PHP array_replace_recursive() 函数 PHP 命名空间(namespace)

PHP 魔术常量

PHP 向它运行的任何脚本提供了大量的预定义常量。

不过很多常量都是由不同的扩展库定义的，只有在加载了这些扩展库时才会出现，或者动态加载后，或者在编译时已经包括进去了。

有八个魔术常量它们的值随着它们在代码中的位置改变而改变。

例如 __LINE__ 的值就依赖于它在脚本中所处的行来决定。这些特殊的常量不区分大小写，如下：

__LINE__

文件中的当前行号。

实例

```
<?php
echo '这是第 " ' . __LINE__ . ' " 行';
?>
```

以上实例输出结果为：

```
这是第 “ 2 ” 行
```

__FILE__

文件的完整路径和文件名。如果用在被包含文件中，则返回被包含的文件名。

自 PHP 4.0.2 起，__FILE__ 总是包含一个绝对路径（如果是符号连接，则是解析后的绝对路径），而在此之前的版本有时会包含一个相对路径。

实例:

实例

```
<?php
echo '该文件位于 " ' . __FILE__ . ' " ';
?>
```

以上实例输出结果为：

```
该文件位于 “ E:\wamp\www\test\index.php ”
```

__DIR__

文件所在的目录。如果用在被包括文件中，则返回被包括的文件所在的目录。

它等价于 dirname(__FILE__)。除非是根目录，否则目录中名不包括末尾的斜杠。（PHP 5.3.0中新增）

实例

```
<?php
echo '该文件位于 " ' . __DIR__ . ' " ';
?>
```

以上实例输出结果为：

```
该文件位于 “ E:\wamp\www\test ”
```

__FUNCTION__

函数名称（PHP 4.3.0 新加）。自 PHP 5 起本常量返回该函数被定义时的名字（区分大小写）。在 PHP 4 中该值总是小写字母的。

实例

```
<?php
function test() {
echo '函数名为: ' . __FUNCTION__ ;
}
test();
?>
```

以上实例输出结果为：

```
函数名为: test
```

__CLASS__

类的名称（PHP 4.3.0 新加）。自 PHP 5 起本常量返回该类被定义时的名字（区分大小写）。

在 PHP 4 中该值总是小写字母的。类名包括其被声明的作用区域（例如 Foo\Bar）。注意自 PHP 5.4 起 __CLASS__ 对 trait 也起作用。当用在 trait 方法中时，__CLASS__ 是调用 trait 方法的类的名字。

实例

```
<?php
class test {
function _print() {
echo '类名为: ' . __CLASS__ . "<br>";
echo '函数名为: ' . __FUNCTION__ ;
}
}
$t = new test();
$t->_print();
?>
```

以上实例输出结果为：

```
类名为: test
函数名为: _print
```

__TRAIT__

Trait 的名字（PHP 5.4.0 新加）。自 PHP 5.4.0 起，PHP 实现了代码复用的一个方法，称为 traits。

Trait 名包括其被声明的作用区域（例如 Foo\Bar）。

从基类继承的成员被插入的 SayWorld Trait 中的 MyHelloWorld 方法所覆盖。其行为 MyHelloWorld 类中定义的方法一致。优先顺序是当前类中的方法会覆盖 trait 方法，而 trait 方法又覆盖了基类中的方法。

实例

```php
<?php
class Base {
public function sayHello() {
echo 'Hello ';
}
}
trait SayWorld {
public function sayHello() {
parent::sayHello();
echo 'World!';
}
}
class MyHelloWorld extends Base {
use SayWorld;
}
$o = new MyHelloWorld();
$o->sayHello();
?>
```

以上例程会输出：

```
Hello World!
```

__METHOD__

类的方法名（PHP 5.0.0 新加）。返回该方法被定义时的名字（区分大小写）。

实例:

实例

```php
<?php
function test() {
echo '函数名为: ' . __METHOD__ ;
}
test();
?>
```

以上实例输出结果为：

```
函数名为: test
```

__NAMESPACE__

当前命名空间的名称（区分大小写）。此常量是在编译时定义的（PHP 5.3.0 新增）。

实例:

实例

```php
<?php
namespace MyProject;
echo '命名空间为: "', __NAMESPACE__, '"'; // 输出 "MyProject"
?>
```

以上实例输出结果为：

```
命名空间为: "MyProject"
```

← PHP array_replace_recursive() 函数 PHP 命名空间(namespace) →

```
$o = new MyHelloWorld();
$o->sayHello();
$o->sayWorld();
$o->sayExclamationMark();
?>
```

最终输出：`Hello World!`

马鹿 1年前 (2018-03-13)

以下代码可以获取当前执行的 PHP 文件名：

```
<?php
echo substr(__FILE__,strlen(__DIR__)-strlen(__FILE__)+1);
?>
```

哦 5个月前 (11-01)