

# 15

## 오류 처리 함수

PHP는 오류에 대한 처리를 할 수 있는 몇 가지 함수들을 제공합니다.

### 15.1 오류

PHP는 다양한 오류의 정의와 메시지 코드를 지원합니다. 내장 함수 `error_reporting()`는 PHP 에러를 정의합니다.

| 내장 함수 |

```
int error_reporting ([ int $level ] )
```

PHP에서 사용되는 대표적인 에러 코드와 기호는 다음과 같습니다.

- 1: E\_ERROR 에러를 출력하고 스크립트의 실행을 중단합니다. 메모리 할당 에러 등의 복구가 힘든 문제의 에러를 의미합니다.

- 2: E\_WARNING 경고를 출력하지만 스크립트는 정상적으로 실행됩니다.
- 4: E\_PARSE 컴파일 타임 구문 분석 오류 메시지입니다.
- 8: E\_NOTICE 뭔가 에러를 감지했으나 출력은 하지 않습니다.
- 16: E\_CODE\_ERROR PHP 코어에 의하여 생성된 에러 메시지입니다.
- 32: E\_CODE\_WARNING PHP 코어에 의하여 생성된 에러 메시지입니다.

#### 예제 파일 | `error_report.php`

```

1  <?php
2
3      // Turn off all error reporting
4      error_reporting(0);
5
6      // Report simple running errors
7      error_reporting(E_ERROR | E_WARNING | E_PARSE);
8
9      // Reporting E_NOTICE can be good too (to report uninitialized
10     // variables or catch variable name misspellings ...)
11     error_reporting(E_ERROR | E_WARNING | E_PARSE | E_NOTICE);
12
13     // Report all errors except E_NOTICE
14     error_reporting(E_ALL & ~E_NOTICE);
15
16     // Report all PHP errors (see changelog)
17     error_reporting(E_ALL);
18
19     // Report all PHP errors
20     error_reporting(-1);
21
22     // Same as error_reporting(E_ALL);
23     ini_set('error_reporting', E_ALL);
24
25     ?>

```

## 15.2 오류 출력

PHP에서 발생한 오류에 대한 정보를 가져오고 로그를 출력할 수 있습니다.

### | 내장 함수 |

```
bool error_log ( string $message [, int $message_type = 0 [, string $destination [, string $extra_headers ]]] )
```

내장 함수 `error_log()`는 메시지를 오류 처리 루틴으로 전달합니다. 지정한 타입에 따라서 메시지를 출력할 곳을 지정할 수 있습니다.

- 타입0: `php.ini`에 지정된 시스템에 에러를 출력합니다.
- 타입1: 지정한 이메일로 에러 메시지를 출력합니다.
- 타입2: 호스트, IP 주소의 PHP 디버깅으로 출력합니다. 디버깅 출력은 `remote debugging` 설정이 되어 있어야 합니다.
- 타입3: 지정한 파일로 출력합니다.

### 예제 파일 | `error_log.php`

```
1  <?php
2      // 지정한 로그 파일에 에러를 출력합니다.
3      $errFile = "./my-errors.log";
4      error_log("You messed up!", 3, $errFile);
5
6  ?>
```

### | 내장 함수 |

```
array error_get_last ( void )
```

내장 함수 `error_get_last()`는 마지막으로 발생한 오류를 가지고 옵니다.

#### 예제 파일 | [error\\_report.php](#)

```
1  <?php
2      echo $a;
3      print_r(error_get_last());
4
5  ?>
```

#### | 내장 함수 |

**void error\_clear\_last** ( void )

내장 함수 `error_clear_last()`는 가장 최근 오류를 지웁니다.

#### 예제 파일 | [error\\_clear\\_last.php](#)

```
1  <?php
2      var_dump(error_get_last());
3      error_clear_last();
4      var_dump(error_get_last());
5
6      @$a = $b;
7
8      var_dump(error_get_last());
9      error_clear_last();
10     var_dump(error_get_last());
11
12  ?>
```

#### | 내장 함수 |

**bool trigger\_error** ( string \$error\_msg [, int \$error\_type = E\_USER\_NOTICE ] )

내장 함수 `trigger_error()`는 사용자 수준의 오류/경고/통지 메시지를 생성합니다.

예제 파일 | `trigger_error.php`

```
1 <?php
2     trigger_error("Cannot divide by zero", E_USER_ERROR);
3
4 ?>
```

#### 콘솔 출력

[Mon Aug 21 16:59:44 2017] ::1:57186 [500]: /trigger\_error.php - **Cannot divide by zero** in C:\php-7.1.4-Win32-VC14-x86\trigger\_error.php on line 3

## 15.3 역추적

PHP는 역추적에 관련된 몇 개의 내장 함수들을 지원합니다.

### | 내장 함수 |

```
array debug_backtrace ([ int $options = DEBUG_BACKTRACE_PROVIDE_OBJECT  
[, int $limit = 0 ]])
```

내장 함수 `debug_backtrace()`는 역추적을 실행합니다.

예제 파일 | `debug_backtrace.php`

```
1 <?php
2     function test($str)
3     {
4         echo "Hello World! $str<br>";
5         var_dump(debug_backtrace());
6     }
7
8     test('jiny');
9
10
11 ?>
```

#### 콘솔 출력

```
Hello World! jiny
array(1) { [0]=> array(4) { ["file"]=> string(47) "C:\php-7.1.4-Win32-VC14-
x86\debug_backtrace.php" ["line"]=> int(8) ["function"]=> string(4) "test"
["args"]=> array(1) { [0]=> string(4) "jiny" } } }
```

#### | 내장 함수 |

```
void debug_print_backtrace ([ int $options = 0 [, int $limit = 0 ] ] )
```

내장 함수 `debug_print_backtrace()`는 역추적을 출력합니다.

#### 예제 파일 | `debug_print_backtrace.php`

```
1  <?php
2
3      function a() {
4          b();
5      }
6
7      function b() {
8          c();
9      }
10
11     function c() {
12         debug_print_backtrace();
13     }
14
15     a();
16
17  ?>
```

#### 화면 출력

```
#0 c() called at [C:\php-7.1.4-Win32-VC14-x86\debug_print_backtrace.php:8]
#1 b() called at [C:\php-7.1.4-Win32-VC14-x86\debug_print_backtrace.php:4]
#2 a() called at [C:\php-7.1.4-Win32-VC14-x86\debug_print_backtrace.php:15]
```

## 15.4 오류 핸들

PHP는 오류 처리 핸들에 관련된 몇 가지 함수들을 지원합니다.

### | 내장 함수 |

```
mixed set_error_handler ( callable $error_handler [, int $error_types = E_ALL | E_STRICT ] )
```

내장 함수 `set_error_handler()`는 사용자 정의 오류 처리기 함수를 설정합니다.

### | 내장 함수 |

```
bool restore_error_handler ( void )
```

내장 함수 `restore_error_handler()`는 이전 오류 처리 함수를 복원합니다.

### | 내장 함수 |

```
callable set_exception_handler ( callable $exception_handler )
```

내장 함수 `set_exception_handler()`는 사용자 정의 예외 처리 함수를 설정합니다.

### | 내장 함수 |

```
bool restore_exception_handler ( void )
```

내장 함수 `restore_exception_handler()`는 이전에 정의된 예외 핸들러 함수를 복원합니다.

