PHP - Variable Arguments

In PHP, it is possible to write a function capable of accepting a list of arguments with variable number of elements. To declare a variable argument list, the name of the argument is prepended by the "..." (three dots) symbol. The values passed are collected into an array with the argument's name.

```
function myfunction(...$arg) {
   Statement1;
   Statement2;
}
```

To call such a function, put any number of comma-separated values in the parenthesis.

```
myfunction(v1, v2, v3, . . . , vn);
```

The formal argument declared in the function is an array of all the values passed. We van use any of the appropriate built_in array functions to perform the process.

Example

In the following example, the user defined function **myfunction()** is capable of receiving variable number of values and finds their average.

Open Compiler

```
<?php
   function  myfunction(...$numbers) {
      $avg = array_sum($numbers)/count($numbers);
      return $avg;
   }
   $avg = myfunction(5, 12, 9, 23, 8);
   echo "average = $avg";
?>
```

It will produce the following **output** –

```
average = 11.4
```

Try changing the size of the passed array and run the program again.

You can use a **foreach** loop to traverse the array inside the function. The function may have any positional arguments before the variable length argument. From the received values, the positional arguments will be populated first, leaving others to be copied to the array.

Example

</> Open Compiler

```php
<?php
   function myfunction($x, ...$numbers) {
      echo "First number: $x" . PHP_EOL;
      echo "Remaining numbers: ";
      foreach ($numbers as $n) {
         echo "$n  ";
      }
   }
   myfunction(5, 12, 9, 23, 8, 41);
?>
```

It will produce the following **output** −

```
First number: 5
Remaining numbers: 12  9  23  8  41
```

Variadic Functions

It is possible to process a variable number of arguments to a function, even without the "..." syntax. PHP has built_in functions like func_num_args(), func_get_arg() and func_get_args(), which can be used with similar result.

- **func_num_args()** – Returns the number of arguments passed to the function.
- **func_get_arg()** – Returns an item from the argument list
- **func_get_args()** – Returns an array comprising a function's argument list

Example

The above example of variable arguments can be rewritten with these functions as below –

Open Compiler

```php
<?php
   function myfunction() {
      $sum = 0;
      foreach (func_get_args() as $n) {
         $sum += $n;
      }
      return $sum;
   }
   echo myfunction(5, 12, 9, 23, 8, 41);
?>
```

It will produce the following **output** –

```
98
```

Example

This program prints all the numbers passed to the function –

Open Compiler

```php
<?php
   function myfunction() {
      $len = func_num_args();
      echo "Numbers : ";
      $i=0;
      for ($i=0; $i<$len; $i++)
      echo func_get_arg($i) . " ";
   }
   myfunction(5, 12, 9, 23, 8, 41);
?>
```

It will produce the following **output** −

```
Numbers : 5 12 9 23 8 41
```