

Koşullu İfadeler

Koşullu ifadeler, farklı koşullara bağlı olarak farklı eylemler gerçekleştirmek için kullanılır.

PHP'de şu koşullu ifade türleri yer almaktadır:

- if
- if-else
- if-elseif-else
- switch

if

if ifadesi, PHP dahil olmak üzere birçok programlama dilindeki en önemli yapı taşlarından biri olup kod bölümlerinin koşullu olarak çalıştırılabilmesini sağlar.

if (ifade)
deyim

Bir ifade kendi mantıksal değeri üzerinden değerlendirilir. ifade'nin sonucu TRUE ise, PHP deyim'i çalıştırır, FALSE ise göz ardı eder. Aşağıdaki 3 örnek de, \$a değeri \$b değerinden büyük olduğu takdirde a > b metnini görüntüleyecektir:

```
1. <?php
2.
3.     if ($a > $b)
4.         echo "a büyüktür b";
5.
6.     if ($a > $b)
7.     {
8.         echo "a büyüktür b";
9.     }
10.
11.     if ($a > $b) echo "a büyüktür b";
12.
13. ?>
```

Bir *if* ifadesinin sonucu TRUE ise yukarıda görüldüğü üzere 3 farklı yapı ile istenilen kod bloğu çalıştırabilir. Eğer süslü parantezlerle {} bir kapsam oluşturulmuşsa bu alan içerisinde istenilen kadar kod satırı yazılabilir ve çalıştırılabilir. Süslü parantezler olmadan bir *if* yapısı kurulmuşsa o zaman ya aynı satırdaki çalıştırılmak istenilen kod bloğu devreye girer ya da bir sonraki satırdaki ilk kod bloğu devreye girer.

else

Belli bir koşul sağlandığında bir deyimin, o koşul sağlanmadığında başka bir deyimin çalıştırılmasını isteyebilirsiniz. Bu gibi durumlar için *else* kullanılmaktadır. *else*, bir *if* deyimini FALSE döndürdüğü takdirde bir deyim çalıştırmak üzere genişletir. Örneğin, aşağıdaki kod \$a değeri \$b değerinden büyükse a büyüktür b, büyük değilse a büyük değildir b iletisini görüntüleyecektir.

```
1. <?php
2.
3.     if ($a > $b)
4.     {
5.         echo "a büyüktür b";
6.     }
7.     else
8.     {
9.         echo "a büyük değildir b";
10.    }
11.
12. ?>
```

Else deyimi yalnızca *if* deyiminin FALSE döndürdüğü durumda çalıştırılacaktır.

elseif

elseif, isminden de anlaşılacağı gibi, *if* ve *else* deyimlerinin bir araya gelmesiyle oluşur. *else* gibi, orijinal *if* ifadesinin FALSE döndürdüğü durumda farklı bir deyimin çalıştırılması için kullanılır. Ancak, *else*'ten farklı olarak, ilgili deyimi sadece kendisine ait koşullu ifade TRUE değerini döndürdüğü takdirde çalıştıracaktır. Örneğin, aşağıdaki kod a büyüktür b, a eşittir b ya da a küçüktür b iletisini görüntüleyecektir.

```
1. <?php
2.
3.     if ($a > $b) {
4.         echo "a büyüktür b";
5.     }
6.     elseif ($a == $b) {
7.         echo "a eşittir b";
8.     }
9.     else {
10.        echo "a küçüktür b";
11.    }
12.
13. ?>
```

Hafta 2 – Ders 2 | Koşullu İfadeler

Aynı *if* deyimine ait birden çok *elseif* bulunabilir. İlk önce hangi *elseif* deyimi TRUE değerini döndürürse o deyim çalıştırılır. Bu deyimi '*elseif*' biçiminde tek sözcük olarak kullanabileceğiniz gibi, '*else if*' biçiminde iki sözcük olarak da kullanabilirsiniz, sonuç aynı olacaktır.

elseif ifadesi yalnızca kendisinden önceki *if* ifadesi ve bu ifadeye bağlı kendisinden önce gelen diğer tüm *elseif* ifadeleri FALSE ile sonuçlandığında çalıştırılır.

switch

switch deyimi, aynı ifade içerisinde yer alan bir dizi *if* deyimine benzer. Birçok durumda, aynı değişkeni (ya da ifadeyi) birden çok değerle karşılaştırmak ve her karşılaştırmamanın sonucunu ayrı bir kod parçası çalıştırarak değerlendirmek isteyebilirsiniz. İşte bu, *switch* deyiminin tercih edilme sebebidir.

Aşağıdaki iki örnek aynı şeyi yazmanın iki farklı yolunu göstermektedir. Birincisi bir dizi *if* ve *elseif* deyiminin kullanımını, ikincisi ise *switch* deyiminin kullanımını örneklemektedir.

```
1. <?php
2.
3.     // if-elseif
4.
5.     if ($i == 0) {
6.         echo "i eşittir 0";
7.     }
8.     elseif ($i == 1) {
9.         echo "i eşittir 1";
10.    }
11.    elseif ($i == 2) {
12.        echo "i eşittir 2";
13.    }
14.
15.    // switch
16.
17.    switch ($i) {
18.        case 0:
19.            echo "i eşittir 0";
20.            break;
21.        case 1:
22.            echo "i eşittir 1";
23.            break;
24.        case 2:
25.            echo "i eşittir 2";
26.            break;
27.    }
28.
29. ?>
```

switch ile dizge kullanımı

```
1. <?php
2.
3.     switch ($i) {
4.         case "elma":
5.             echo "i bir elmadır";
6.             break;
7.         case "çikolata":
8.             echo "i bir çikolatadır";
9.             break;
10.        case "kek":
11.            echo "i bir kektir";
12.            break;
13.        }
14.
15. ?>
```

Hatalardan kaçınmak için *switch* deyiminin nasıl çalıştığını anlamak önemlidir. *switch* deyimi satır satır ele alınır (aslında deyim deyim). Başlangıçta, hiçbir kod çalıştırılmaz. Yalnızca *switch* ifadesiyle birlikte verilen değer ile uyuşan *case* deyimi bulunduğu anda PHP ilgili deyimleri çalıştırmaya başlar. PHP *switch* bloğu sonlanana kadar ya da ilk *break* deyimi ile karşılaşınca kadar çalışmasına devam eder. Bir *case* bloğunun sonuna *break* koymazsanız, PHP sonraki *case* deyiminden kodları çalıştırmaya devam eder. Örnek:

```
1. <?php
2.
3.     switch ($i) {
4.         case 0:
5.             echo "i eşittir 0";
6.         case 1:
7.             echo "i eşittir 1";
8.         case 2:
9.             echo "i eşittir 2";
10.        }
11.
12. ?>
```

Burada, $\$i$ 0'a eşit ise, PHP tüm echo deyimlerini çalıştıracaktır! $\$i$ 1'e eşit ise, PHP son iki echo deyimini çalıştıracaktır! Beklediğiniz sonucu - 'i eşittir 2' metninin görüntülenmesi - yalnızca $\$i$ 2'ye eşitse almanız mümkün olacaktır. Bu durumun oluşmaması için, *break* deyimini kullanmayı unutmamanız önemlidir.

default durumu özel bir durumdur. Bu durum diğer hiçbir *case* ile uyuşmayan değerler için kullanılır ve en sonda yer almalıdır. Örneğin:

```
1. <?php
2.
3.     switch ($i) {
4.     case 0:
5.         echo "i eşittir 0";
6.         break;
7.     case 1:
8.         echo "i eşittir 1";
9.         break;
10.    case 2:
11.        echo "i eşittir 2";
12.        break;
13.    default:
14.        echo "i ne 0, ne 1 ne de 2'ye eşittir";
15.    }
16.
17. ?>
```

Kaynaklar

PHP Kılavuzu - <http://php.net/manual/tr/index.php>

PHP Tutorials - <https://www.w3schools.com/php/default.asp>