errors del programa solen assenyalar el final de l'execució del programa, normalment no es poden recuperar del seu estat d'error i haurien de provocar la sortida del programa actual.	Les excepcions del programa no assenyalen el final de l'execució del programa i es podrien gestionar o recuperar de l'estat d'excepció en lloc de sortir del programa actual.
5.	Els desenvolupadors normalment no poden endevinar errors.	Els desenvolupadors poden encertar les excepcions fàcilment i es poden agafar mitjançant el bloc de captura try.

Exception and Error handling in PHP 7

There are three important keywords in this exception model as described below.

Try – The block of code or expressions which are anticipated to throw an error or exception should be enclosed inside the “try” block. This block will execute as a normal code block if no exception is thrown. If an exception is triggered inside the “try” block then it will throw an exception and pass the control to the exception or error matching catch block.

Throw – It triggers an exception. Each “throw” must have at least one “catch”.

Catch – The catch block has an exception or error class that matches with the type of exception that is thrown in the try block and invokes an object that contains the required exception information.

Try

Bloc de codi on hi ha la previsió de generació d'error i on desitgem controlar aquesta possible situació.

Throw

Disparador que ha de executar-se si al bloc anterior és produïx una excepció o un error

Catch

Bloc que captura l'excepció i/o error. En aquest bloc és on es dona solució a la situació generada per l'error

Examples: try -catch

```
01 <?php
02 try {
03     // init bootstrapping phase
04
05     $config_file_path = "config.php";
06
07     if (!file_exists($config_file_path))
08     {
09         throw new Exception("Configuration file not found.");
10     }
11
12     // continue execution of the bootstrapping phase
13 } catch (Exception $e) {
14     echo $e->getMessage();
15     die();
16 }
17 ?>
```

Exemples: Funció +Try-Cath

```
<?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();
}
?>
```

Exemples: Classe d'exceptions

```
<?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 valid E-Mail address';
        return $errorMsg;
    }
}

$email = "someone@example...com";

try {
    //check if
    if(filter_var($email, FILTER_VALIDATE_EMAIL) === FALSE) {
        //throw exception if email is not valid
        throw new customException($email);
    }
}

catch (customException $e) {
    //display custom message
    echo $e->errorMessage();
}

?>
```

Exemples: Classe d'exepcions

```
<?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 valid E-Mail address';
        return $errorMsg;
    }
}
```

```
try {
    //check if
    if(filter_var($email, FILTER_VALIDATE_EMAIL) === FALSE) {
        //throw exception if email is not valid
        throw new customException($email);
    }
    //check for "example" in mail address
    if(strpos($email, "example") !== FALSE) {
        throw new Exception("$email is an example e-mail");
    }
}

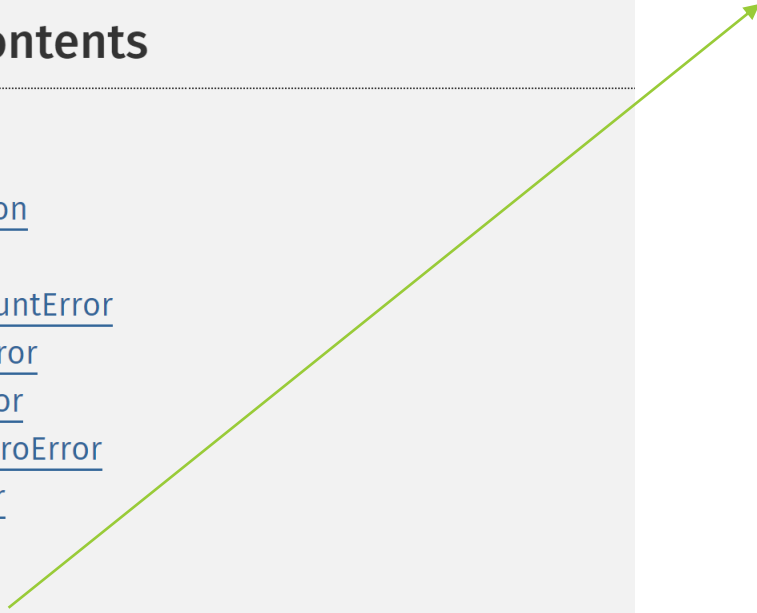
catch (customException $e) {
    echo $e->errorMessage();
}

catch(Exception $e) {
    echo $e->getMessage();
}
?>
```


Classes de PHP per tractar errors i excepcions

Predefined Exceptions

Table of Contents

- [Exception](#)
 - [ErrorException](#)
 - [Error](#)
 - [ArgumentCountError](#)
 - [ArithmeticError](#)
 - [AssertionError](#)
 - [DivisionByZeroError](#)
 - [CompileError](#)
 - [ParseError](#)
 - [TypeError](#)
- 

```
TypeError extends Error {
```

```
/* Inherited properties */  
protected string \$message ;  
protected int \$code ;  
protected string \$file ;  
protected int \$line ;
```

```
/* Inherited methods */  
final public Error::getMessage ( void ) : string  
final public Error::getPrevious ( void ) : Throwable  
final public Error::getCode ( void ) : mixed  
final public Error::getFile ( void ) : string  
final public Error::getLine ( void ) : int  
final public Error::getTrace ( void ) : array  
final public Error::getTraceAsString ( void ) : string  
public Error::\_\_toString ( void ) : string  
final private Error::\_\_clone ( void ) : void  
}
```

Classes de PHP per tractar errors i excepcions

```
TypeError extends Error {  
  
    /* Inherited properties */  
    protected string $message ;  
    protected int $code ;  
    protected string $file ;  
    protected int $line ;  
  
    /* Inherited methods */  
    final public Error::getMessage ( void ) : string  
    final public Error::getPrevious ( void ) : Throwable  
    final public Error::getCode ( void ) : mixed  
    final public Error::getFile ( void ) : string  
    final public Error::getLine ( void ) : int  
    final public Error::getTrace ( void ) : array  
    final public Error::getTraceAsString ( void ) : string  
    public Error::__toString ( void ) : string  
    final private Error::__clone ( void ) : void  
}
```

An example:

```
<?php  
  
function test($x):int {  
    return $x;  
}  
  
try {  
    test('ss');  
}catch(TypeError $e){  
    echo "Error !";  
}
```