Chapter

18

스크립트

18.1 종료

내장 함수 exit()는 스크립트를 종료합니다.

| 내장 함수 |

```
void exit ([ string $status ] )
```

status가 문자열이면이 종료 직전의 문자열을 출력합니다. status가 정수일 때는 값이 화면에 출력되지는 않습니다. 정수값은 0~255 사이의 값을 가집니다. 만일 0의 값은 정상적인 종료를 의미합니다.

예제 파일 | exit.php

```
1  <?php
2
3      //일반적인 프로그램 종료
4      exit;
```

```
    exit();
    exit(0);

    // 문자열을 출력하고 종료합니다.
    exit("finish");

    //exit with an error code
    exit(1);
    exit(0376); //octal

?>
```

| 내장 함수 |

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

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

예제 파일 | **die.php**

```
<?php
   die ("데이터베이스 접속 실패");
?>
```

| 내장 함수 |

void **register_shutdown_function** (callable $callback [, mixed $parameter [, mixed $...]])

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

예제 파일 | **register_shutdown_function.php**

```
<?php
   function shutdown() {
      // 스크립트가 완료되기 전에 마지막 작업을 수행할 수 있습니다.
      echo 'Script executed with success', PHP_EOL;
```

```
    }

    register_shutdown_function('shutdown');

?>
```

화면 출력

```
Script executed with success
```

18.2 코드 실행

내장 함수 eval()은 문자열로 된 소스 코드를 실행합니다. eval() 함수는 해킹의 용도로 사용될 수 있기 때문에 주의하여 사용해야 합니다.

| 내장 함수 |

```
mixed eval ( string $code )
```

예제 파일 | eval.php

```
<?php
    $string = 'cup';
    $name = 'coffee';

    // 작은따옴표는 일반 텍스트로 문자를 인식합니다.
    // $string, $name는 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;
?>
```

화면 출력

```
This is a $string with my $name in it.
This is a cup with my coffee in it.
```

18.3 접속 상태

PHP 스크립트는 내부적으로 세 가지의 접속 상태가 존재합니다.

0 – NORMAL
1 – ABORTED
2 – TIMEOUT

18.3.1 connection_status

내장 함수 connection_status()는 연결 상태 반환합니다.

| 내장 함수 |

```
int connection_status ( void )
```

NORMAL 상태는 보통 스크립트를 실행하고 정상적인 동작을 할 때입니다.

예제 파일 | connection_status.php

```
<?php
   $status = connection_status();
   if ($status!=0) {
      die;
   } else {
      echo $status."<br>";
      echo "접속상태 입니다.";
   }

?>
```

화면 출력

```
0
접속상태 입니다.
```

18.3.2 ignore_user_abort

ABORTED 상태는 스크립트를 동작하고 있는 중에 접속 클라이언트의 사용자가 임의적으로 중지 버튼을 누르거나 접속을 차단했을 때의 상태입니다. 또는 시간 제한에 의하여 타임아웃 상태가 된 경우입니다.

기본적으로 ABORTED 상태가 되면 스크립트는 중단합니다. 하지만 ABORTED 상태가 되어도 실행 요청한 스크립트가 지속적으로 동작하여 정상 수행을 다 마치길 바랄 때도 있을 것입니다.

이런 경우 PHP.ini 설정의 ignore_user_abort를 변경하거나 아파치와 같은 웹 서버에서 설정을 변경할 수 있으며, 또는 PHP 소스상에서 ignore_user_abort() 함수를 통하여 설정할 수도 있습니다.

ignore_user_abort() 함수는 클라이언트 연결 끊기가 스크립트 실행을 중단해야 하는지 여부를 설정합니다.

| 내장 함수 |

```
int ignore_user_abort ([ bool $value ] )
```

예제 파일 | ignore_user_abort.php

```
1  <?php
2     echo 'PHP connection';
3
4     ignore_user_abort(true);
5     set_time_limit(0);
6
7     while (1) {
```

```
8
9         // 접속상태 실패
10        if (connection_status() != CONNECTION_NORMAL)
11        {
12           break;
13        }
14
15        // 10초간 지연
16        sleep(10);
17     }
18
19  ?>
```

18.3.3 connection_aborted

보통 스크립트가 종료될 때 register_shutdown_function()을 통하여 종료 처리를 할 수 있습니다. 하지만 클라이언트의 연결이 끊겨서 스크립트를 종료해야 할 경우도 있습니다.

| 내장 함수 |

```
int connection_aborted ( void )
```

이런 경우 onnection_aborted() 함수를 통하여 별도의 처리를 수행할 수 있습니다.

| 내장 함수 |

```
bool set_time_limit ( int $seconds )
```

내장 함수 set_time_limit()는 최대 실행 시간을 제한합니다.

18.4 지연 함수, 실행 시간 설정

18.4.1 지연

내장 함수 sleep()은 딜레이를 적용합니다. 입력되는 값은 초 단위로 입력합니다.

| 내장 함수 |

```
int sleep ( int $seconds )
```

예제 파일 | sleep.php

```
<?php

    // 현재 시간
    echo date('h:i:s') . "<br>";

    // 10초간 지연
    sleep(10);

    // 시간 다시 표시
    echo date('h:i:s') . "<br>";

?>
```

화면 출력

```
09:01:09
09:01:19
```

| 내장 함수 |

```
void usleep ( int $micro_seconds )
```

내장 함수 usleep()은 마이크로 초 단위로 지연합니다.

예제 파일 | usleep.php

```php
<?php

    // 현재 시간
    echo date('h:i:s') . "<br>";

    // 2초간 대기합니다.
    usleep(2000000);

    // 시간 다시 표시
    echo date('h:i:s') . "<br>";

?>
```

화면 출력

```
09:05:09
09:05:11
```

18.4.2 실행 시간

스크립트의 실행의 최대 시간을 설정할 수 있습니다. 작업이 많이 필요한 스크립트의 경우 실행 시간을 조정할 수 있습니다.

| 내장 함수 |

void **set_time_limit** (int $seconds)

예제 파일 | set_time_limit-01.php

```php
<?php
    // 스크립트 최대 실행 시간 100초
    set_time_limit(100);
    echo "스크립트 시간 100초<br>";

    while ($i<=50) {
        echo "sleep=$i , ";
```

```
8           sleep(1);
9           $i++;
10      }
11  ?>
```

화면 출력

```
스크립트 시간 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 ,
```

기본 설정값은 30초 또는 php.ini에 정의된 max_execution_time 값입니다. 만일, 지정한 값보다 시간을 초과할 때는 치명적인 오류를 반환합니다.

스크립트를 실행 도중에 set_time_limit()를 실행하면 제한 시간 카운터를 0에서 다시 시작합니다. 예로 기본 timeout이 30초이고, 25초 되는 시점에 set_time_limit(20)과 같이 호출하게 되면 총 시간은 45초 동안 실행됩니다.

예제 파일 | set_time_limit-02.php

```
1   <?php
2
3       set_time_limit(20);
4
5       while ($i<=10) {
6           echo "i=$i ";
7           sleep(100);
8           $i++;
9       }
10
11  ?>
```

화면 출력

```
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 읽어올 수 있습니다. 이와 관련된 몇 가지 내장 함수들을 지원합니다.

| 내장 함수 |

```
string get_current_user ( void )
```

내장 함수 get_current_user()는 현재 PHP 스크립트를 실행하고 있는 소유자(owner)를 확인할 수 있습니다.

예제 파일 | get_current_user.php

```
1   <?php
2       echo "스크립트 onwer: " . get_current_user();
3   ?>
```

화면 출력

```
스크립트 onwer: infoh
```

| 내장 함수 |

```
int getlastmod ( void )
```

내장 함수 getlastmod()는 현재 스크립트의 최종 수정 일자를 확인할 수 있습니다. 스크립트 파일 일자를 확인하여 업데이트와 같은 갱신 처리할 때 등 유용하게 사용할 수 있습니다.

예제 파일 | getlastmod.php

```
1   <?php
2
3       $scriptDate = date ("F d Y H:i:s.", getlastmod());
```

```
4       echo "최종 수정일: " . $scriptDate . "<br>";
5
6       if (date ("Y-m-d", getlastmod()) <= "2017-10-03") {
7          echo "스크립트가 최신 버전이 아닙니다. 업그레이드를 해주세요<br>";
8       } else {
9          echo "최신<br>";
10      }
11
12  ?>
```

화면 출력

```
최종 수정일: June 21 2017 08:04:13.
스크립트가 최신 버전이 아닙니다. 업그레이드를 해주세요
```

| 내장 함수 |

int **getmygid** (void)

내장 함수 getmygid()는 PHP 스크립트 소유자의 GID를 가지고 옵니다.