范围解析操作符（也可称作 Paamayim Nekudotayim）或者更简单地说是一对冒号，可以用于访问静态成员，类常量，还可以用于覆盖类中的属性和方法。

当在类定义之外引用到这些项目时，要使用类名。

自 PHP 5.3.0 起，可以通过变量来引用类，该变量的值不能是关键字（如 self，parent 和 static）。

把 Paamayim Nekudotayim 选作双冒号操作符的名字似乎有些奇怪。然而，这是 Zend 开发小组在写 Zend Engine 0.5（被用于 PHP 3 中）时所作出的决定。事实上这个词在希伯莱文就是双冒号的意思。

Example #1 在类的外部使用 :: 操作符

|  |
| --- |
|  |

|  |
| --- |
| <?php  class MyClass {      const CONST\_VALUE = 'A constant value';  }    $classname = 'MyClass';  echo $classname::CONST\_VALUE; // 自 PHP 5.3.0 起    echo MyClass::CONST\_VALUE;  ?> |

self，parent 和 static 这三个特殊的关键字是用于在类定义的内部对其属性或方法进行访问的。

Example #2 在类定义内部使用 ::

|  |
| --- |
|  |

|  |
| --- |
| <?php  class OtherClass extends MyClass  {      public static $my\_static = 'static var';        public static function doubleColon() {          echo parent::CONST\_VALUE . "\n";          echo self::$my\_static . "\n";      }  }  $classname = 'OtherClass';  echo $classname::doubleColon(); // 自 PHP 5.3.0 起    OtherClass::doubleColon();  ?> |

当一个子类覆盖其父类中的方法时，PHP 不会调用父类中已被覆盖的方法。是否调用父类的方法取决于子类。这种机制也作用于构造函数和析构函数，重载以及魔术方法。

Example #3 调用父类的方法

|  |
| --- |
|  |

<?php

class MyClass

{

    protected function myFunc() {

        echo "MyClass::myFunc()\n";

    }

}

class OtherClass extends MyClass

{

    // 覆盖了父类的定义

    public function myFunc()

    {

        // 但还是可以调用父类中被覆盖的方法

        parent::myFunc();

        echo "OtherClass::myFunc()\n";

    }

}

$class = new OtherClass();

$class->myFunc();

?>