

عملگرها، شرط‌ها و حلقه‌های تکرار، PHP مفاهیم پایه

جابجایی مقادیر متغیرها (روش سنتی) ۱.

نشان می‌دهد. این روش \$khali این برنامه نحوه جابجایی مقادیر متغیرها (اعداد و رشته‌ها) را با استفاده از یک متغیر کمکی موقت به نام ساده‌ترین راه برای تضمین حفظ مقادیر است.

```
<?php
$a1 = 10;
$a2 = "a";
$a3 = 20;
$a4 = "b";

echo " مقدار متغیر a1=$a1 ";
echo "";
echo " مقدار متغیر a2=$a2";
echo "";
echo " مقدار متغیر a3=$a3";
echo "";
echo " مقدار متغیر a4=$a4";
echo "";

$khali = $a1;
$a1 = $a3;
$a3 = $khali;
$khali = $a2;
$a2 = $a4;
$a4 = $khali;

echo "مقدار متغیر بعد از جابجایی a1=$a1 ";
echo "";
echo "مقدار متغیر بعد از جابجایی a2=$a2";
echo "";
echo "مقدار متغیر بعد از جابجایی a3=$a3";
echo "";
echo "مقدار متغیر بعد از جابجایی a4=$a4";
?>
```

جابجایی مقادیر متغیرها (روش ریاضی) ۲.

این برنامه نشان می‌دهد که چگونه می‌توان مقادیر دو متغیر عددی را بدون استفاده از هیچ متغیر موقت، بلکه تنها با استفاده از عملیات جمع و تفریق، جابجا کرد.

```
<?php
$a1 = 10;
$a2 = 30;
$a3 = 20;
$a4 = 40;

echo " مقدار متغیر a1=$a1 ";
// (نمایش مقادیر اولیه) ...

$a1= $a1+$a3;
$a3= $a1-$a3;
$a1= $a1-$a3;

$a2= $a2+$a4;
$a4= $a2-$a4;
$a2= $a2-$a4;

echo " مقدار متغیر بعد از جابجایی a1=$a1 ";
// (نمایش مقادیر نهایی) ...
?>
```

۳. اولویت عملگرها و افزایش/کاهش (Increment/Decrement)

را در یک عبارت ریاضی بیچیده نشان می‌دهد تا تاثیر ($++$y$) و پسوندی ($$x++$) این قطعه کد عملکرد و اولویت عملگرهای افزایشی بیشوندی آن‌ها بر مقادیر متغیرها در حین محاسبه مشخص شود.

```
<?php
$x=6;
$y=8;
$r=(++$y*$x++) - ($y*$x+3);
echo $r;
?>
```

۴. عملگرهای منطقی (Logical Operators)

را نمایش می‌دهد، که برای ارزیابی چند شرط به طور (|) OR و (&&) AND این بخش دو نمونه از نحوه عملکرد عملگرهای منطقی مانند همراهان به کار می‌برند.

الف) مثال اول

نمایش داده (True/False) ارزیابی شده و نتایج منطقی \$x=15 بر روی شرایط مختلف متغیر (|) OR و (&&) AND در این مثال، عملگرهای می‌شود.

```
<?php
// اول.php.txt
// PHP Logical Operators

$x = 15;
$a = ($x >= 15);
$b = ($x <= 14);
$c = ($a && $b);
$d = ($a || $b);

echo "a = $a
";
echo "b = $b
";
echo "c = $c
";
echo "d = $d
";
?>
```

ب) مثال دوم

با استفاده از عملگر سه‌تایی (? :) نمایش (است b a && b که حاصل) ارزیابی شده و نتیجه عبارت منطقی \$x=10 در این مثال، متغیر داده می‌شود.

```
<?php
$x = 10;
$a = ($x >= 40);
$b = ($x <= 70);
$c = ($a && $b);

echo "a = $a
";
echo "b = $b
";
echo "c = $c
";
echo ($c) ? "بیست" : ".است.";
?>
```

سوال و جواب (مدیریت تاریخ و روزهای ماه)

این بخش یک سوال مهم امتحانی درباره تغییر منطق شرطی برای محاسبه روزهای ماه و پاسخ کدنویسی شده آن را نشان می‌دهد.

متن سوال بسیار مهم امتحانی:

مثال ۹ صفحه ۴۲ کتاب را طوری تغییر دهید که اگر عدد ۱۲ وارد شد در خروجی ۲۹ چاپ شود و اگر بالای ۱۲ وارد شود باقیمانده تقسیم بر ۱۲ را بچای عدد ماه در نظر گیرد.

کد جواب سوال بسیار مهم امتحانی:

را برای تعیین تعداد روزهای ماه پیاده‌سازی می‌کند. همچنین شرط‌های خاص خواسته شده در سوال (اگر ماه ۱۲ باشد، ۲۹ روز؛ اگر ماه بالای ۱۲ باشد، استفاده از باقیمانده تقسیم بر ۱۲) لحاظ شده‌اند.

```
<?php
$mon =13; // ۱۲ برای تست شرط بالای ۱۲
if ($mon<=12 && $mon>=1)
{
    if ($mon > 12) { echo "برنامه ادامه پیدا نمیکند";
    elseif ($mon == 12) {echo 29; }
    elseif ($mon <= 6) {echo 31; }
    else { echo 30; }
}
else
{
    $mon = $mon % 12; // اگر ماه بالای ۱۲ بود، باقیمانده تقسیم بر ۱۲ در نظر گرفته می‌شود
    echo $mon;
    echo " ";
    if ($mon == 12) {echo 29; }
    if ($mon <= 6) {echo 31; }
    if ($mon >=6 && $mon<=12) {echo 30; }
}
?>
```

حلقه‌های تکرار (ایجاد جدول ضرب) ۶.

تو در تو را نشان می‌دهد. حلقه بیرونی برای سطرها و حلقه درونی برای ستون‌ها (که در اینجا هر for این قطعه کد نحوه استفاده از حلقه‌های دو تا ۱۰ تکرار می‌شوند) استفاده شده‌اند تا مقادیر جدول ضرب تولید و چاپ شوند.

```
<?php  
border="1">  
  
for ($i=1 ; $i <=10; $i++)  
{  
    for ($d=1 ; $d <=10; $d++)  
        echo $d * $i ;  
    echo "  
-----  
";  
}  
?>
```

مثال ۱) حلقه‌های تکرار و دستور continue

برقرار است، اجرای فعلی if باعث می‌شود که وقتی شرط continue از ۱ تا ۴۵ را اجرا می‌کند. استفاده از دستور for این برنامه یک حلقه بلافاصله متوقف شده و به تکرار بعدی برود، در نتیجه اعداد موجود در بازه‌های ۱ تا ۱۰ (منحصرآ) و ۳۰ تا ۴۰ (منحصرآ) از خروجی حذف می‌شوند.

```
<?php  
for ( $i=1 ; $i<=45 ; $i++)  
{  
    if ( $i>0 && $i<11) continue;  
    if ( $i>30 && $i<40) continue;  
    echo "number $i  
";  
}  
?>
```

مثال ۲) حلقه‌های تکرار و دستور break

را در یک حلقه تکرار نشان می‌دهد. اعدادی که continue این مثال پیش‌رفته‌تر، نحوه حذف جند بازه عددی مختلف با استفاده از چندین دستور در بازه‌های (۱ تا ۱۱)، (۱۱ تا ۳۱) و (۳۱ تا ۴۹) قرار دارند، از خروجی حذف می‌شوند.

```
<?php  
for ( $i=1 ; $i<=45 ; $i++)  
{  
    if ( $i>0 && $i<11) continue;  
    if ( $i>14 && $i<31) continue;  
    if ( $i>39) continue;  
    echo "number $i  
";  
}  
?>
```

از جمله مدیریت متغیرها، استفاده از عملگرهای ریاضی و PHP توضیح نهایی: این فایل‌ها در مجموع نمایش‌دهنده تکنیک‌های اساسی در هستند. همانند چیدن آجرهای یک ساختمان، در (for و continue) و کنترل جریان حلقه، (if/else) منطقی، ساختارهای شرطی برنامه‌نویسی نیز این قطعات کوچک (تعریف متغیر، استفاده از شرط‌ها و تکرار) با هم ترکیب می‌شوند تا ساختارهای منطقی بزرگتر و پیچیده‌تری را شکل دهند.