



Excepciones

Excepciones

En PHP 5 se introdujo un modelo de excepciones similar al existente en otros lenguajes de programación:

- El código susceptible de producir algún error se introduce en un bloque try
- Cuando se produce algún error, se lanza una excepción utilizando la instrucción throw
- Después del bloque try debe haber como mínimo un bloque catch encargado de procesar el error
- Si una vez acabado el bloque try no se ha lanzado ninguna excepción, se continúa con la ejecución en la línea siguiente al bloque o bloques catch
- Si hay algo que se deba ejecutar tanto si se produce una excepción como si no se produce, lo pondremos dentro de un bloque finally, después del último bloque catch



Excepciones

Para poder capturar los distintos tipos de excepciones que tenemos disponibles en php debemos utilizar los bloques **try catch finally**

```
try {  
    //todo el código a ejecutar  
}  
  
catch(Error $e) { //se ejecuta cuando sucede un error fatal  
}  
  
catch(Exception $e) { //se ejecuta cuando hay una excepción general o de usuario  
}  
  
finally{ //se ejecutará siempre se haya ejecutado el catch o no.  
}
```



Nuevas excepciones en PHP 7

El [manejo de errores en php 7](#) se simplifica gracias a la [interfaz Throwable](#), la cual es capaz de capturar los dos tipos de excepciones.

Nuevas excepciones en PHP 7

Jerarquía de Error

- Throwable
 - Error
 - ArithmeticError
 - DivisionByZeroError
 - AssertionError
 - ParseError
 - TypeError
 - Exception
 - ...

Excepciones Exception y Error

- **Exception** es la clase base para todas las excepciones en PHP 5, y la clase base para todas las excepciones de usuario en PHP 7.
<http://php.net/manual/es/class.exception.php>
- Para los **errores fatales** y recuperables, PHP 7 lanza excepciones de la clase **Error**. <http://php.net/manual/es/class.error.php>
- Proporcionan métodos para obtener información de la excepción y de traza o getMessage. Devuelve el mensaje, en caso de que se haya puesto alguno o getCode. Devuelve el código de error si existe

Para lanzar una excepción no es necesario indicar ningún parámetro, aunque de forma opcional se puede pasar un mensaje de error y también un código de error.

- • Para unificar ambos sistemas las dos clases implementan la interfaz **Throwable**



Excepciones

```
$var = 1;
try
{
    // Como es un error fatal. Lanza una excepción Error en PHP 7

    $var->method();
}
catch (Error $e){

    echo $e->getMessage();
}
```


Excepciones en PHP. Bloques try y catch

```
class Pruebas
{
    public function division($a, $b)
    {
        if($b === 0)
        {
            //La división por 0 no es error fatal, tenemos que lanzar la excepción
            throw new DivisionByZeroError("División por cero capturada");
        }
        return $a / $b;
    }
}

try
{
    $prueba = new Pruebas;
    $division = $prueba->division(10, 0);
    echo $division;
}
catch(DivisionByZeroError $e)
{
    echo $e->getMessage();
}
```


Excepciones y errores fatales

- En el pasado era casi imposible manejar errores fatales en PHP
- Estos provocaban la detención del script
- En PHP 7 cuando se produce un error fatal se lanza una excepción
- Si la excepción no se maneja utilizando un bloque try catch, el script se detiene

Errores en PHP 7

- PHP 7 cambia la mayoría de los errores notificados por PHP
- La mayoría de los errores ahora son notificados lanzando excepciones de la clase `Error`.
- Al igual que las excepciones normales, las excepciones `Error` se propagarán hasta alcanzar el primer bloque *catch* coincidente
- Si no hay bloques coincidentes, será invocado cualquier manejador de excepciones predeterminado instalado con `set_exception_handler()`. Ver ejemplo de manejador.
- Si no hubiera ningún manejador de excepciones predeterminado, la excepción será convertida en un error fatal y será manejada como un error tradicional



Excepciones error

- La jerarquía de Error no hereda de Exception. Ambas heredan de Throwable
- El siguiente código no capturará excepciones generadas por errores fatales de PHP 7:

```
try { ... }
```

```
catch (Exception $e)
```

```
{ ... }
```

- Se requiere para ello un bloque como el siguiente:
 - `catch (Error $e)`