PHP – Closure::call()

In PHP, a **closure** is an anonymous function that has access to the variables in the scope in which it was created, even after that scope has closed. You need to specify use keyword in it.

Closures are objects that encapsulate the function code and the scope in which they were created. With PHP 7, a new **closure::call()** method was introduced to bind an object scope to a closure and invoke it.

Methods in the Closure Class

The Closure class has the following methods including the call() method −

```
final class Closure {

  /* Methods */
  private __construct()
  public static bind(Closure $closure, ?object $newThis, object|string|null $newS
  public bindTo(?object $newThis, object|string|null $newScope = "static"): ?Clos
  public call(object $newThis, mixed ...$args): mixed
  public static fromCallable(callable $callback): Closure

}
```

The **call() method** is a static method of Closure class. It has been introduced as a shortcut the bind() or bindTo() methods.

The **bind() method** Duplicates a closure with a specific bound object and class scope while the bindTo() method duplicates the closure with a new bound object and class scope.

The call() method has the following **signature** −

```
public Closure::call(object $newThis, mixed ...$args): mixed
```

The call() method temporarily binds the closure to newThis, and calls it with any given parameters.

With version prior to PHP 7, the bindTo() method can be used as follows −

```
</>                                                    Open Compiler
```

```php
<?php
   class A {
      private $x = 1;
   }

   // Define a closure Pre PHP 7 code
   $getValue = function() {
      return $this->x;
   };

   // Bind a clousure
   $value = $getValue->bindTo(new A, 'A');
   print($value());
?>
```

The program binds the **$getValue** which is a closure object, to the object of A class and prints the value of its private variable **$x** – it is 1.

With PHP 7, the binding is achieved by call() method as shown below −

</>   Open Compiler

```php
<?php
   class A {
      private $x = 1;
   }

   // PHP 7+ code, Define
   $value = function() {
      return $this->x;
   };

   print($value->call(new A));
?>
```