jinyPHP 3권 - 15. 오류처리 함수

**15. 오류처리 함수**

====================

PHP는 오류에 대한 처리를 할 수 있는 몇가지 함수들을 제공합니다.

**15.1 오류**

====================

PHP는 다양한 오류의 정의와 메시지 코드를 지원합니다. 내부함수 error\_reporting()는 PHP 에러를 정의합니다.

**|내부함수|**

|  |
| --- |
| int **error\_reporting** ([ int $level ] ) |

PHP에서 사용되는 대표적인 에러코드와 기호는 다음과 같습니다.

* 1: E\_ERROR 에러를 출력하고 스크립트의 실행을 중단합니다.. 메모리 할당 에러등의 복구가 힘든 문제의 에러를 의미합니다.
* 2: E\_WARNING 경고를 출력하지만 스크립트는 정상적으로 실행됩니다.
* 4: E\_PARSE
* 8: E\_NOTE 뭔가 에러를 감지를 하였으나 출력은 하지 않습니다.
* 16: E\_CODE\_ERROR PHP 코어에 의하여 생성된 에러 메시지 입니다.
* 32: E\_CODE\_WARNING PHP 코어에 의하여 생성된 에러 메시지 입니다.

**예제) error\_report.php**

|  |
| --- |
| <?php  // Turn off all error reporting  error\_reporting(0);  // Report simple running errors  error\_reporting(E\_ERROR | E\_WARNING | E\_PARSE);  // Reporting E\_NOTICE can be good too (to report uninitialized  // variables or catch variable name misspellings ...)  error\_reporting(E\_ERROR | E\_WARNING | E\_PARSE | E\_NOTICE);  // Report all errors except E\_NOTICE  error\_reporting(E\_ALL & ~E\_NOTICE);  // Report all PHP errors (see changelog)  error\_reporting(E\_ALL);  // Report all PHP errors  error\_reporting(-1);  // Same as error\_reporting(E\_ALL);  ini\_set('error\_reporting', E\_ALL);  ?> |

**15.2 오류출력**

====================

PHP에서 발생한 오류에 대한 정보를 가지고 오고 로그를 출력할 수 있습니다.

**|내부함수|**

|  |
| --- |
| bool **error\_log** ( string $message [, int $message\_type = 0 [, string $destination [, string $extra\_headers ]]] ) |

내부함수 error\_log()는 메시지를 오류처리 루틴으로 전달합니다. 지정한 타입에 따라서 메시지를 출력할 곳을 지정할 수 있습니다.

* 타입0: php.ini 에 지정된 시스템에 에러를 출력합니다.
* 타입1: 지정한 이메일로 에러메시지를 출력합니다.
* 타입2: 호스트,IP주소의 PHP 디버깅으로 출력합니다. 디버깅 출력은 remote debugging 설정이 되어 있어야 합니다.
* 타입3: 지정한 파일로 출력합니다.

**예제) error\_log.php**

|  |
| --- |
| <?php  // 지정한 로그파일에 에러를 출력합니다.  $errFile = "./my-errors.log";  error\_log("You messed up!", 3, $errFile);  ?> |

**|내부함수|**

|  |
| --- |
| array **error\_get\_last** ( void ) |

내부함수 error\_get\_last()는 마지막으로 발생한 오류 가지고 옵니다.

**예제) error\_get\_last.php**

|  |
| --- |
| <?php  echo $a;  print\_r(error\_get\_last());  ?> |

**|내부함수|**

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

내부함수 error\_clear\_last()는 가장 최근 오류를 지웁니다.

**예제) error\_clear\_last.php**

|  |
| --- |
| <?php  var\_dump(error\_get\_last());  error\_clear\_last();  var\_dump(error\_get\_last());  @$a = $b;  var\_dump(error\_get\_last());  error\_clear\_last();  var\_dump(error\_get\_last());  ?> |

**|내부함수|**

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

내부함수 trigger\_error()는 사용자 수준의 오류 / 경고 / 통지 메시지를 생성합니다.

**예제) trigger\_error.php**

|  |
| --- |
| <?php  trigger\_error("Cannot divide by zero", E\_USER\_ERROR);  ?> |

**콘솔출력)**

[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 내부함수들을 지원합니다.

**|내부함수|**

|  |
| --- |
| array **debug\_backtrace** ([ int $options = DEBUG\_BACKTRACE\_PROVIDE\_OBJECT [, int $limit = 0 ]] ) |

내부함수 debug\_backtrace()는 역 추적을 실행합니다.

**예제) debug\_backtrace.php**

|  |
| --- |
| <?php  function test($str)  {  echo "Hello World! $str<br>";  var\_dump(debug\_backtrace());  }  test('jiny');  ?> |

**화면출력)**

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" } } }

**|내부함수|**

|  |
| --- |
| void **debug\_print\_backtrace** ([ int $options = 0 [, int $limit = 0 ]] ) |

내부함수 debug\_print\_backtrace()는 역추적을 출력합니다.

**예제) debug\_print\_backtrace.php**

|  |
| --- |
| <?php  function a() {  b();  }  function b() {  c();  }  function c(){  debug\_print\_backtrace();  }  a();  ?> |

**화면출력)**

#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 오류처리 핸들에 관련된 몇가지 함수들을 지원합니다.

**|내부함수|**

|  |
| --- |
| mixed **set\_error\_handler** ( callable $error\_handler [, int $error\_types = E\_ALL | E\_STRICT ] ) |

내부함수 set\_error\_handler()는 사용자 정의 오류 처리기 함수를 설정합니다.

**|내부함수|**

|  |
| --- |
| bool **restore\_error\_handler** ( void ) |

내부함수 restore\_error\_handler()는 이전 오류 처리 함수를 복원합니다.

**|내부함수|**

|  |
| --- |
| callable **set\_exception\_handler** ( callable $exception\_handler ) |

내부함수 set\_exception\_handler()는 사용자 정의 예외 처리 함수를 설정합니다.

**|내부함수|**

|  |
| --- |
| bool **restore\_exception\_handler** ( void ) |

내부함수 restore\_exception\_handler()는 이전에 정의 된 예외 핸들러 함수를 복원합니다.