jinyPHP 3권 - 18. 스크립트

**18. 스크립트**

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

**18.1 종료**

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

내부함수 exit()는 스크립트를 종료합니다.

**|내부함수|**

|  |
| --- |
| void **exit** ([ string $status ] ) |

status가 문자열이면이 종료 직전의 문자열을 출력합니다. status가 정수일때는 값이 화면에 출력이 되지는 않습니다. 정수값은 0~255 사이의 값을 가집니다. 만일 0의 값은 정상적인 종료를 의미합니다.

**예제) exit.php**

|  |
| --- |
| <?php  //일반적인 프로그램 종료  exit;  exit();  exit(0);    // 문자열을 출력하고 종료합니다.  exit("finish");  //exit with an error code  exit(1);  exit(0376); //octal  ?> |

**|내부함수|**

|  |
| --- |
| void **die** ( [ string $status ] ); |

내부함수 die()는 스크립트를 종료합니다. exit() 함수와 동일한 동작을 수행합니다.

**예제) die.php**

|  |
| --- |
| <?php  die ("데이터베이스 접속실패");  ?> |

**|내부함수|**

|  |
| --- |
| void **register\_shutdown\_function** ( callable $callback [, mixed $parameter [, mixed $... ]] ) |

스크립트 실행이 끝나거나 exit ()가 호출 된 후에 실행될 콜백을 등록합니다.

**예제) register\_shutdown\_function.php**

|  |
| --- |
| <?php  function shutdown() {  // 스크립트가 완료되기 전에 마지막 작업을 수행 할 수 있습니다.  echo 'Script executed with success', PHP\_EOL;  }  register\_shutdown\_function('shutdown');  ?> |

**화면출력)**

Script executed with success

**18.2 코드실행**

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

내부함수 eval()는 문자열로 된 소스코드를 실행합니다. eval() 함수는 해킹의 용도로 사용될 수 있기 때문에 주의하여 사용을 하여야 합니다.

**|내부함수|**

|  |
| --- |
| mixed **eval** ( string $code ) |

**예제) eval.php**

|  |
| --- |
| <?php  $string = 'cup';  $name = 'coffee';  // 작은 따옴표는 일반 텍스트로 문자를 인식합니다.  // $string, $name 는 변수명 자체를 화면에 출력합니다.  $str = 'This is a $string with my $name in it.';  echo $str. "<br>";  // PHP 코드 처리됩니다.  $code = "\$aaa = \"$str\";";  eval($code);  echo $aaa;  ?> |

**화면출력)**

This is a $string with my $name in it.

This is a cup with my coffee in it.

**18.3 접속상태**

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

PHP 스크립트는 내부적으로 3가지의 접속 상태가 존재합니다.

0 - NORMAL

1 - ABORTED

2 - TIMEOUT

**18.3.1 connection\_status**

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

내부함수 connection\_status()는 연결 상태 반환합니다.

**|내부함수|**

|  |
| --- |
| int **connection\_status** ( void ) |

NORMAL 상태는 보통 스크립트를 실행하고 정상적인 동작을 할때 입니다.

**예제) connection\_status.php**

|  |
| --- |
| <?php  $status = connection\_status();  if ($status!=0){  die;  } else {  echo $status."<br>";  echo "접속상태 입니다.";  }  ?> |

**화면출력)**

0

접속상태 입니다.

**18.3.2 ignore\_user\_abort**

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

ABORTED 상태는 스크립트를 동작하고 있는 중에 접속 클라이언트의 사용자가 임의적으로 중지 버튼을 누르거나 접속을 차단했을때의 상태입니다. 또는 시간제한에 의하여 타임아웃 상태가 된경우 입니다.

기본적으로 ABORTED 상태가 되면 스크립트는 중단을 합니다. 하지만 ABORTED 상태가 되어도 실행 요청한 스크립트가 지속적으로 동작하여 정상수행을 다 마치길 바랄때도 있을 것입니다.

이런경우 PHP.ini 설정의 ignore\_user\_abort 변경하거나 아파치와 같은 웹서버에서 설정을 변경할 수도 있습니다. 또는 PHP 소스상에서 ignore\_user\_abort() 함수를 통하여 설정을 할 수도 있습니다.

ignore\_user\_abort() 함수는 클라이언트 연결 끊기가 스크립트 실행을 중단해야하는지 여부 설정합니다.

**|내부함수|**

|  |
| --- |
| int **ignore\_user\_abort** ([ bool $value ] ) |

**예제) ignore\_user\_abort.php**

|  |
| --- |
| <?php  echo 'PHP connection';  ignore\_user\_abort(true);  set\_time\_limit(0);  while(1) {  // 접속상태 실패  if(connection\_status() != CONNECTION\_NORMAL)  {  break;  }  // 10 초간 지연  sleep(10);  }  ?> |

**18.3.3 connection\_aborted**

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

보통 스크립트가 종료될때 register\_shutdown\_function()를 통하여 종료처리를 할 수 있습니다. 하지만 클라이언트의 연결이 끊어 져서 스크립트를 종료해야 할 경우도 있습니다.

**|내부함수|**

|  |
| --- |
| int **connection\_aborted** ( void ) |

이런 경우 onnection\_aborted() 함수를 통하여 별도의 처리를 수행할 수 있습니다.

**|내부함수|**

|  |
| --- |
| bool **set\_time\_limit** ( int $seconds ) |

내부함수 set\_time\_limit()는 최대 실행 시간을 제한합니다.

**18.4 지연함수, 실행시간 설정**

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

**18.4.1 지연**

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

내부함수 sleep()는 딜레이를 적용합니다. 입력되는 값은 초 단위로 입력합니다.

**|내부함수|**

|  |
| --- |
| int **sleep** ( int $seconds ) |

**예제) sleep.php**

|  |
| --- |
| <?php  // 현재 시간  echo date('h:i:s') . "<br>";  // 10 초간 지연  sleep(10);  // 시간 다시 표시  echo date('h:i:s') . "<br>";  ?> |

**화면출력)**

09:01:09

09:01:19

**|내부함수|**

|  |
| --- |
| void **usleep** ( int $micro\_seconds ) |

내부함수 usleep()은 마이크로 초 단위로 지연 합니다.

**예제) usleep.php**

|  |
| --- |
| <?php  // 현재 시간  echo date('h:i:s') . "<br>";  // 2 초간 대기합니다.  usleep(2000000);  // 시간 다시 표시  echo date('h:i:s') . "<br>";  ?> |

**화면출력)**

09:05:09

09:05:11

**18.4.2 실행시간**

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

스크립트의 실행의 최대 시간을 설정할 수 있습니다. 기본값은 30초 입니다. 작업이 많이 필요한 스크립트의 경우 실행 시간을 조정할 수 있습니다.

**|내부함수|**

|  |
| --- |
| void **set\_time\_limit** ( int $seconds ) |

**예제) set\_time\_limit.php**

|  |
| --- |
| <?php  // 스크립트 최대 실행 시간 100초  set\_time\_limit(100);  echo "스크립트시간 100초<br>";  while ($i<=50){  echo "sleep=$i , ";  sleep(1);  $i++;  }  ?> |

**화면출력)**

스크립트시간 100초

sleep= , sleep=1 , sleep=2 , sleep=3 , sleep=4 , sleep=5 , sleep=6 , sleep=7 , sleep=8 , sleep=9 , sleep=10 , sleep=11 , sleep=12 , sleep=13 , sleep=14 , sleep=15 , sleep=16 , sleep=17 , sleep=18 , sleep=19 , sleep=20 , sleep=21 , sleep=22 , sleep=23 , sleep=24 , sleep=25 , sleep=26 , sleep=27 , sleep=28 , sleep=29 , sleep=30 , sleep=31 , sleep=32 , sleep=33 , sleep=34 , sleep=35 , sleep=36 , sleep=37 , sleep=38 , sleep=39 , sleep=40 , sleep=41 , sleep=42 , sleep=43 , sleep=44 , sleep=45 , sleep=46 , sleep=47 , sleep=48 , sleep=49 , sleep=50 ,

스크립트의 최대 실행 시간을 제한합니다. 기본 설정값은 30 초 또는 php.ini에 정의 된 max\_execution\_time 값입니다. 만일, 지정한 값보다 시간을 초과할때는 치명적인 오류를 반환합니다.

스크립트를 실행도중에 set\_time\_limit()를 실행하면 제한 시간 카운터를 0에서 다시 시작합니다. 예로 기본 timeout이 30 초이고, 25초 되는 시점에 set\_time\_limit (20)과 같이 호출을 하게 되면 총 시간은 45 초 동안 실행되게 됩니다.

**예제) set\_time\_limit2.php**

|  |
| --- |
| <?php  set\_time\_limit(20);  while ($i<=10){  echo "i=$i ";  sleep(100);  $i++;  }  ?> |

화면출력)

i=0 i=1 i=2 i=3 i=4 i=5 i=6 i=7 i=8 i=9 i=10

**18.5 스크립트 정보**

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

PHP 스크립트의 정보를 읽어 올 수 있습니다. 이와 관련된 몇 개의 내부함수들을 지원합니다.

**|내부함수|**

|  |
| --- |
| string **get\_current\_user** ( void ) |

내부함수 get\_current\_user()는 현재 PHP 스크립트를 실행하고 있는 소유자(owner)를 확인할 수 있습니다.

**예제) get\_current\_user.php**

|  |
| --- |
| <?php  echo "스크립트 onwer: " . get\_current\_user();  ?> |

**화면출력)**

스크립트 onwer: infoh

**|내부함수|**

|  |
| --- |
| int **getlastmod** ( void ) |

내부함수 getlastmod()는 현재 스크립트의 최종 수정일자를 확인할 수 있습니다. 스크립트 파일일자를 확인하여 업데이트 와 같은 갱신처리 할때 등 유용하게 사용을 할 수 있습니다.

**예제) getlastmod.php**

|  |
| --- |
| <?php  $scriptDate = date ("F d Y H:i:s.", getlastmod());  echo "죄종수정일: " . $scriptDate . "<br>";  if (date ("Y-m-d", getlastmod()) <= "2017-10-03"){  echo "스크립트가 최신버젼이 아닙니다. 업그레이드를 해주세요<br>";  } else {  echo "최신<br>";  }  ?> |

**화면출력)**

죄종수정일: June 21 2017 08:04:13.

스크립트가 최신버젼이 아닙니다. 업그레이드를 해주세요

**|내부함수|**

|  |
| --- |
| int **getmygid** ( void ) |

내부함수 getmygid()는 PHP 스크립트 소유자의 GID를 가지고 옵니다.