



# PHP

# Exception-Handling



- Häufig findet man folgende Fehlerbehandlung in PHP-Skripten...
  - sofortiger Abbruch mit `die` oder `exit` oder
    - `die("Wenden Sie sich an den Administrator");`
  - Unterdrückung von Fehlermeldungen mit `@`
    - `@$f = fopen("myfile.txt", "r");`
- beide Arten bringen Probleme mit sich...
  - eine unvollständige Seite wird zum Browser geschickt
  - es entstehen leere Seiten oder Layout-Katastrophen
  - der Anwender erhält keine Rückmeldung



- PHP bietet nun eine echte Ausnahmebehandlung
- Fehler können mit try/catch behandelt werden
- in den neueren PHP-Versionen existiert auch try/catch/finally sowie try/finally
- dabei können sinnvolle Meldungen ausgegeben werden
- die Web-Seite kann ordentlich abgeschlossen werden

```
function problematisch() {  
    throw new Exception("Meine Fehlermeldung");  
}  
  
try {  
    problematisch();  
}  
catch (Exception $e) { // Fehlerbehandlung  
    echo $e->getMessage();  
}
```



- die Basisklasse aller Ausnahmen ist die Klasse Exception
  - beliebige eigene Ausnahmen sind daraus ableitbar
  - es gibt aber keine vordefinierten Exception-Klassenhierarchie in PHP, also nicht sowas wie **NullPointerException** in Java
- noch problematischer ist jedoch:  
Keine der vorhandenen PHP-Funktionen wirft Ausnahmen!



```
class MyException extends Exception{
    [...]
}

class AnotherException extends Exception{
    [...]
}

try {
    // Mache etwas Gefährliches
    throw new MyException("Mein Fehler", 99);
} catch (MyException $e) {
    echo "MyException: ", $e->getMessage();
} catch (AnotherException $e) {
    echo "AnotherException: ", $e->getMessage();
}
```