Excepciones en el lenguaje PHP

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.

Ejemplo

```php
$var = 1;
try
{
    $var->method(); // Lanza una excepción Error en PHP 7
}
catch (Error $e)
{
    echo $e->getMessage();
}
```

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.

Las clases 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.
- Para los errores fatales y recuperables de PHP 7 se lanzan excepciones de la clase **Error**.
- Proporcionan métodos para obtener información de la excepción y de traza.
  - **getMessage**. Devuelve el mensaje, en caso de que se haya puesto alguno.
  - **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.

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()**.
- Si no hubiera ningún manejador de excepciones predeterminado, la excepción será convertida en un error fatal y será manejada como un error tradicional.

Excepciones tipo Error

- La jerarquía de Error no hereda de **Exception**.
- El siguiente código no capturará excepciones de PHP 7:

```
try {...}
catch (Exception $e)
{ ... }
```

- Se requiere, por tanto, un bloque
  - catch (Error $e)

Capturar excepciones Error

```php
$var = 1;
try
{
    $var->method(); // Throws an Error object in PHP 7.
}
catch (Error $e) {
    // Handle error
}
```

Excepciones definidas por el usuario

- Podemos definir excepciones personalizadas, heredando de la clase **Exception**.
- Si generas con **throw** una excepción propia, esta será manejada por el primer bloque catch que la capture de forma específica.
- Si no existe un bloque **catch** específico de la excepción, esta será manejada por el primer bloque **catch** que capture la excepción **Exception**.
- Es importante poner los bloques **catch** de las excepciones específicas antes que los de las generales.

Ejemplo

```
class ValidationExcepcion extends Exception {
   public function __construct($campo, $valor) {
       if (empty($valor))
           $message = "El campo $campo está vacío";
       else
           $message = "El campo $campo no es correcto. Valor actual: $valor";
       parent::__construct($message, 0, null);
   }
   public function __tostring() {
       return $this->getMessage();
   }
   public function RegistraError() {
       error_log($this->getMessage(), 0);
   }
}
```

Ejemplo de uso

```
try {
    if (isset($_POST['enviar'])) {
        if (isset($_POST['usuario'])) {
            if (empty($_POST['usuario']))
                throw new ValidationExcepcion("usuario", $_POST['usuario']);
            elseif($_POST['usuario'] !== "alex")
                throw new ValidationExcepcion("usuario", $_POST['usuario']);
            else
                echo "Validación OK";
        }
        else
            throw new ValidationExcepcion("usuario", "");
    }
}
catch (ValidationExcepcion $e) {
    echo $e;
    $e->RegistraError();
}
```