PHP FUNDAMENTAL
STEP 03

- PHP Array
- Indexed Array
- Associative Array
- Multidimensional Array
- Array Functions
- PHP String
- String Functions

PHP Arrays

PHP array is an ordered map (contains value on the basis of key). It is used to hold multiple values of similar type in a single variable.

Less Code

We don't need to define multiple variables.

Easy to traverse

By the help of single loop, we can traverse all the elements of an array.

Sorting

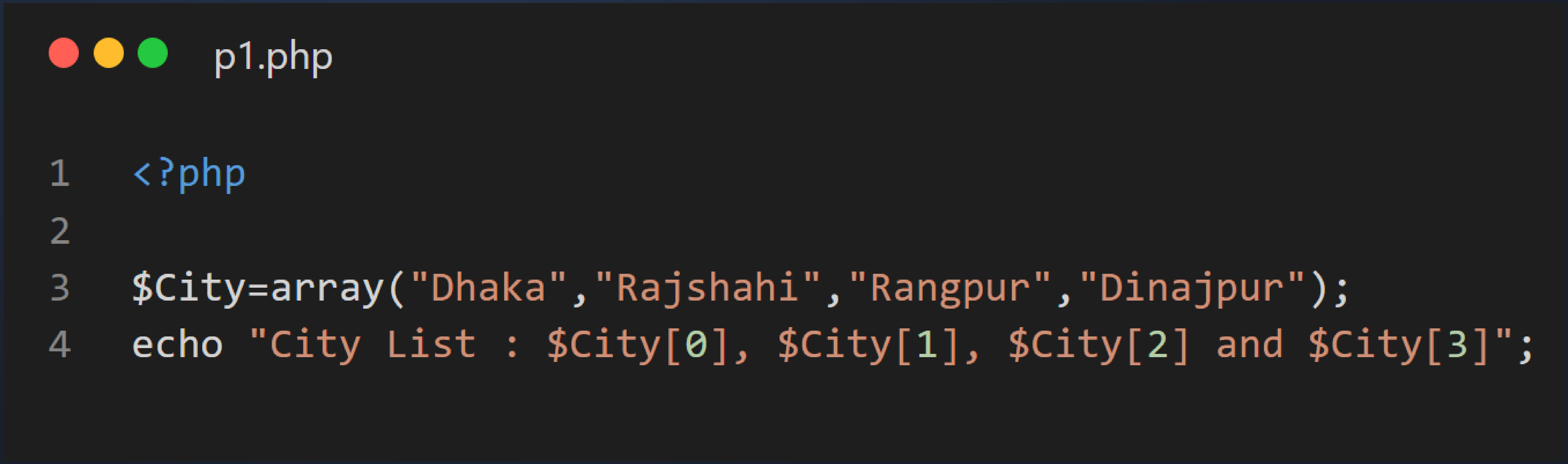
We can sort the elements of array.

Random Access

Access any values from any index

Indexed Array

PHP index is represented by number which starts from 0. We can store number, string and object in the PHP array. All PHP array elements are assigned to an index number by default.



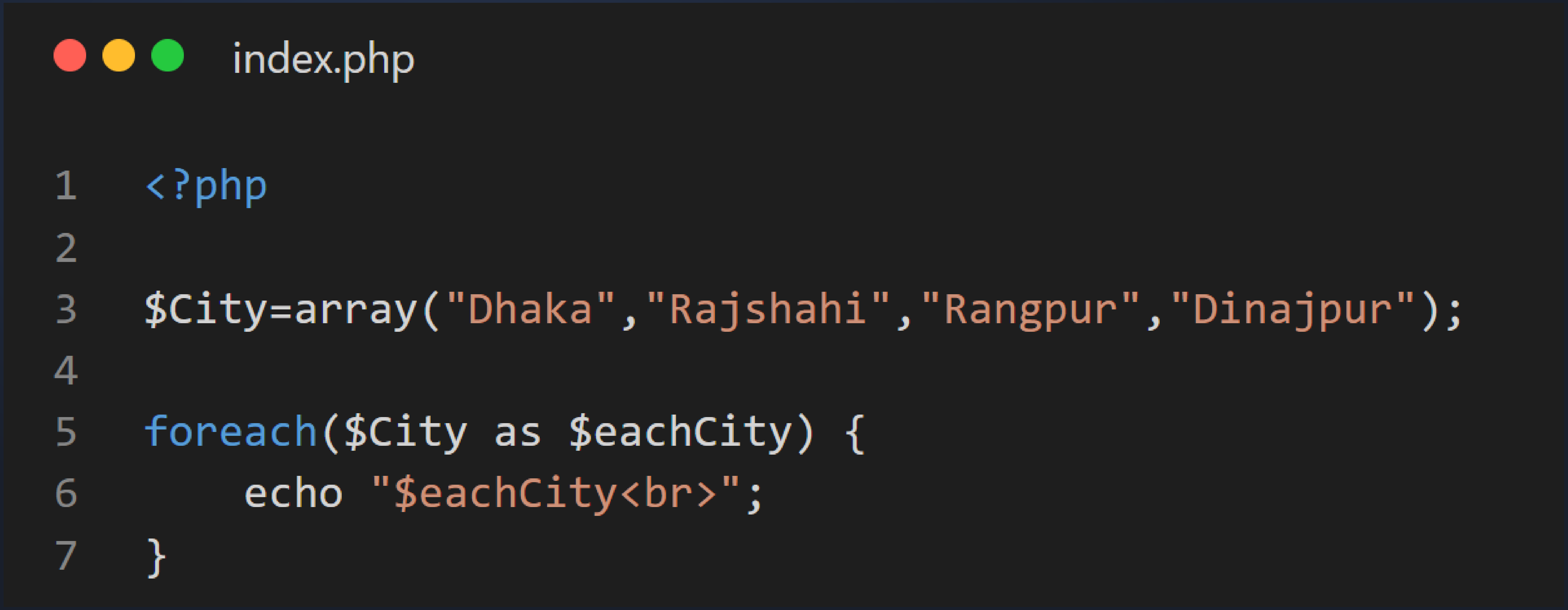
p1.php

```php
<?php

$City=array("Dhaka","Rajshahi","Rangpur","Dinajpur");
echo "City List : $City[0], $City[1], $City[2] and $City[3]";
```

For Each Loop Over

Indexed Array



index.php

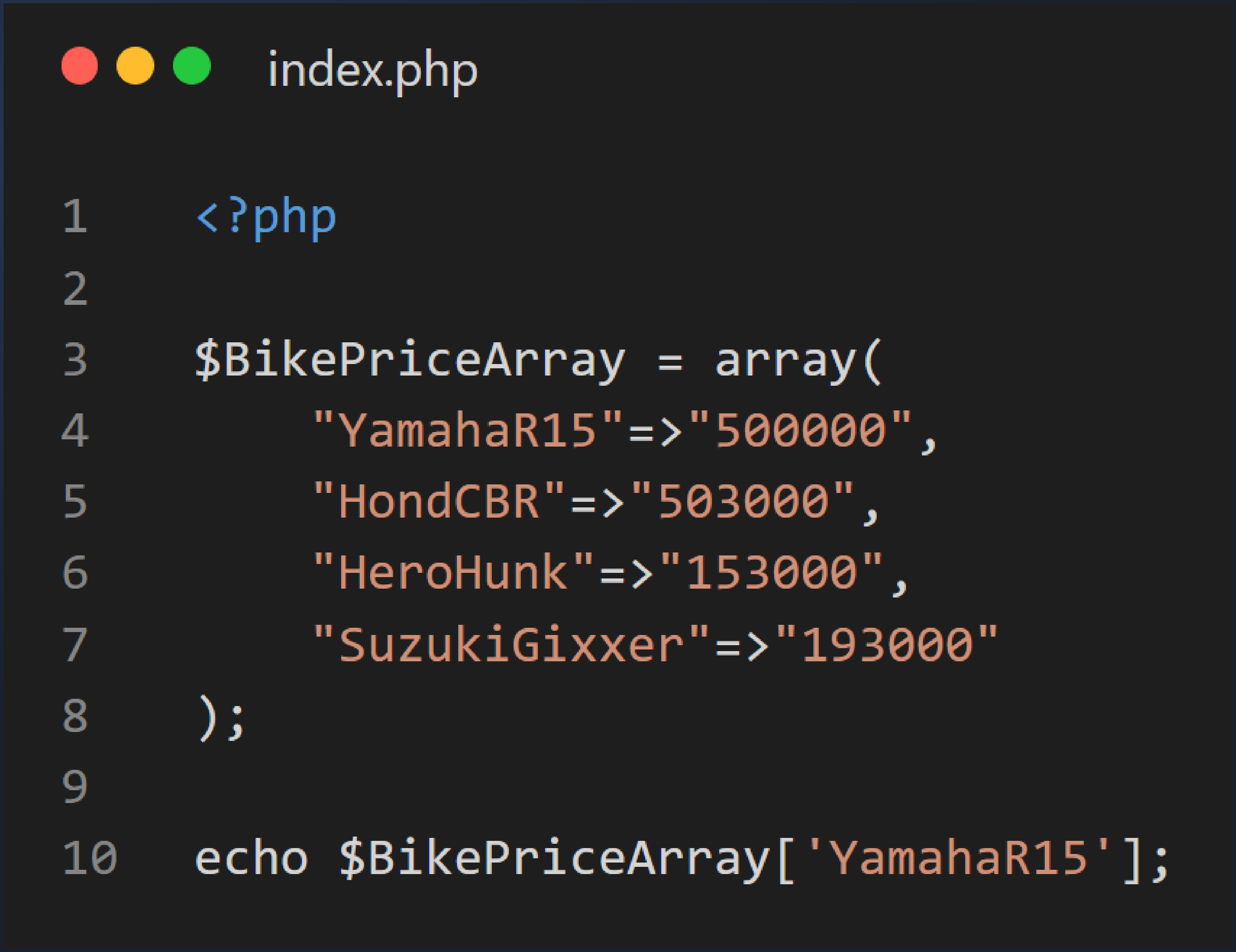
```php
<?php

$City=array("Dhaka","Rajshahi","Rangpur","Dinajpur");

foreach($City as $eachCity) {
    echo "$eachCity<br>";
}
```

PHP Associative Array

PHP allows you to associate name/label with each array elements in PHP using => symbol. Such way, you can easily remember the element because each element is represented by label than an incremented number.



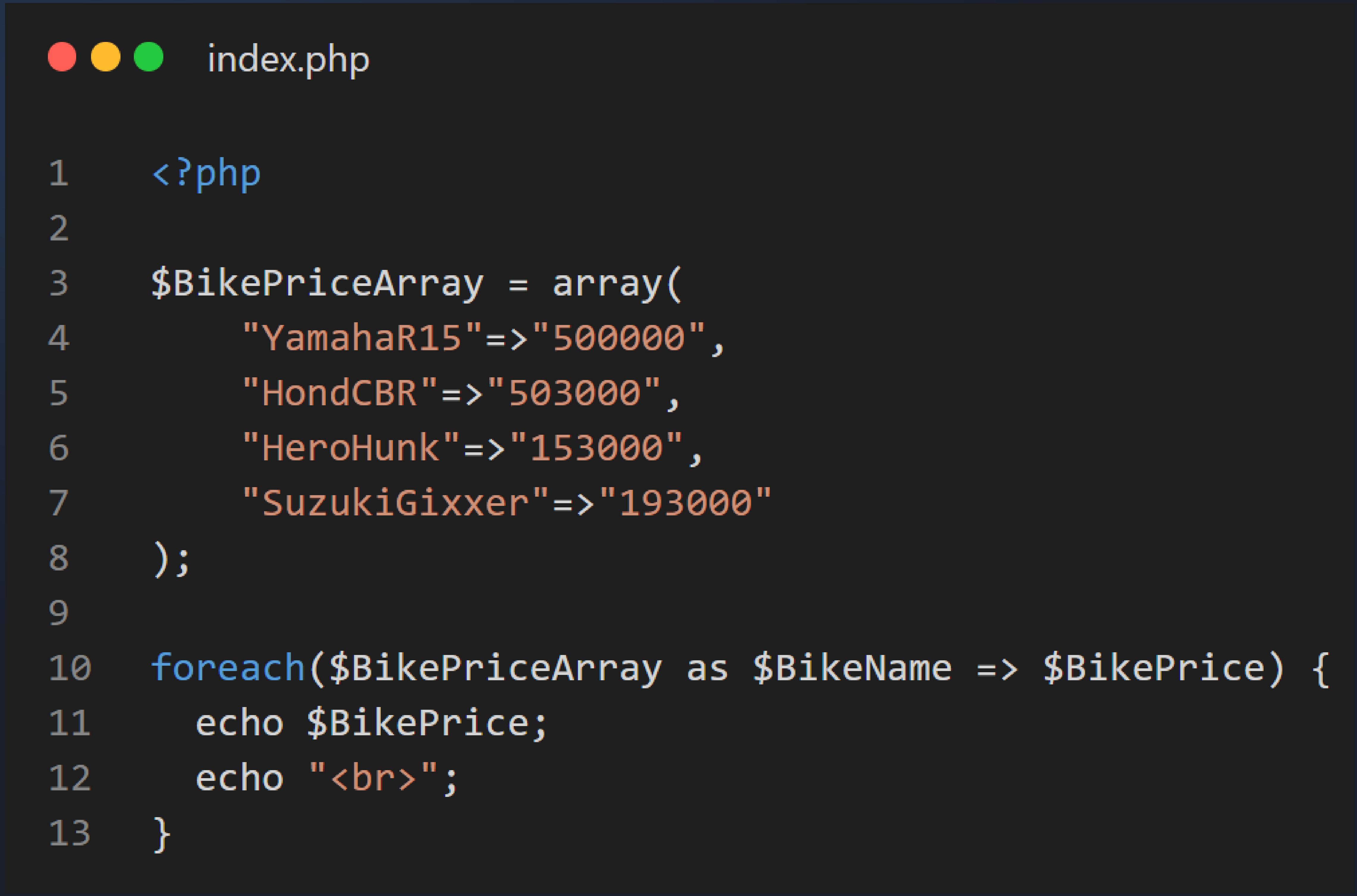
```php
<?php

$BikePriceArray = array(
    "YamahaR15"=>"500000",
    "HondCBR"=>"503000",
    "HeroHunk"=>"153000",
    "SuzukiGixxer"=>"193000"
);

echo $BikePriceArray['YamahaR15'];
```

For Each Loop Over

PHP Associative Array



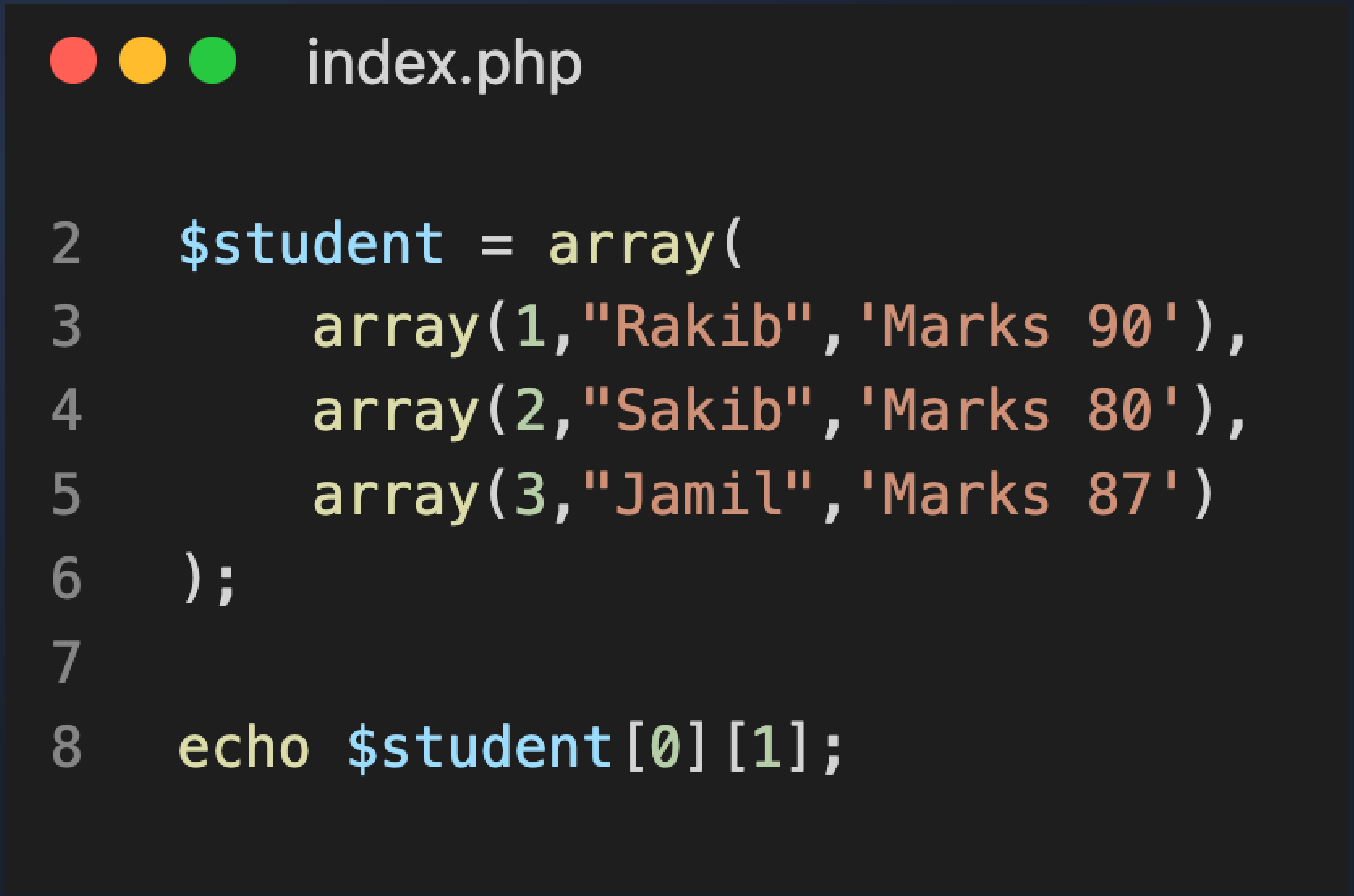
```php
<?php

$BikePriceArray = array(
    "YamahaR15"=>"500000",
    "HondCBR"=>"503000",
    "HeroHunk"=>"153000",
    "SuzukiGixxer"=>"193000"
);

foreach($BikePriceArray as $BikeName => $BikePrice) {
  echo $BikePrice;
  echo "<br>";
}
```

PHP Multidimensional Array

PHP multidimensional array is also known as array of arrays.



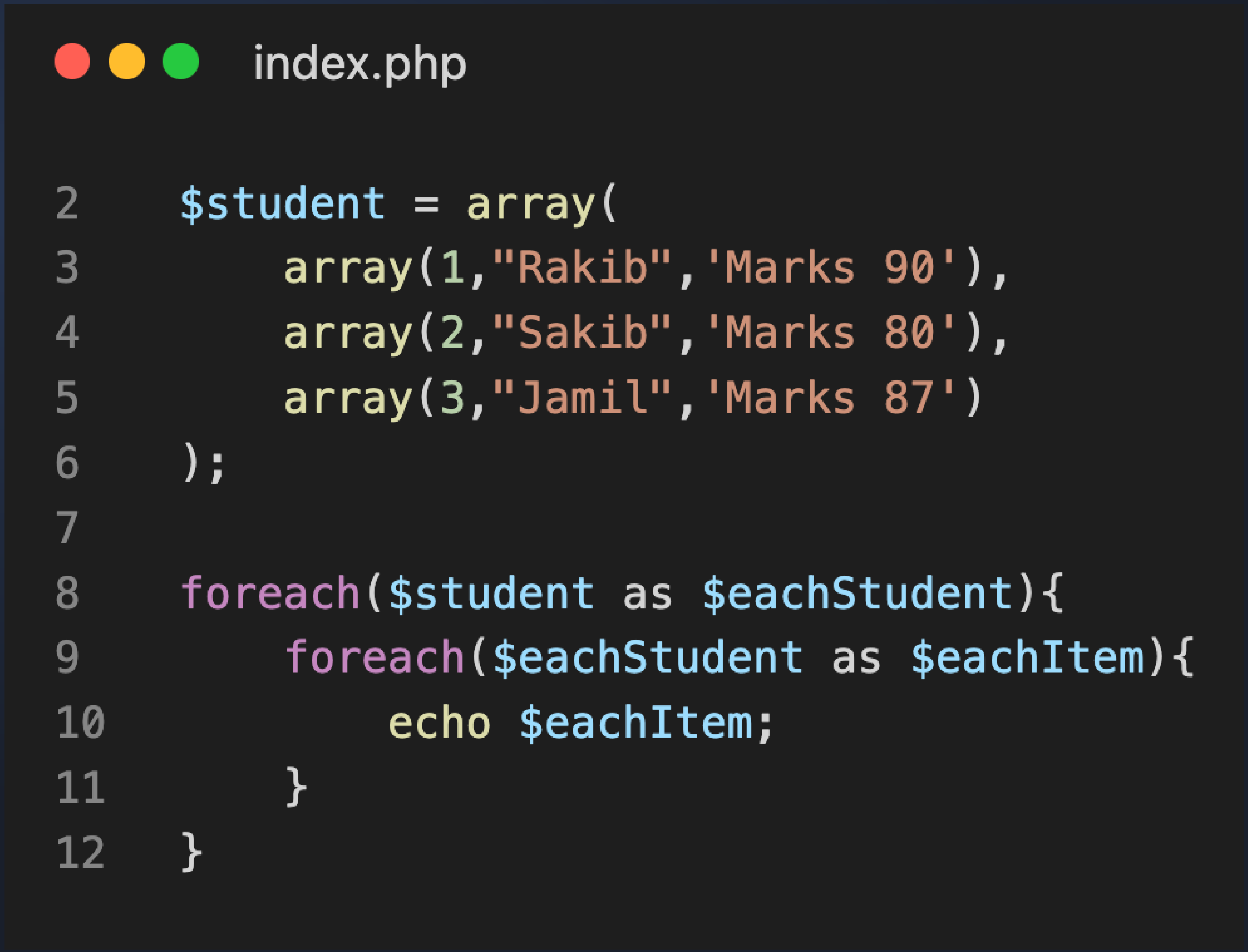
index.php

```php
$student = array(
        array(1,"Rakib",'Marks 90'),
        array(2,"Sakib",'Marks 80'),
        array(3,"Jamil",'Marks 87')
);

echo $student[0][1];
```

For Each Loop Over

PHP Multidimensional Array



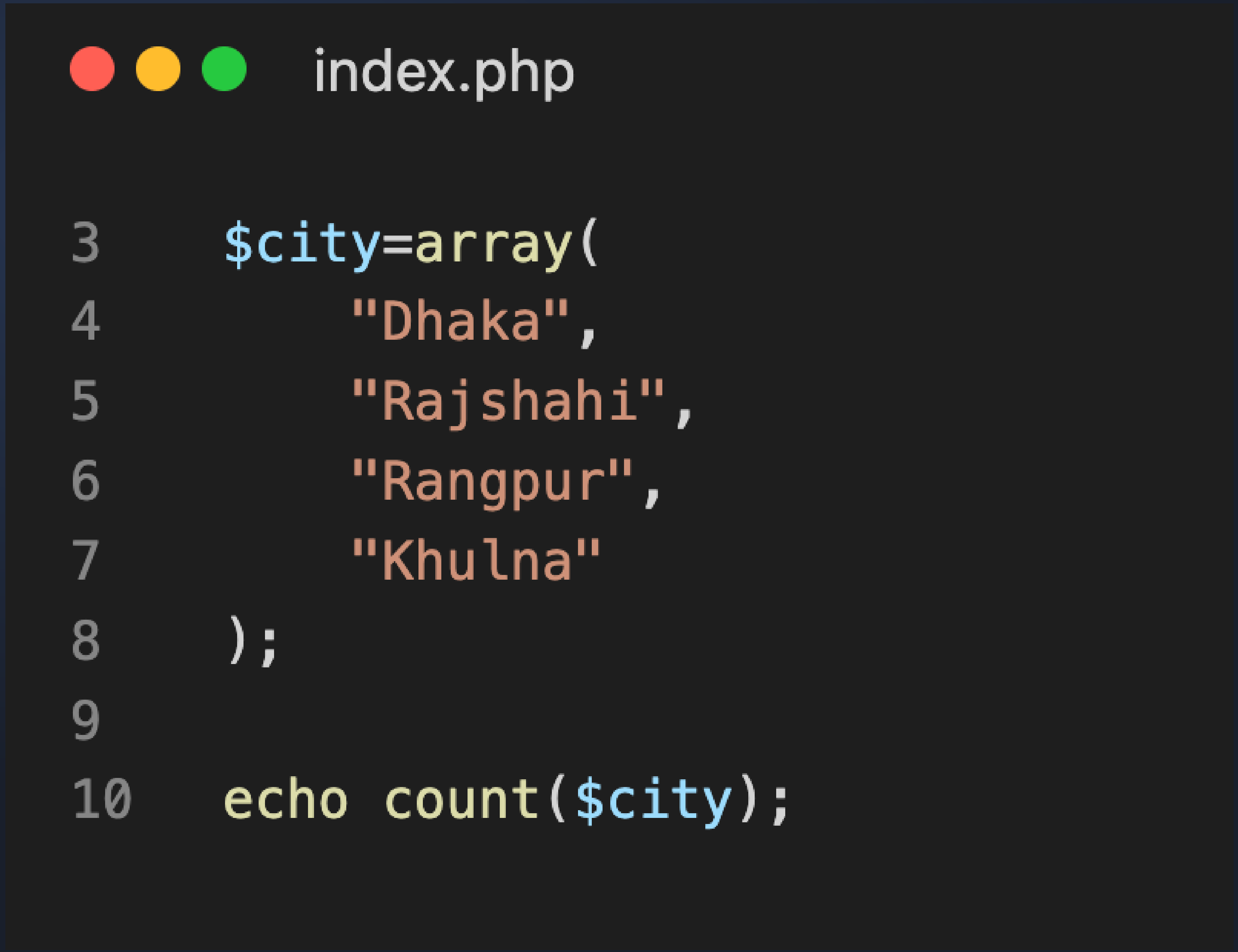
index.php

```php
$student = array(
    array(1,"Rakib",'Marks 90'),
    array(2,"Sakib",'Marks 80'),
    array(3,"Jamil",'Marks 87')
);

foreach($student as $eachStudent){
    foreach($eachStudent as $eachItem){
        echo $eachItem;
    }
}
```

count()

PHP count() function counts all elements in an array.

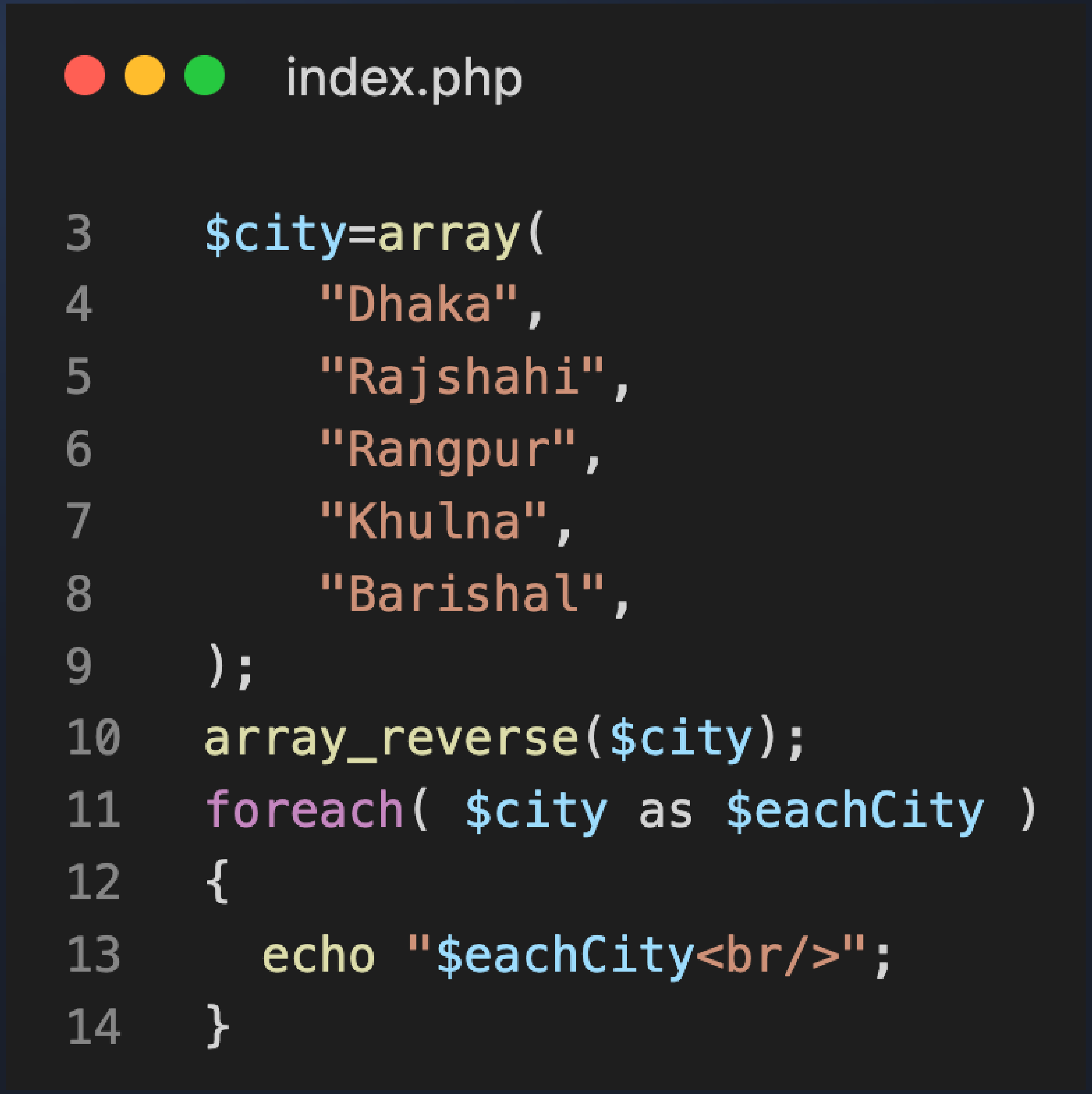


```php
$city=array(
    "Dhaka",
    "Rajshahi",
    "Rangpur",
    "Khulna"
);

echo count($city);
```

array_reverse()

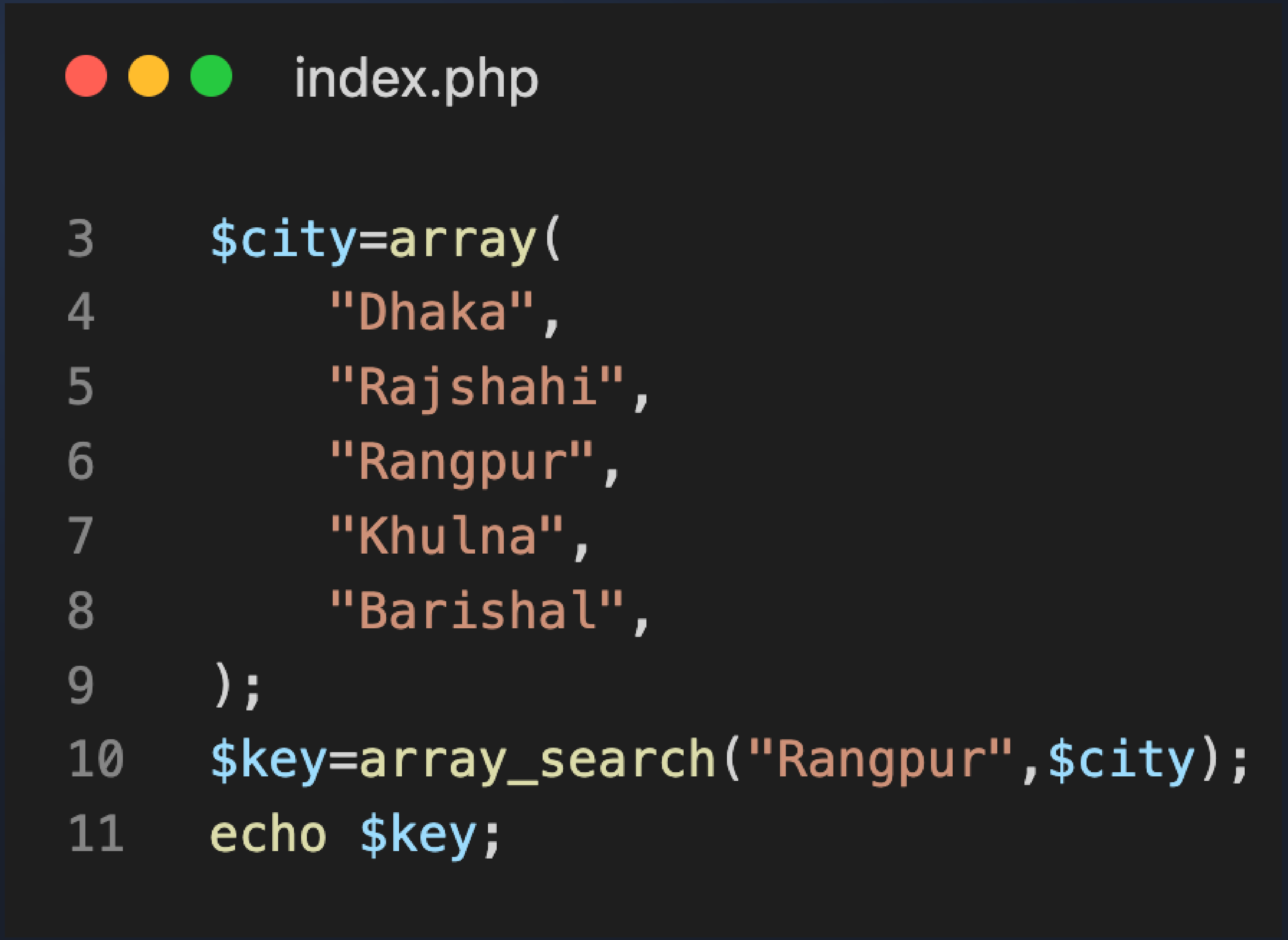
PHP array_reverse() function returns an array containing elements in reversed order.



```php
$city=array(
    "Dhaka",
    "Rajshahi",
    "Rangpur",
    "Khulna",
    "Barishal",
);
array_reverse($city);
foreach( $city as $eachCity )
{
  echo "$eachCity<br/>";
}
```

array_search()

PHP array_search() function searches the specified value in an array. It returns key if search is successful.



```php
$city=array(
    "Dhaka",
    "Rajshahi",
    "Rangpur",
    "Khulna",
    "Barishal",
);
$key=array_search("Rangpur",$city);
echo $key;
```

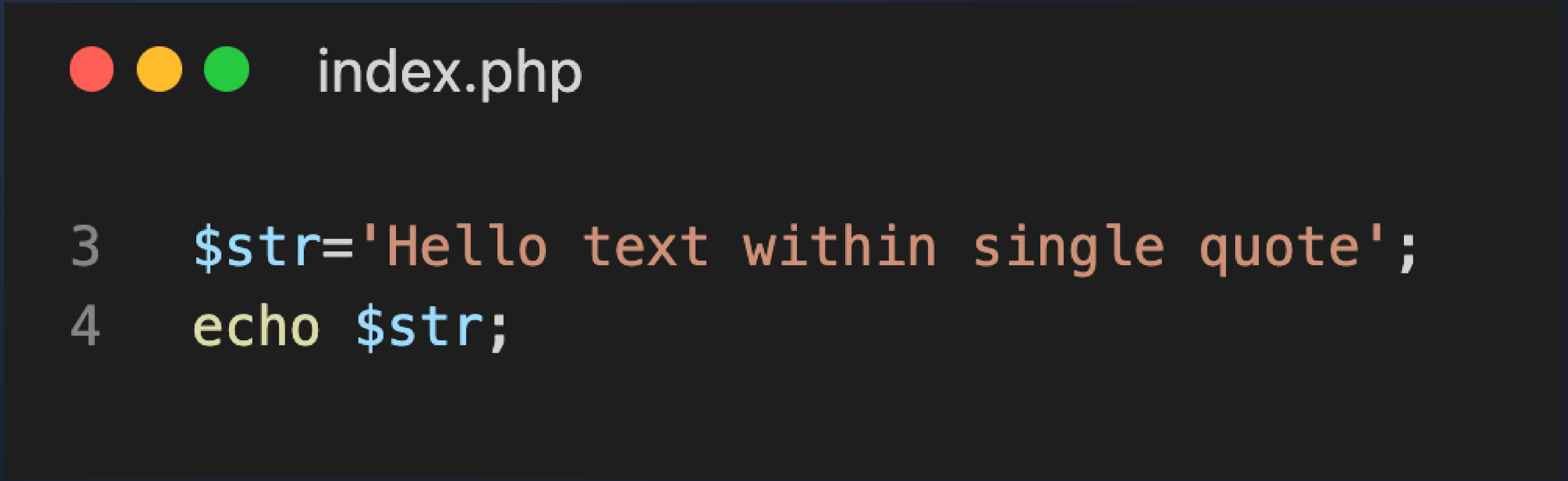
PHP STRING

PHP string is a sequence of characters i.e., used to store and manipulate text.

- Single quoted
- Double quoted
- Heredoc syntax
- Nowdoc syntax

Single Quoted

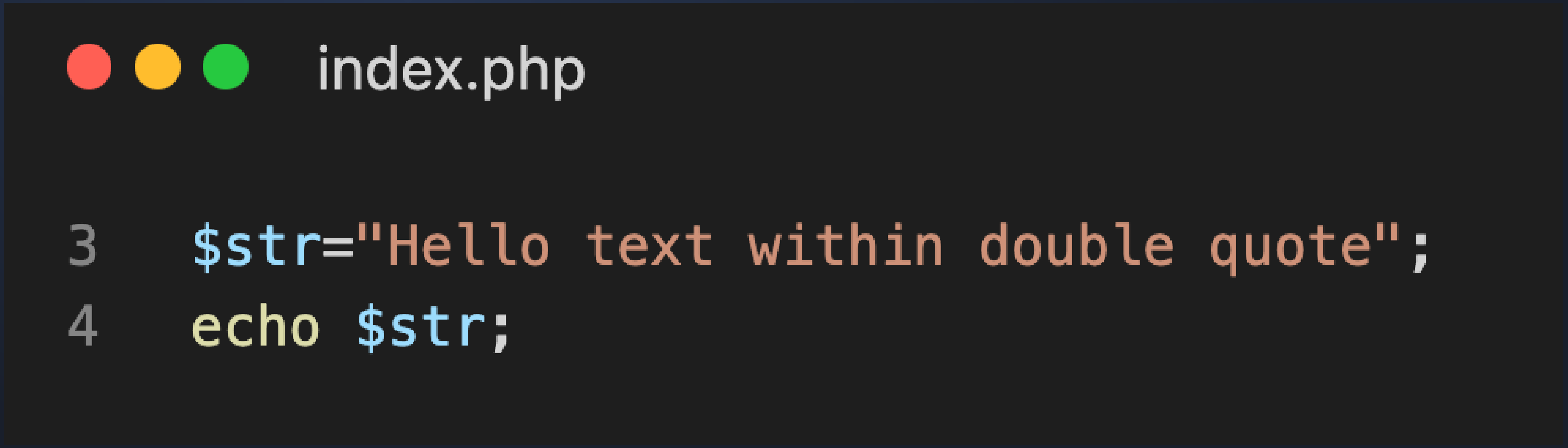
We can create a string in PHP by enclosing the text in a single-quote. It is the easiest way to specify string in PHP.



```php
$str='Hello text within single quote';
echo $str;
```

Double Quoted

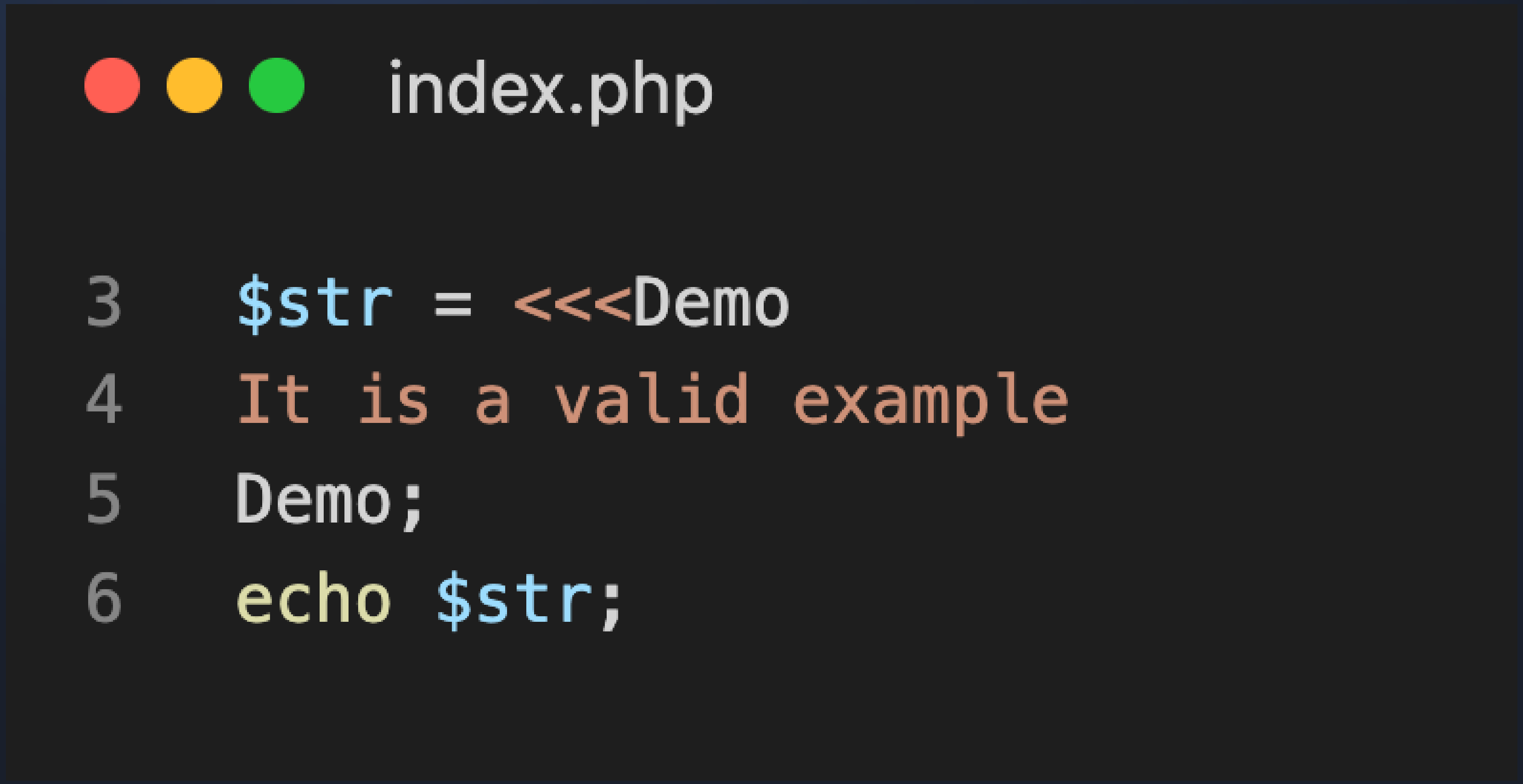
In PHP, we can specify string through enclosing text within double quote also.



```php
$str="Hello text within double quote";
echo $str;
```

Heredoc

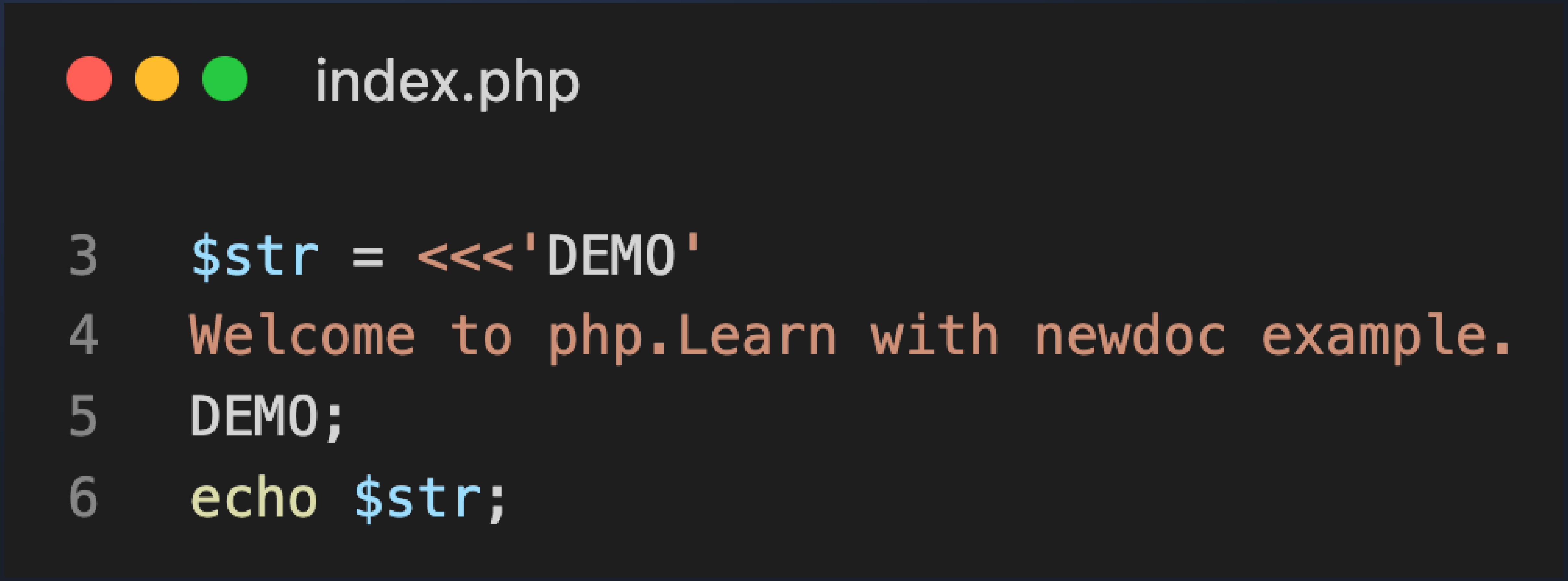
Heredoc syntax (<<<) is the third way to delimit strings. In Heredoc syntax, an identifier is provided after this heredoc <<< operator, and immediately a new line is started to write any text.



```php
$str = <<<Demo
It is a valid example
Demo;
echo $str;
```

Newdoc

Newdoc is similar to the heredoc, but in newdoc parsing is not done. It is also identified with three less than symbols <<< followed by an identifier. But here identifier is enclosed in single-quote, e.g. <<<'EXP'. Newdoc follows the same rule as heredocs. The difference between newdoc and heredoc is that - Newdoc is a single-quoted string whereas heredoc is a double-quoted string.

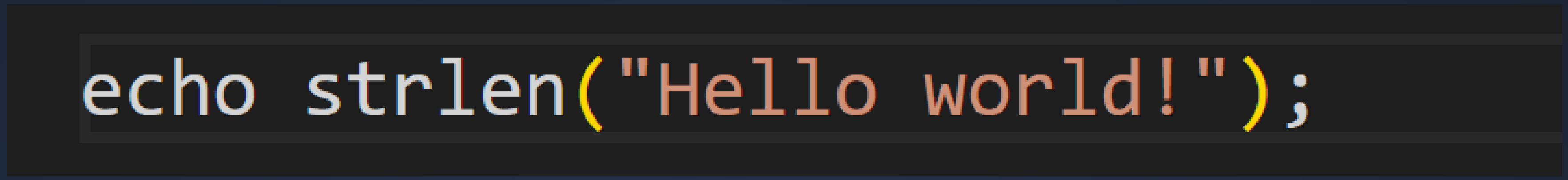


```php
$str = <<<'DEMO'
Welcome to php.Learn with newdoc example.
DEMO;
echo $str;
```

String Functions

strlen()

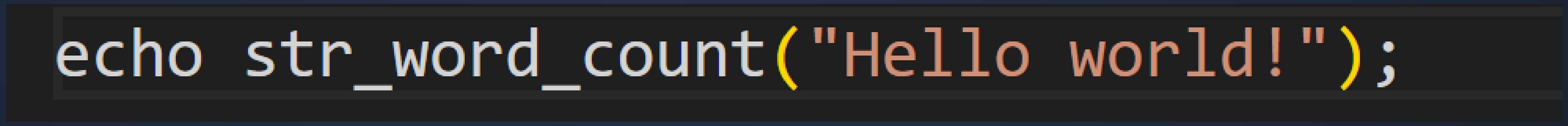
The PHP strlen() function returns the length of a string.



```
echo strlen("Hello world!");
```

str_word_count()

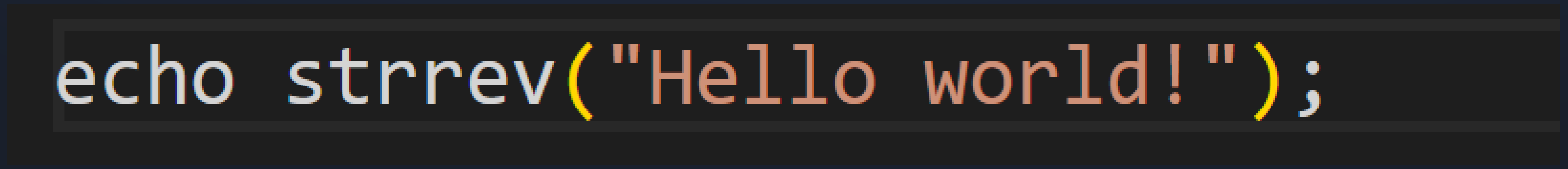
The PHP str_word_count() function counts the number of words in a string.



```
echo str_word_count("Hello world!");
```

strrev()

The PHP strrev() function reverses a string.

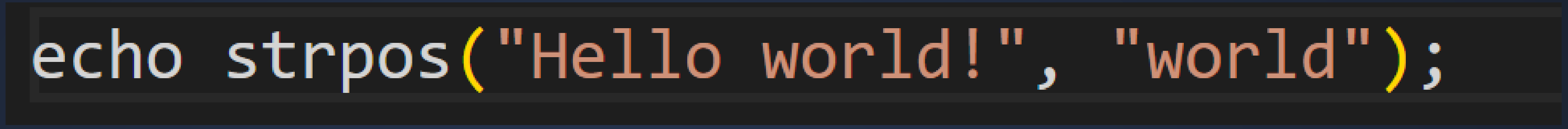


```
echo strrev("Hello world!");
```

String Functions

strpos()

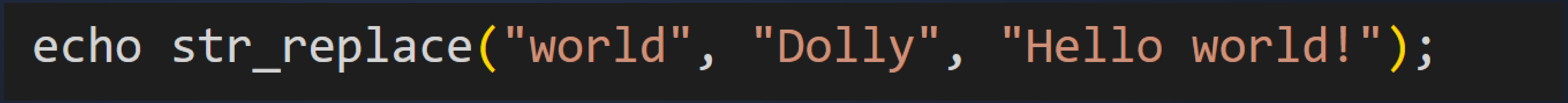
The PHP strpos() function searches for a specific text within a string. If a match is found, the function returns the character position of the first match. If no match is found, it will return FALSE.



```
echo strpos("Hello world!", "world");
```

str_replace()

The PHP str_replace() function replaces some characters with some other characters in a string.

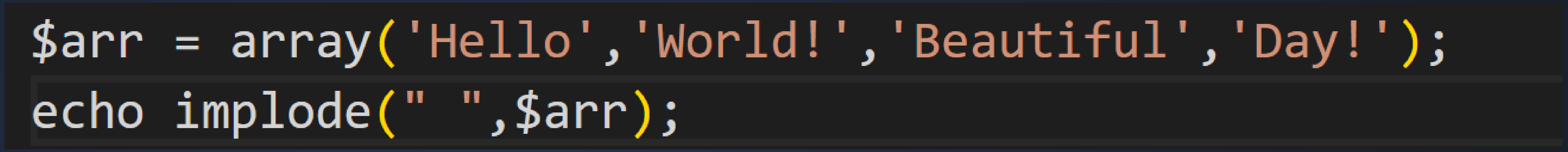


```
echo str_replace("world", "Dolly", "Hello world!");
```

String Functions

implode()

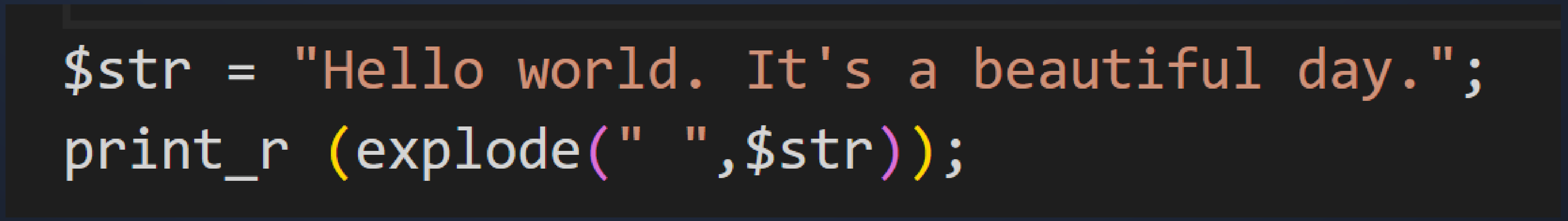
The implode() function returns a string from the elements of an array.



```
$arr = array('Hello','World!','Beautiful','Day!');
echo implode(" ",$arr);
```

explode()

he explode() function breaks a string into an array.

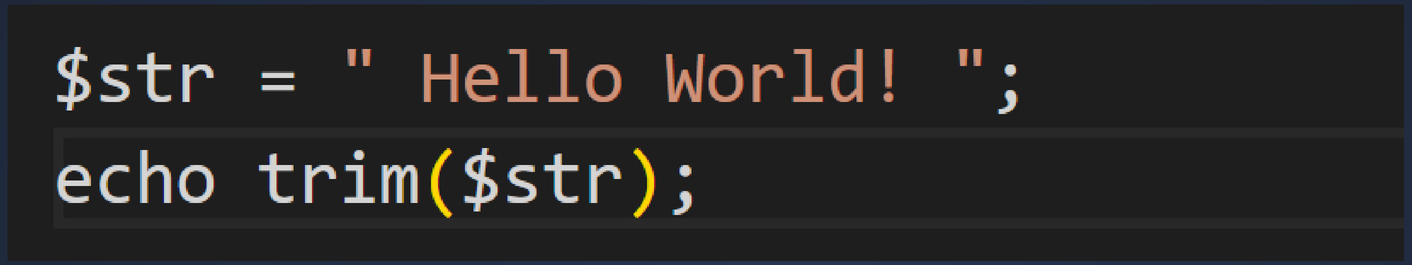


```
$str = "Hello world. It's a beautiful day.";
print_r (explode(" ",$str));
```

String Functions

trim()

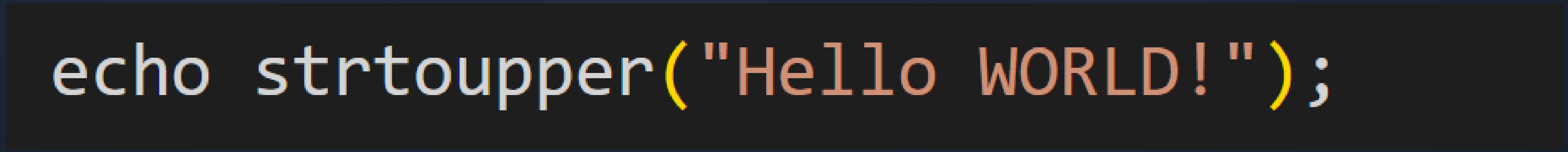
The trim() function removes whitespace and other predefined characters from both sides of a string.



```
$str = " Hello World! ";
echo trim($str);
```

strtoupper()

The strtoupper() function converts a string to uppercase.

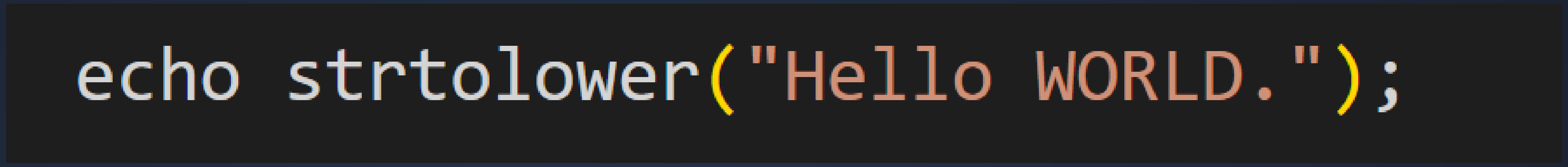


```
echo strtoupper("Hello WORLD!");
```

String Functions

strtolower()

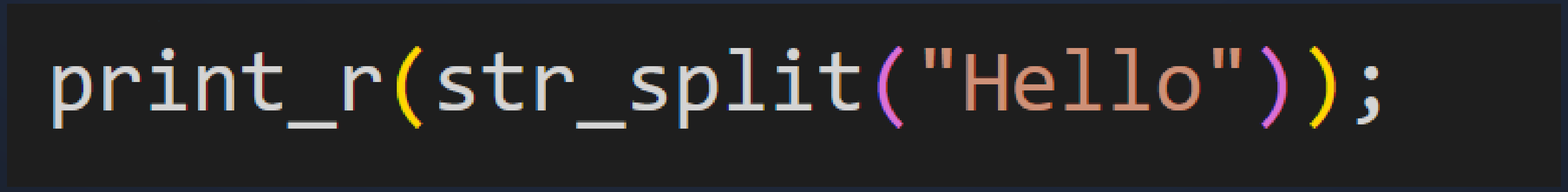
The strtolower() function converts a string to lowercase.



```
echo strtolower("Hello WORLD.");
```

str_split()

The str_split() function splits a string into an array.

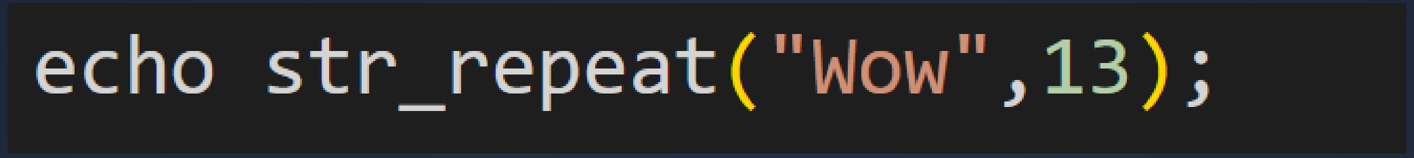


```
print_r(str_split("Hello"));
```

String Functions

str_repeat()

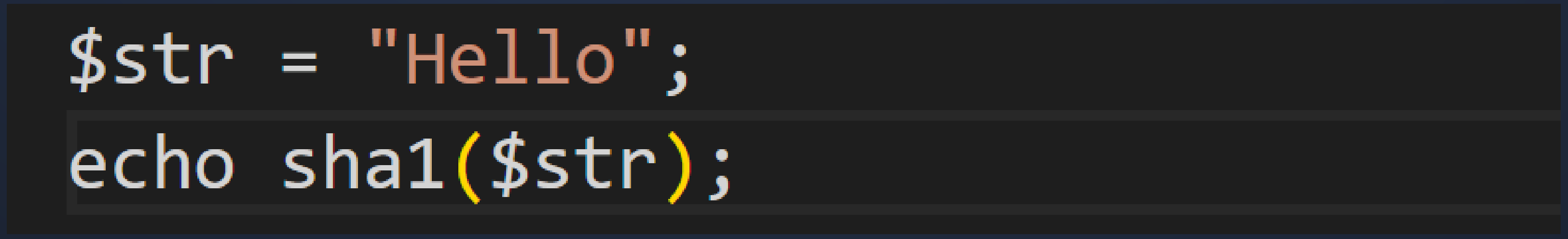
The str_repeat() function repeats a string a specified number of times.



```
echo str_repeat("Wow",13);
```

sha1()

The sha1() function calculates the SHA-1 hash of a string.

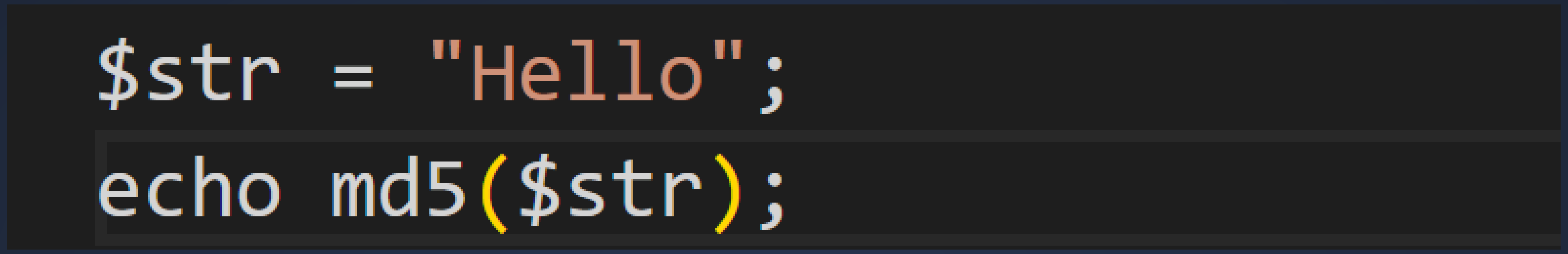


```
$str = "Hello";
echo sha1($str);
```

String Functions

md5()

he md5() function calculates the MD5 hash of a string.



```
$str = "Hello";
echo md5($str);
```