Chapter

16

함수

PHP는 수많은 함수들을 포함하고 있습니다. 또한 다양한 함수들을 통하여 PHP 응용프로그램을 개발합니다.

함수들은 내부적으로 제공 함수와 사용자가 직접 작성하여 생성한 함수로 구분할 수 있습니다. 또한 함수 이름은 중복해서 사용할 수 없습니다. PHP는 내부적으로 함수를 관리하고 처리할 수 있는 별도의 특수 함수들을 제공합니다.

16.1 함수 목록

함수를 작성하거나 사용할 때 함수의 이름은 매우 중요합니다. 함수는 PHP 내에서 중복해서 사용할 수 없습니다. 만일 중복된 함수명을 생성하면 오류를 발생합니다.

또한 존재하지 않은 함수를 호출할 때도 PHP는 오류를 발생합니다. PHP는 지정한 이름의 함수가 존재하는지 확인할 수 있는 특별한 함수를 제공합니다.

| 내장 함수 |

bool **function_exists** (string $function_name)

내장 함수 function_exists()는 스크립트 내에 주어진 이름의 함수에 정의되어 있는지를 확인합니다.

예제 파일 | **function_exists.php**

```
<?php
   if (function_exists('str_replace')) {
      echo "함수가 정의되어 있습니다.<br />\n";
   } else {
      echo "없는 함수가 있습니다.<br />\n";
   }

?>
```

화면 출력

함수가 정의되어 있습니다.

| 내장 함수 |

array **get_defined_functions** ([bool $exclude_disabled = FALSE])

내장 함수 get_defined_functions()는 정의된 모든 함수의 목록을 배열로 반환합니다.

예제 파일 | **get_defined_functions.php**

```
<?php
   $arr = get_defined_functions();
   print_r($arr);

?>
```

16.2 함수 인자

우리가 함수를 이용하는 것은 코드의 중복되는 처리들을 블록화하는 것입니다. 함수는 처리를 위한 기본값의 매개변수와 반환값들을 가지고 있습니다.

PHP는 유연한 함수의 매개변수 처리를 지원합니다. 이와 관련하여 매개변수를 처리할 수 있는 몇 가지 함수들을 제공합니다.

| 내장 함수 |

```
int func_num_args ( void )
```

내장 함수 func_num_args()는 함수에 전달된 인수의 개수를 반환합니다.

예제 파일 | func_num_args.php

```
1  <?php
2     function foo()
3     {
4        $numargs = func_num_args();
5        echo "Number of arguments: $numargs\n";
6     }
7
8     foo(1, 2, 3);
9
10 ?>
```

화면 출력

```
Number of arguments: 3
```

위의 실험에서 사용자 정의 함수의 매개변수를 지정하지 않았습니다. 그리고 3개의 값을 전달하여 함수를 호출했습니다. func_num_args()를 통하여 전달되는 인자의 개수를 확인할 수 있습니다.

| 내장 함수 |

mixed **func_get_arg** (int $arg_num)

내장 함수 func_get_arg()는 인수 목록에서 항목 반환합니다.

예제 파일 | **func_get_arg.php**

```
1  <?php
2     function foo()
3     {
4        $numargs = func_num_args();
5        echo "Number of arguments: $numargs <br>";
6        if ($numargs >= 2) {
7           echo "Second argument is: " . func_get_arg(1) . " <br>";
8        }
9     }
10
11    foo(1, 2, 3);
12
13 ?>
```

화면 출력

```
Number of arguments: 3
Second argument is: 2
```

| 내장 함수 |

array **func_get_args** (void)

내장 함수 func_get_args()는 함수의 인수 목록을 구성하는 배열을 반환합니다.

예제 파일 | **func_get_args.php**

```
1  <?php
2     function foo()
3     {
```

```
4         $numargs = func_num_args();
5         echo "Number of arguments: $numargs <br>";
6         if ($numargs >= 2) {
7            echo "Second argument is: " . func_get_arg(1) . "<br>";
8         }
9
10        $arg_list = func_get_args();
11        for ($i = 0; $i < $numargs; $i++) {
12           echo "Argument $i is: " . $arg_list[$i] . "<br>";
13        }
14     }
15
16     foo(1, 2, 3);
17
18  ?>
```

화면 출력

```
Number of arguments: 3
Second argument is: 2
Argument 0 is: 1
Argument 1 is: 2
Argument 2 is: 3
```

16.3 콜백 호출

| 내장 함수 |

```
mixed call_user_func_array ( callable $callback , array $param_arr )
```

내장 함수 call_user_func_array()는 매개변수 배열을 사용하여 콜백 호출합니다.

예제 파일 | call_user_func_array.php

```
1  <?php
2     // 함수 호출
3     function foobar($arg, $arg2) {
```

```
4         echo __FUNCTION__, " got $arg and $arg2 <br>";
5      }
6
7      // 함수명
8      // 매개변수 배열
9      call_user_func_array("foobar", array("one", "two"));
10
11     // 객체 메서드 호출
12     class foo {
13        function bar($arg, $arg2) {
14           echo __METHOD__, " got $arg and $arg2<br>";
15        }
16     }
17
18     $foo = new foo;
19     // 인스턴스, 메서드 배열
20     // 매개변수 배열
21     call_user_func_array(array($foo, "bar"), array("three", "four"));
22
23  ?>
```

화면 출력

```
foobar got one and two
foo::bar got three and four
```

| 내장 함수 |

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

내장 함수 call_user_func()는 매개변수에 의해 콜백 호출 처리합니다.

예제 파일 | call_user_func.php

```
1  <?php
2     function barber($type)
3     {
4        echo "You wanted a $type haircut <br>";
5     }
```

```
6
7       call_user_func('barber', "mushroom");
8       call_user_func('barber', "shave");
9
10  ?>
```

화면 출력

```
You wanted a mushroom haircut
You wanted a shave haircut
```

16.4 메서드 호출

| 내장 함수 |

mixed **forward_static_call** (callable $function [, mixed $parameter [, mixed $...]])

내장 함수 forward_static_call()은 정적 메서드를 호출합니다.

예제 파일 | **forward_static_call .php**

```
1   <?php
2
3       class A
4       {
5           const NAME = 'A';
6           public static function test() {
7               $args = func_get_args();
8               echo static::NAME, " ".join(',', $args)." <br>";
9           }
10      }
11
12      class B extends A
13      {
14          const NAME = 'B';
15
16          public static function test() {
17              echo self::NAME, "<br>";
```

```
            forward_static_call(array('A', 'test'), 'more', 'args');
            forward_static_call( 'test', 'other', 'args');
        }
    }

    B::test('foo');

    function test() {
        $args = func_get_args();
        echo "C ".join(',', $args)." <br>";
    }

?>
```

화면 출력

```
B
B more,args
C other,args
```

| 내장 함수 |

mixed **forward_static_call_array** (callable $function , array $parameters)

내장 함수 forward_static_call_array()는 정적 메서드를 호출합니다. 인수는 배열로 전달합니다.

예제 파일 | **forward_static_call_array.php**

```
<?php

    class A
    {
        const NAME = 'A';
        public static function test() {
            $args = func_get_args();
            echo "Class == ".static::NAME, " ".join(',', $args)." <br>";
        }
    }
```

```
11
12      class B extends A
13      {
14          const NAME = 'B';
15
16          public static function test() {
17              echo "Class : ". self::NAME, "<br>";
18
19              // A클래스 test 메서드 호출
20              forward_static_call_array(array('A', 'test'), array('more',
                'args'));
21
22              // 함수 호출
23              forward_static_call_array( 'test', array('other', 'args'));
24          }
25      }
26
27      B::test('foo');
28
29      function test() {
30          $args = func_get_args();
31          echo "function call = ".join(',', $args)." <br>";
32      }
33
34  ?>
```

화면 출력

```
Class : B
Class == B more,args
function call = other,args
```

16.5 틱 실행

| 내장 함수 |

bool **register_tick_function** (callable $function [, mixed $arg [, mixed $...]])

내장 함수 register_tick_function()은 틱에서 실행될 함수를 등록합니다.

예제 파일 | register_tick_function.php

```php
<?php
   declare(ticks=1);

   register_tick_function('my_function', true);

   $object = new my_class();
   register_tick_function(array(&$object, 'my_method'), true);

?>
```

| 내장 함수 |

```
void unregister_tick_function ( string $function_name )
```

내장 함수 unregister_tick_function()은 틱에서 실행될 함수를 해제합니다.