



# Web advanced

## Exceptions & errors

### **DE HOGESCHOOL MET HET NETWERK**

Hogeschool PXL – Dep. PXL-IT – Elfde-Liniestraat 26 – B-3500 Hasselt  
[www.pxl.be](http://www.pxl.be) - [www.pxl.be/facebook](http://www.pxl.be/facebook)



# Voorbeeld1

▼ src  
▼ Util  
  Math.php  
  autoload.php  
  app.php

src/Util/Math.php<sup>1</sup>

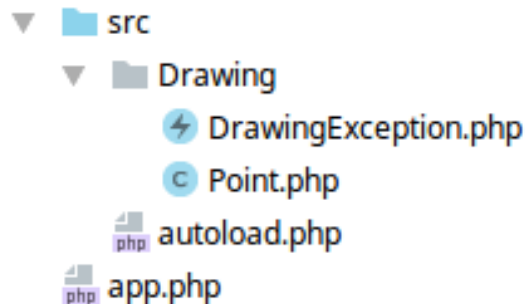
```
<?php namespace Util;
class Math
{
    public static function sum()
    {
        $arguments=func_get_args();
        $sum=0;
        foreach ($arguments as $argument){
            if ( is_numeric($argument) ){
                $sum+=$argument;
            } else {
                throw new \Exception("$argument is not numeric");
            }
        }
        return $sum;
    }
}
```

# Voorbeeld1

app.php

```
<?php
require_once 'src/autoload.php';
use Util\Math;
try{
    $sum=Math::sum( 1, 2 );
    print($sum);
} catch ( Exception $exception ) {
    print( $exception->getMessage() );
}
```

# Eigen exception def.



Exception 1 wordt opgeworpen in setX  
opgevangen in constructor  
vanuit constructor werpen we een nieuwe  
exception op, maar de info over Exception 1  
blijft behouden (komt in \$previous)

## src/Drawing/DrawingException.php

```
<?php namespace Drawing;  
  
class DrawingException extends \Exception  
{  
    public function __construct($message, $code = 0,  
                                \Exception $previous = null)  
    {  
        parent::__construct($message, $code, $previous);  
    }  
}
```

## src/Drawing/Point.php

```
<?php namespace Drawing;

class Point
{
    private $x;
    private $y;
    const MAX_XY = 100;

    public function __construct($x, $y)
    {
        try {
            $this->setX($x);
        } catch (DrawingException $drawingException) {
            throw new DrawingException("$x out of bounds".
                " in __construct", 0, $drawingException);
        }
        try {
            $this->setY($y);
        } catch (DrawingException $drawingException) {
            throw new DrawingException("$y out of bounds".
                " in __construct", 0, $drawingException);
        }
    }

    public function setX($x)
    {
        if($x < 0 || $x > self::MAX_XY ){
            throw new DrawingException("$x out of bounds".
                " in setX");
        }
        $this->x = $x;
    }
}
```

## app.php

```
<?php
require_once 'src/autoload.php';

use Drawing\Point;
use Drawing\DrawingException;
try{
    $point=new Point(111,1);
}catch (Exception $exception){
    print( $exception);
}
```

---

[/usr/bin/php /home/jan/PhpstormProjects/example/app.php](#)

Drawing\DrawingException: 111 out of bounds in setX in [/home/jan/PhpstormProjects/example/src/Drawing/Point.php:25](#)

Stack trace:

#0 [/home/jan/PhpstormProjects/example/src/Drawing/Point.php\(12\)](#): Drawing\Point->setX(111)

#1 [/home/jan/PhpstormProjects/example/app.php\(7\)](#): Drawing\Point->\_\_construct(111, 1)

#2 {main}

Next Drawing\DrawingException: 111 out of bounds in \_\_construct in [/home/jan/PhpstormProjects/example/src/Drawing/Point.php:14](#)

Stack trace:

#0 [/home/jan/PhpstormProjects/example/app.php\(7\)](#): Drawing\Point->\_\_construct(111, 1)

#1 {main}

Process finished with exit code 0

|

# Errors

## 3. Errors

Errors kunnen niet behandeld worden in een try catch-block.

Bijvoorbeeld, de functie `fopen` genereert een error van type `E_WARNING` bij een niet bestaand bestand.

<http://php.net/manual/en/function.fopen.php>

### app.php

```
<?php
$f=fopen('ikbestaniet.txt', 'r+');
```

```
/usr/bin/php /home/jan/PhpstormProjects/example/app.php
```

```
PHP Warning: fopen(ikbestaniet.txt): failed to open stream: No such file or directory in /home/jan/PhpstormProjects/example/app.php
```

# Errors

Er zijn verschillende niveaus van errors gedefinieerd:

<http://php.net/manual/en/errorfunc.constants.php>

Afhankelijk van het niveau van de error wordt de verdere executie van het PHP-script aangehouden of niet gestaakt.

In het configuratiebestand `php.ini` wordt geconfigureerd welke errors getoond moeten worden.

**`/etc/php/7.0/cli/php.ini`**

```
...  
;   E_ALL & ~E_NOTICE & ~E_STRICT (Show all errors, except for  
notices and coding standards warnings.)  
...  
error_reporting = E_ALL & ~E_DEPRECATED & ~E_STRICT
```



# Errors

error\_reporting: php.ini overschrijven

**app.php**

```
<?php
error_reporting(E_ALL);
print($ikBestaNiet);
```

/usr/bin/php /home/jan/PhpstormProjects/example/app.php

PHP Notice: Undefined variable: ikBestaNiet in /home/jan/PhpstormProjects/example/app.php on line 3

# Errors

Het printen van de error-boodschap kan ook onderdrukt worden via een @-symbool.

## app.php

```
<?php  
error_reporting(E_ALL);  
@ print($ikBestaNiet);
```

```
/usr/bin/php /home/jan/PhpstormProjects/example/app.php
```

```
Process finished with exit code 0
```

Via `trigger_error` kan een error gegenereerd worden.

<http://php.net/manual/en/function.trigger-error.php>

De onderstaande code genereert een `E_USER_ERROR` voor een niet bestaand bestand of voor een bestand dat niet leesbaar is. Deze error staakt de executie van `app.php`

#### **src/Util/File.php**

```
<?php namespace Util;

class File
{
    public static function readFile($fileName)
    {
        if (file_exists($fileName) && is_readable($fileName)){
            $file = fopen($fileName, 'r');
            $contents = fread($file, filesize($fileName));
            fclose($file);
            return $contents;
        }else{
            trigger_error("$fileName not correct", E_USER_ERROR);
        }
    }
}
```

#### **app.php**

```
<?php
require 'src/autoload.php';
error_reporting(E_ALL);
use Util\File;
print(File::readFile('ikbestaniet.txt'));
print('done');
```

# Errors

Ten slotte kan er ook een errorhandler gespecificeerd worden.

<http://php.net/manual/en/function.set-error-handler.php>

## app.php

```
<?php
require 'src/autoload.php';
set_error_handler('errorHandler');
function errorHandler($errno, $errstr, $file, $line)
{
    print("$errstr\n");
    exit();
}
error_reporting(E_ALL);
use Util\File;
print(File::readFile('ikbestaniet.txt'));
print('done');
```