

19

정보

PHP는 내장 함수를 통하여 PHP의 버전 및 설정 상태 등의 다양한 정보를 읽어서 처리할 수 있습니다.

19.1 시스템

내장 함수 `phpversion()`은 현재 PHP 버전을 가져옵니다.

| 내장 함수 |

```
string phpversion ([ string $extension ] )
```

예제 파일 | `phpversion.php`

```
1  <?php
2      echo phpversion();
3  ?>
```

화면 출력

7.1.4

| 내장 함수 |

```
mixed version_compare ( string $version1 , string $version2 [, string $operator ] )
```

내장 함수 `version_compare()`는 2개의 PHP 버전 번호를 비교합니다.

예제 파일 | `version_compare.php`

```
1  <?php
2      if (version_compare(PHP_VERSION, '7.0.0') >= 0) {
3          echo 'I am at least PHP version 7.0.0, my version: ';
4          echo PHP_VERSION."<br>";
5      }
6
7      if (version_compare(PHP_VERSION, '5.3.0') >= 0) {
8          echo 'I am at least PHP version 5.3.0, my version: ';
9          echo PHP_VERSION."<br>";
10     }
11
12  ?>
```

화면 출력

```
I am at least PHP version 7.0.0, my version: 7.1.4
I am at least PHP version 5.3.0, my version: 7.1.4
```

| 내장 함수 |

```
string zend_version ( void )
```

내장 함수 `zend_version()`은 Zend 엔진의 버전을 가져옵니다.

예제 파일 | **zend_version.php**

```
1  <?php
2      echo "Zend engine version: " . zend_version();
3  ?>
```

화면 출력

Zend engine version: 3.1.0

| **내장 함수** |

```
int zend_thread_id ( void )
```

내장 함수 `zend_thread_id()`는 현재 스레드의 고유 ID를 반환합니다. 이 함수는 PHP가 ZTS(Zend Thread Safety) 지원 및 디버그 모드(`--enable-debug`)로 빌드된 경우에만 사용할 수 있습니다.

예제 파일 | **zend_thread_id.php**

```
1  <?php
2      $thread_id = zend_thread_id();
3      echo 'Current thread id is: ' . $thread_id;
4  ?>
```

| **내장 함수** |

```
string php_uname ([ string $mode = "a" ] )
```

내장 함수 `php_uname()`은 PHP가 실행되는 운영체제 정보를 반환합니다.

예제 파일 | **php_uname.php**

```
1  <?php
2      echo php_uname() . "<br>";
3      echo PHP_OS;
4  ?>
```

화면 출력

Windows NT LAPTOP-M0820HEF 10.0 build 15063 (Windows 10) i586
WINNT

19.2 정보 함수

PHP 환경 설정과 정보들을 내장 함수를 통하여 읽어올 수 있습니다.

| 내장 함수 |

```
bool phpinfo ([ int $what = INFO_ALL ] )
```

내장 함수 `phpinfo()`는 PHP의 설정에 관한 정보를 출력합니다. 전반적인 PHP의 설치 환경 정보를 웹 페이지를 통하여 출력할 수 있습니다. 또는 지정 모듈만 선택하여 출력할 수도 있습니다.

- `INFO_GENERAL`
- `INFO_CREDITS`
- `INFO_CONFIGURATION`
- `INFO_MODULES`
- `INFO_ENVIRONMENT`
- `INFO_VARIABLES`
- `INFO_LICENSE`
- `INFO_ALL`

예제 파일 | `phpinfo.php`

```
1 <?php
2
3 // 전체 정보를 출력합니다.
4 phpinfo();
5
```

```

6      // 지정한 모듈 정보만 출력합니다.
7      phpinfo(INFO_MODULES);
8
9      ?>

```

| 내장 함수 |

```
string get_cfg_var ( string $option )
```

내장 함수 `get_cfg_var()`은 PHP의 설정 옵션값을 확인할 수 있습니다.

`get_cfg_var()` 함수는 컴파일하여 PHP를 설치했을 때 또는 아파치 등의 웹 서버에서 정보를 읽어 참조했을 때는 값을 반환하지 않습니다.

예제 파일 | `get_cfg_var.php`

```

1  <?php
2      // php.ini 값을 확인합니다.
3      echo "SMTP Server = ". get_cfg_var("SMTP") . "<br>";
4      echo "SMTP Server = ". get_cfg_var("smtp_port") . "<br>";
5
6      ?>

```

화면 출력

```

SMTP Server = localhost
SMTP Server = 25

```

| 내장 함수 |

```
string getenv ( string $varname )
```

내장 함수 `getenv()`는 환경 변수 값을 읽어올 수 있습니다.

예제 파일 | **cfg-07.php**

```
1 <?php
2     $ip = getenv('REMOTE_ADDR');
3     echo "접속 주소 : $ip <br>";
4
5     // 글로벌 환경 변수 값을 통하여도 읽어들 수 있습니다.
6     $ipAddr = $_SERVER['REMOTE_ADDR'];
7     echo "접속 주소 : $ipAddr <br>";
8
9 ?>
```

| 내장 함수 |

bool putenv (string \$setting)

내장 함수 putenv()는 환경 변수 값을 설정합니다.

예제 파일 | **cfg-08.php**

```
1 <?php
2     print "env is: "._ENV["USER"]."<br>";
3
4     putenv("USER=jiny");
5     print "getenv is: ".getenv("USER")."<br>";
6
7 ?>
```

화면 출력

env is:
getenv is: jiny

19.3 ini

PHP의 환경 설정 파일 ini에 대한 정보를 읽거나 설정할 수 있습니다. 이와 관련된 몇 개

의 내장 함수들을 제공합니다.

| 내장 함수 |

```
string ini_get ( string $varname )
```

내장 함수 `ini_get()`은 구성 옵션의 값을 가져옵니다.

예제 파일 | `ini_get.php`

```
1  <?php
2      echo 'display_errors = ' . ini_get('display_errors') . "<br>";
3      echo 'register_globals = ' . ini_get('register_globals') . "<br>";
4      echo 'post_max_size = ' . ini_get('post_max_size') . "<br>";
5      echo 'post_max_size+1 = ' . (ini_get('post_max_size')+1) . "<br>";
6      echo 'post_max_size in bytes = ' . ini_get('post_max_size');
7
8  ?>
```

화면 출력

```
display_errors =
register_globals =
post_max_size = 8M
post_max_size+1 = 9
post_max_size in bytes = 8M
```

| 내장 함수 |

```
array ini_get_all ([ string $extension [, bool $details = true ] ] )
```

내장 함수 `ini_get_all()`은 모든 구성 옵션을 가져옵니다.

예제 파일 | `ini_get_all.php`

```
1  <?php
2      print_r(ini_get_all("pcre"));
```

```

3     print_r(ini_get_all());
4
5     ?>

```

| 내장 함수 |

```
string ini_set ( string $varname , string $newvalue )
```

내장 함수 `ini_set()`는 구성 옵션의 값을 설정합니다.

예제 파일 | `ini_set.php`

```

1  <?php
2      echo ini_get('display_errors');
3
4      if (!ini_get('display_errors')) {
5          ini_set('display_errors', '1');
6      }
7
8      echo ini_get('display_errors');
9
10  ?>

```

화면 출력

```
1
```

| 내장 함수 |

```
void ini_restore ( string $varname )
```

내장 함수 `ini_restore()`는 구성 옵션의 값을 복원합니다.

예제 파일 | `ini_restore.php`

```

1  <?php
2      $setting = 'y2k_compliance';

```



```

3
4     echo 'Current value for \'' . $setting . '\': ' . ini_get($setting),
      "<br>";
5
6     ini_set($setting, ini_get($setting) ? 0 : 1);
7     echo 'New value for \'' . $setting . '\': ' . ini_get($setting), "<br>";
8
9     ini_restore($setting);
10    echo 'Original value for \'' . $setting . '\': ' . ini_get($setting),
      "<br>";
11  ?>

```

화면 출력

Current value for 'y2k_compliance':
 New value for 'y2k_compliance':
 Original value for 'y2k_compliance':

| 내장 함수 |

string **php_ini_loaded_file** (void)

내장 함수 `php_ini_loaded_file()`은 `php.ini` 파일의 경로를 검색합니다.

예제 파일 | `php_ini_loaded_file.php`

```

1  <?php
2      $inipath = php_ini_loaded_file();
3
4      if ($inipath) {
5          echo 'Loaded php.ini: ' . $inipath;
6      } else {
7          echo 'A php.ini file is not loaded';
8      }
9      ?>

```

화면 출력

Loaded php.ini: C:\php-7.1.4-Win32-VC14-x86\php.ini

| 내장 함수 |

```
string php_ini_scanned_files ( void )
```

내장 함수 `php_ini_scanned_files()`는 ini 디렉터리에서 파싱된 .ini 파일 목록을 반환합니다.

19.4 프로세스

PHP가 동작되는 프로세스에 대한 정보와 처리를 할 수 있습니다. 이와 관련된 몇 가지 내장 함수들을 제공합니다.

| 내장 함수 |

```
string cli_get_process_title ( void )
```

내장 함수 `cli_get_process_title()`은 현재 프로세스 제목을 반환합니다.

예제 파일 | `cli_get_process_title.php`

```
1  <?php
2      echo "Process title: " . cli_get_process_title() . "\n";
3
4  ?>
```

화면 출력

Process title: 명령 프롬프트 - php -S localhost:8000

| 내장 함수 |

```
bool cli_set_process_title ( string $title )
```

내장 함수 `cli_set_process_title()`은 프로세스 제목을 설정합니다.

예제 파일 | `cli_set_process_title.php`

```
1  <?php
2      $title = "jiny PHP Script";
3
4      cli_set_process_title($title);
5      echo "Process title: " . cli_get_process_title() . "\n";
6
7  ?>
```

화면 출력

Process title: jiny PHP Script

| 내장 함수 |

`int getmypid (void)`

내장 함수 `getmypid()`는 현재 PHP의 프로세스 ID를 확인할 수 있습니다.

예제 파일 | `cfg-05.php`

```
1  <?php
2      echo getmypid();
3
4  ?>
```

화면 출력

3868

| 내장 함수 |

`int getmyuid (void)`

내장 함수 `getmyuid()`는 현재 PHP의 UID를 확인할 수 있습니다.

예제 파일 | [cfg-06.php](#)

```
1 <?php
2     echo getmyuid();
3
4 ?>
```

| 내장 함수 |

int **getmyinode** (void)

내장 함수 `getmyinode()`는 현재 스크립트의 inode 값을 구합니다. inode는 유닉스 계통에서 사용하는 자료구조입니다. 각각의 파일은 1개의 inode 값을 갖고 있습니다.

예제 파일 | [getmyinode.php](#)

```
1 <?php
2     echo getmyinode();
3
4 ?>
```

| 내장 함수 |

string **uniqid** ([string \$prefix = "" [, bool \$more_entropy = false]])

내장 함수 `uniqid()`는 고유 ID를 생성합니다. 마이크로 초 단위로 현재 시각에 따라 접두사를 포함한 식별자를 생성합니다.

예제 파일 | [uniqid.php](#)

```
1 <?php
2     // 고유 ID 값을 생성합니다.
3     printf("uniqid(): %s <br>", uniqid());
4
```

```

5 // 접두사를 붙여서 고유 ID 값을 생성합니다.
6 $prefix = "php_";
7 printf("uniqid('$prefix'): %s <br>", uniqid($prefix));
8
9 // more_entropy
10 printf("uniqid('', true): %s <br>", uniqid('', true));
11
12 ?>

```

화면 출력

```

uniqid(): 599934c16f714
uniqid('php_'): php_599934c16f71a
uniqid('', true): 599934c16f71b2.39586057

```

19.5 리소스

PHP 동작 상태에 대한 메모리 등 리소스 등을 확인할 수 있습니다. 이와 관련된 몇 가지 내장 함수를 제공합니다.

| 내장 함수 |

```
array get_resources ([ string $type ] )
```

내장 함수 `get_resources()`는 활성화된 리소스를 반환합니다.

예제 파일 | `get_resources.php`

```

1 <?php
2 $fp = tmpfile();
3 var_dump(get_resources());
4
5 ?>

```

화면 출력

```
array(1) { [2]=> resource(2) of type (stream) }
```

| 내장 함수 |

```
array getrusage ([ int $who = 0 ] )
```

내장 함수 `getrusage()`는 현재 리소스 사용량을 가지고 옵니다.

예제 파일 | `getrusage.php`

```
1 <?php
2     $dat = getrusage();
3     print_r($dat);
4
5 ?>
```

화면 출력

```
Array ( [ru_majflt] => 3686 [ru_maxrss] => 12852 [ru_utime.tv_usec] => 31250
[ru_utime.tv_sec] => 0 [ru_stime.tv_usec] => 78125 [ru_stime.tv_sec] => 0 )
```

| 내장 함수 |

```
int memory_get_usage ([ bool $real_usage = false ] )
```

내장 함수 `memory_get_usage()`는 PHP에 할당된 메모리 양을 반환합니다.

예제 파일 | `memory_get_usage.php`

```
1 <?php
2     // 시작 전 메모리
3     echo memory_get_usage() . "<br>";
4
5     // 메모리 할당
6     $a = str_repeat("Hello", 4242);
7     echo memory_get_usage() . "<br>";
8
9     // 메모리 해제
10    unset($a);
11    echo memory_get_usage() . "<br>";
```

```
12
13  ?>
```

화면 출력

```
344128
368704
344128
```

| 내장 함수 |

```
int memory_get_peak_usage ([ bool $real_usage = false ] )
```

내장 함수 `memory_get_peak_usage()`는 PHP가 할당한 메모리 피크치를 반환합니다.

19.6 include

내장 함수 `get_included_files()`는 `include` 또는 `require` 파일의 이름을 가진 배열을 반환합니다.

| 내장 함수 |

```
array get_included_files ( void )
```

예제 파일 | [get_included_files.php](#)

```
1  <?php
2
3      include 'test1.php';
4      include_once 'test2.php';
5      require 'test3.php';
6      require_once 'test4.php';
7
8      $included_files = get_included_files();
```

```

9
10     foreach ($included_files as $filename) {
11         echo "$filename\n";
12     }
13
14     ?>

```

| 내장 함수 |

string **set_include_path** (string \$new_include_path)

내장 함수 `set_include_path()`는 `include_path` 구성 옵션을 설정합니다.

예제 파일 | **set_include_path.php**

```

1  <?php
2      set_include_path(' /usr/lib/pear');
3
4      // 또는 ini_set()를 사용할 수도 있습니다.
5      ini_set('include_path', ' /usr/lib/pear');
6
7      ?>

```

| 내장 함수 |

void **restore_include_path** (void)

내장 함수 `restore_include_path()`는 `include_path` 구성 옵션의 값을 복원합니다.

예제 파일 | **restore_include_path.php**

```

1  <?php
2
3      echo get_include_path();
4      set_include_path('/inc');
5

```



```

6     echo get_include_path();
7     restore_include_path();
8
9     ?>

```

| 내장 함수 |

```
string get_include_path ( void )
```

내장 함수 `get_include_path()`는 현재 `include_path` 구성 옵션을 가져옵니다.

예제 파일 | [get_include_path.php](#)

```

1  <?php
2      echo get_include_path();
3      echo ini_get('include_path');
4
5  ?>

```

19.7 extention

내장함수 `extension_loaded()`를 확장 프로그램이 로드되었는지 확인합니다. 콘솔상에서 `php -m` 옵션을 통하여 설치된 모든 확장 모듈을 확인할 수 있습니다.

| 내장 함수 |

```
bool extension_loaded ( string $name )
```

예제 파일 | [extension_loaded.php](#)

```

1  <?php
2      if (extension_loaded('curl')) {
3          echo "모듈 설치됨";
4      } else {

```

```

5      echo "모듈이 설치되지 않았습니다.";
6    }
7
8    ?>

```

화면 출력

모듈 설치됨

| 내장 함수 |

array **get_loaded_extensions** ([bool \$zend_extensions = false])

내장 함수 `get_loaded_extensions()`를 컴파일된 모든 모듈명을 배열로 반환합니다.

예제 파일 | `get_loaded_extension.php`

```

1  <?php
2      print_r(get_loaded_extensions());
3
4  ?>

```

화면 출력

```

Array ( [0] => Core [1] => bcmath [2] => calendar [3] => ctype [4] => date [5]
=> filter [6] => hash [7] => iconv [8] => json [9] => mcrypt [10] => SPL [11]
=> pcre [12] => readline [13] => Reflection [14] => session [15] => standard
[16] => mysqlnd [17] => tokenizer [18] => zip [19] => zlib [20] => libxml [21]
=> dom [22] => PDO [23] => Phar [24] => SimpleXML [25] => xml [26] => wddx [27]
=> xmlreader [28] => xmlwriter [29] => cli_server [30] => curl )

```

| 내장 함수 |

array **get_extension_funcs** (string \$module_name)

내장 함수 `get_extension_funcs()`는 모듈의 함수 이름을 가진 배열을 반환합니다.

예제 파일 | `get_extension_funcs.php`

```
1  <?php
2      print_r(get_extension_funcs("xml"));
3
4  ?>
```

19.8 그 외

| 내장 함수 |

`string php_sapi_name (void)`

내장 함수 `php_sapi_name()`은 웹 서버와 PHP 사이의 인터페이스 유형을 반환합니다.

예제 파일 | `php_sapi_name.php`

```
1  <?php
2      $sapi_type = php_sapi_name();
3      echo $sapi_type;
4  ?>
```

화면 출력

cli-server

| 내장 함수 |

`array getopt (string $options [, array $longopts [, int &$optind]])`

내장 함수 `getopt()`는 콘솔상에서 명령행 인수 목록 옵션을 가지고 옵니다. 콘솔 모드를 이용한 응용프로그램을 개발할 때 인수 분석용으로 매우 유용합니다.

예제 파일 | **getopt.php**

```
1  <?php
2      $options = getopt("f:hp:");
3      var_dump($options);
4
5  ?>
```

화면 출력

```
C:\php-7.1.4-Win32-VC14-x86>php getopt.php -fvalue -h
array(2) {
    ["f"]=>
        string(5) "value"
    ["h"]=>
        bool(false)
}
```

| 내장 함수 |

```
array get_defined_constants ([ bool $categorize = false ] )
```

내장 함수 `get_defined_constants()`는 정의된 모든 상수를 이름과 값을 가진 연관 배열 형태로 반환합니다.

예제 파일 | **get_defined_constants.php**

```
1  <?php
2      print_r(get_defined_constants());
3  ?>
```

| 내장 함수 |

```
bool get_magic_quotes_gpc ( void )
```

내장 함수 `get_magic_quotes_gpc()`는 자동 따옴표 기능 활성화 상태를 체크할 수 있습니다.

예제 파일 | [cfg-10.php](#)

```
1  <?php
2      if (get_magic_quotes_gpc()) {
3          echo "자동 따옴표 기능 활성화";
4      } else {
5          echo "자동 따옴표 기능 비활성화";
6      }
7
8  ?>
```

화면 출력

자동 따옴표 기능 비활성화

| 내장 함수 |

```
bool phpcredits ([ int $flag = CREDITS_ALL ] )
```

내장 함수 `phpcredits()`는 PHP에 대한 크레딧을 출력합니다.

예제 파일 | [phpcredits.php](#)

```
1  <?php
2      phpcredits(CREDITS_GENERAL);
3
4  ?>
```

