

### Döngüler

Programlama dillerinin en faydalı olduğu alanlardan biri de tekrarlanan işleri otomatikleştirmektedir. Eğer bir şeyi birkaç kez aynı ya da benzer şekilde yapmanız gerekiyorsa kodlarınızın bazı kısımlarını tekrar etmek için döngülere başvurabilirsiniz.

PHP'de şu döngü türleri yer almaktadır:

- while
- do-while
- for
- foreach

### while

*while* döngüsü döngü tiplerinin en basit olanıdır. Bir *if* ifadesi gibi bu da bir koşula bağlıdır. Bir *while* döngüsü ile bir *if* ifadesi arasındaki fark, *if* ifadesinin eğer koşul TRUE ise takip eden kod bloğunun bir kez çalışmasıdır. Bir *while* döngüsü ise koşul TRUE olduğu sürece kod bloğunu çalıştırmayı sürdürür. Bazen, *while* ifadesi daha başlangıçta FALSE değerini verir, bu durumda *while* etki alanındaki deyimler tek bir defa bile çalıştırılmazlar.

Bir *while* deyiminin basit kullanımı:

```
while (ifade)
    deyim
```

*if* deyiminde olduğu gibi, birden çok deyimi aynı *while* döngüsü içinde süslü parantezler arasında gruplayabilirsiniz.

```
1. <?php
2.
3.     $i = 1;
4.
5.     while($i <= 5) {
6.         echo "$i <br>";
7.         $i++;
8.     }
9.
10. ?>
```

Bu örnekte *i* değişkenine 1 değeri atanmakta ve *while* döngüsü içerisinde 5'ten küçük ya da 5'e eşit olduğu sürece kod bloğunun çalıştırılması sağlanmaktadır. Döngünün her bir adımında *i* değişkeni ekrana yazdırılıp 1 birim artırılmaktadır. *i* değişkeni bir birim artış sonrası 6 olduğunda

koşulu sağlamayı bırakacağı için ekrana yazdırma ve değerini artıma işlemleri sona erecek yani kod bloğu çalıştırılmayı bırakacaktır. Tüm bu döngü aracılığı ile 1'den 5'e kadar olan sayılar ekrana alt alta yazdırılmış olacaktır.

### do-while

*do-while* döngüsü, *while* döngüsüne çok benzerdir. *do-while* döngüsünde farklı olarak, koşul ifadesi her döngünün başında değil sonunda test edilmektedir.

```
do{
  deyim
} while (ifade)
```

Koşul ifadesinin sonda yer alması da bir *do-while* döngüsü içerisindeki kod bloğunun en az bir kez çalıştırılacağı anlamına gelir. Çünkü kontrol aşaması, kod bloğu birinci döngü turunun sonunda devreye girer.

```
1. <?php
2.
3.     $i = 0;
4.
5.     do {
6.         echo "$i <br>";
7.         $i++;
8.     } while ($i > 0);
9.
10. ?>
```

Yukarıdaki döngü tam olarak bir defa çalışacaktır (ekran çıktısı 0), ilk tekrardan sonra ifadenin doğruluğuna bakıldığında FALSE değerini verecek (\$i sıfırdan büyük değildir) ve döngünün çalışması sonlanacaktır.

```
1. <?php
2.
3.     $i = 1;
4.
5.     do {
6.         echo "$i <br>";
7.         $i++;
8.     } while ($i <= 5);
9.
10. ?>
```

Yukarıdaki döngü *while* anlatımındaki örnek kod yapısının *do-while* döngüsüne dönüştürülmüş hali olup aynı sonuçları ekrana yazdıracaktır.

### for

*for* döngüsü birer başlangıç değerine, kontrol sayacına ve artış miktarına sahip bir döngü türüdür. Temelde *while* ile benzer amacı taşımaktadır. Ancak *while* döngüsünde sınır değerler belirtilmek zorunda değilken *for* döngüsünde başlangıç ve bitiş değerleri baştan belirtilmelidir.

for (başlangıç değeri; kontrol sayacı; artış miktarı)  
deyim

- başlangıç değeri: İlk başta, bir kere çalıştırılır. Burada genelde bir sayacın ilk değeri ayarlanır.
- kontrol sayacı: Bu ifade her döngüden önce kontrol edilir. Eğer ifade FALSE sonucunu döndürürse döngü durur, ifade TRUE olduğu sürece döngü devam eder. Burada genellikle bir limit belirtilir.
- artış miktarı: Her döngünün sonunda çalışır. Burada başlangıçta belirtilen sayacın değeri ayarlanır.
- deyim: Her döngüde bir kez çalıştırılır. Bu deyim genellikle bir kod bloğudur ve döngü ile asıl yapılmak istenen kodları barındırır.

```
1. <?php
2.
3.     for ($y = 1; $y <= 10; $y++) {
4.         echo "$y <br>";
5.     }
6.
7. ?>
```

Yukarıdaki örnekte başlangıç değeri olarak *y* değişkenine 1 değeri atanmıştır. Kontrol sayacı kısmında *y* değişkeninin 10'dan küçük veya 10'a eşit olduğu sürece döngünün tekrarlanması sağlanmıştır. Artış miktarı ise her döngü sonunda *y*'nin 1 birim artmasına ayarlanmıştır. Döngünün deyim yani kod bloğunda da her döngüde *y* değişkenin ekrana alt alta yazdırılması amaçlanmıştır. Bu döngü sonunda 1'den 10'a kadar olan sayılar alt alta ekranda görüntülenecektir.

### foreach

*foreach* döngüsü dizilerde kullanılan bir döngü türüdür. Bir dizinin elemanlarına bir döngü aracılığı ile tek tek ulaşılmasına olanak sağlar.

foreach (dizi\_ifadesi as \$deger)  
deyim

Bu sözdizimi ile *dizi\_ifadesi* ile belirtilen dizinin üzerinde döngü oluşturur. Her yinelemede, sırası gelen elemanın değeri *deger* değişkenine atanır ve *dizi* göstericisi bir arttırılır, böylece bir sonraki yinelemede dizinin bir sonraki elemanına bakılması sağlanmış olur.

```
1. <?php
2.
3.     $arabalar = array("BMW", "Mercedes", "Ferrari", "Tesla");
4.
5.     foreach ($arabalar as $deger) {
6.         echo "$deger <br>";
7.     }
8.
9. ?>
```

Yukarıdaki örnekte arabalar adındaki bir dizi değişkeninde yer alan elemanlar *foreach* döngüsü aracılığıyla her bir yinelemede sırayla deger değişkenine atanmakta ve alt alta ekrana yazdırılmaktadır.

### Kaynaklar

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

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

Thomson, L. & Welling, L. (2016) *PHP and MySQL Web Development* (5th ed. pp. 46-50). Addison-Wesley Professional.