



Chapter 5

PHP





PHP



PHP

PHP Variables Scope

In PHP, variables can be declared anywhere in the script.

The scope of a variable is the part of the script where the variable can be referenced/used.

PHP has three different variable scopes:

- Global
- Local
- Static

```
<!-- ... height="450" viewBox="0 0 450 450" ... -->
<defs>
  <linearGradient x1="100%" y1="0%" x2="0%" y2="100%" id="bg-gradient">
    <stop stop-color="#000000" offset="0%" />
    <stop stop-color="#101010" offset="100%" />
  </linearGradient>
</defs>
<rect width="800" height="450" rx="0" fill="url(#bg-gradient)" />
</svg>
<div class="media-control">
  <svg width="96" height="96" viewBox="0 0 96 96" id="media-control">
    <defs>
      <linearGradient x1="0%" y1="0%" x2="100%" y2="100%" id="media-control-gradient">
        <stop stop-color="white" stop-opacity="0.5" offset="0%" />
        <stop stop-color="white" stop-opacity="0.5" offset="100%" />
      </linearGradient>
      <filter x="-500%" y="-500%" width="1000%" height="1000%" id="media-control-blur">
        <feOffset dy="16" in="SourceGraphic" result="offset"></feOffset>
        <feGaussianBlur stdDeviation="20" in="offset" result="blur"></feGaussianBlur>
        <feColorMatrix values="0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0" type="matrix"></feColorMatrix>
      </filter>
      <fill-rule="evenodd" />
    </defs>
  </svg>
</div>
```



PHP

Global Scope

A variable declared outside a function has a **GLOBAL SCOPE** and can only be accessed **outside a function**:

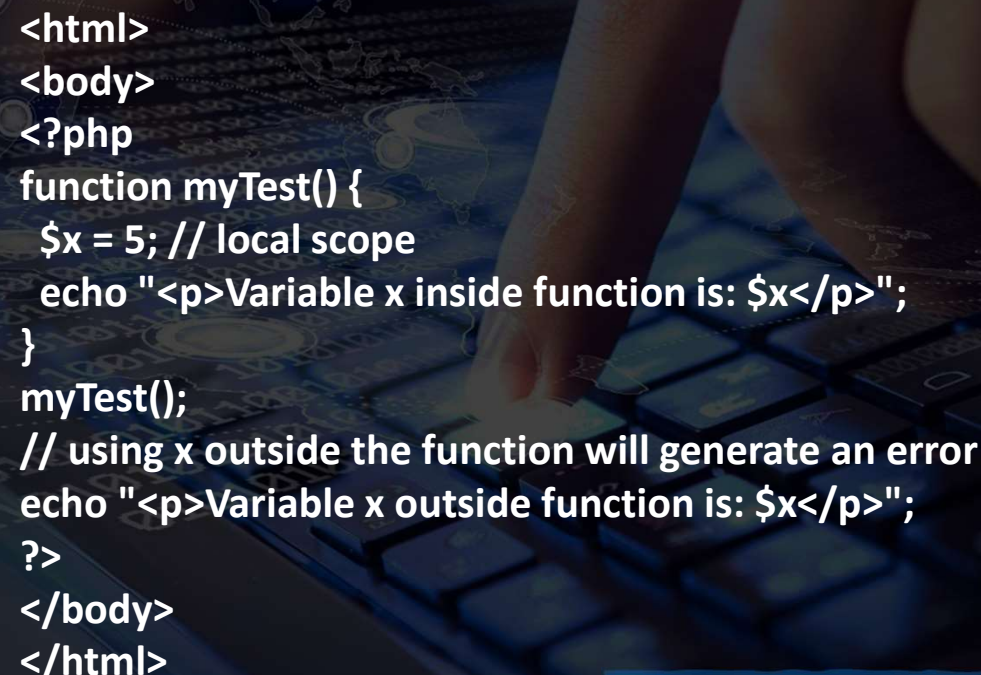
```
<html>
<body>
<?php
$x = 5; // global scope
function myTest() {
    // using x inside this function will generate error
    echo "<p>Variable x inside function is: $x</p>";
}
myTest();
echo "<p>Variable x outside function is: $x</p>";
?>
</body>
</html>
```

Variable x inside function is:
Variable x outside function is: 5



local Scope

A variable declared within a function has a **LOCAL SCOPE and can only be accessed within that function:**

A close-up photograph of a person's hand typing on a laptop keyboard. The image is dark and moody, with a blue tint. Overlaid on the image is a block of PHP code in a white, monospaced font. The code demonstrates a function that echoes a message and then attempts to use a variable outside its scope, which would result in an error.

```
<html>
<body>
<?php
function myTest() {
    $x = 5; // local scope
    echo "<p>Variable x inside function is: $x</p>";
}
myTest();
// using x outside the function will generate an error
echo "<p>Variable x outside function is: $x</p>";
?>
</body>
</html>
```

Variable x inside function is:5
Variable x outside function is:



PHP

PHP The global Keyword

the global keyword is used to access a global variable from within a function. To do this, use the global keyword before the variables (inside the function):

```
<html>
<body>
<?php
$x = 5;
$y = 10;
function myTest() {
    global $x, $y;
    $y = $x + $y;
}
myTest(); // run function
echo $y; // output the new value for variable $y
?>
</body>
</html>
```

15



PHP

PHP The static Keyword

Normally, when a function is completed/executed, all of its variables are deleted. However, sometimes we want a local variable NOT to be deleted. We need it for a further job.

To do this, use the static keyword when you first declare the variable:

```
<html>
<body>
<?php
function myTest() {
    static $x = 0;
    echo $x;
    $x++;
}
myTest();
echo "<br>";
myTest();
echo "<br>";
myTest();
?>
</body>
</html>
```

0
1
2

```
<defs>
  <linearGradient x1="100%" y1="0%" x2="0%" y2="100%" id="grad1">
    <stop stop-color="black" offset="0%" />
    <stop stop-color="white" offset="100%" />
  </linearGradient>
</defs>
<rect width="800" height="450" rx="0" fill="url(#grad1)" />
</svg>
<div class="media-control">
  <svg width="96" height="96" viewBox="0 0 96 96">
    <defs>
      <linearGradient x1="0%" y1="0%" x2="100%" y2="100%" id="grad2">
        <stop stop-color="black" offset="0%" />
        <stop stop-color="white" offset="100%" />
      </linearGradient>
      <filter x="500" y="500" width="500" height="500">
        <feOffset dy="16" in="SourceAlpha" result="offsetAlpha">
          <feGaussianBlur stdDeviation="4" in="offsetAlpha">
            <feColorMatrix values="0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0" type="matrix">
              </feColorMatrix>
            </feGaussianBlur>
          </feOffset>
        </filter>
      </defs>
      <rect width="96" height="96" rx="48" fill="url(#grad2)" />
      <rect width="96" height="96" rx="48" fill="url(#grad2)" />
    </svg>
  </div>
```



PHP

```
<!-- ... -->
<defs>
  <linearGradient x1="100%" y1="0%" x2="0%" y2="100%" id="grad1">
    <stop stop-color="808080" offset="0%" />
    <stop stop-color="000000" offset="100%" />
  </linearGradient>
</defs>
<rect width="800" height="450" rx="0" fill="url(#grad1)" />
</svg>
<div class="media-control">
  <svg width="96" height="96" viewBox="0 0 96 96">
    <defs>
      <linearGradient x1="0%" y1="0%" x2="100%" y2="100%" id="grad2">
        <stop stop-color="808080" offset="0%" />
        <stop stop-color="000000" offset="100%" />
      </linearGradient>
      <filter x="500" y="500" width="500" height="500">
        <feOffset dy="16" in="SourceGraphic" result="offset"></feOffset>
        <feGaussianBlur stdDeviation="4" in="offset" result="blur"></feGaussianBlur>
        <feColorMatrix values="0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0" type="matrix" />
      </filter>
      <fill-rule="evenodd" />
    </defs>
    <rect width="96" height="96" rx="48" fill="url(#grad2)" />
    <rect width="96" height="96" rx="48" fill="url(#grad2)" />
  </svg>
</div>
```

```
function increment() {
  $x = 0;
  echo $x;
  $x++;
}
```

```
increment();
increment();
increment();
```

```
function increment() {
  static $x = 0;
  echo $x;
  $x++;
}
```

```
increment();
increment();
increment();
```

0
0
0

0
1
2



PHP

```
<?php
function codes()
{
    $x = 100;
    echo "<p>$x</p>";
    $x++;
}
codes();
codes();
codes();
codes();
echo "<hr>";
function cracker()
{
    static $y = 200;
    echo "<p>$y</p>";
    $y--;
}
cracker();
cracker();
cracker();
cracker();
?>
```

100
100
100
100

200
199
198
197

```
<?php
function codes()
{
    $x = 100;
    echo "<p>$x</p>";
    $x++;
}
codes();
codes();
codes();
codes();
echo "<hr>";
function cracker()
{
    static $y = 200;
    echo "<p>$y</p>";
    $y--;
}
cracker();
cracker();
cracker();
cracker();
?>
```

The image features a central photograph of two hands shaking in a firm grip, signifying a business agreement or partnership. The hands are wearing dark suit sleeves and white shirt cuffs. The background is a blurred office setting. Overlaid on the center of the image is a bright red rectangular box containing the word "THANKS" in large, white, bold, sans-serif capital letters. On the left side, there is a vertical column of ten colored squares in a grid pattern, with colors including dark blue, red, yellow, teal, purple, orange, dark grey, light blue, lime green, and light grey. A solid red vertical bar is positioned to the right of the grid. On the right side, there are several horizontal red bars of varying lengths, some with rounded ends, creating a dynamic, layered effect.

THANKS