

Fonksiyonlar

Fonksiyonlar, bir programda tekrar tekrar kullanılacak ifade bloklarıdır. Bu bloklar içerisinde geçerli her tür PHP kodu kullanılabilir. Bir fonksiyon, çağrı yapan bir arabirim gerektiren, belirli bir işlemi gerçekleştiren ve sonra da isteğe bağlı olarak bir sonuç döndüren bir modüldür. Dolayısıyla bu modüller sayfa yüklenir yüklenmez değil çağrıldığında çalışır.

PHP’de hâlihazırda kullanılan birçok tanımlı fonksiyon bulunmaktadır. Bunların yanı sıra siz de kendi fonksiyonlarınızı yazabilirsiniz. Daha önceden PHP’de tanımlanmış herhangi bir fonksiyonun yapamadığı, tamamen size özgü işlevlere sahip bir fonksiyon geliştirebilirsiniz.

Fonksiyon Tanımlama

Kullanıcı tanımlı bir fonksiyon bildirimi, *function* kelimesi ile başlar. Belirlen fonksiyon adı da bir parantez () açıp kapatılarak bitirilir. Bu parantezin içinin dolu olup olmaması fonksiyonun parametrelili ya da parametresiz bir fonksiyon olduğunu, süslü parantezler {} ise fonksiyonun kapsam alanını gösterir.

```
1. <?php
2.
3.     function fonksiyonAdi() {
4.         // Çalıştırılacak kodlar buraya yazılacak
5.     }
6.
7. ?>
```

Fonksiyon isimleri, PHP'deki diğer isimlerle aynı kurallara tabidir. Geçerli bir fonksiyon ismi bir harf ya da alt çizgi ile başlar, herhangi bir sayıda geçerli harf, sayı ya da alt çizgi ile devam eder. Fonksiyon isimlerinde büyük-küçük harf duyarlılığı yoktur. Yani mesajYaz(), mesajyaz() ya da MESAJYAZ() fonksiyonları birebir aynı fonksiyonlardır.

```
1. <?php
2.
3.     function mesajYaz() {
4.         echo "Merhaba dünya!";
5.     }
6.
7.     mesajYaz(); // fonksiyonu çağırma
8.
9. ?>
```

mesajYaz() fonksiyonu çağrıldığı her yere “Merhaba Dünya” çıktısını gönderecektir.

Parametrelİ Fonksiyonlar

Parametrelİ fonksiyonlar, tekrar tekrar kullanılabilir kod bloklarına bir ya da birden çok deęişken aracılıęıyla dinamik veri-veriler gönderme yöntemidir.

```
1. <?php
2.
3.     function fonksiyonAdi($deger) {
4.         echo "Merhaba $deger"; // $deger deęişkeni dinamik olarak çıktıya eklenecek
5.     }
6.
7. ?>
```

Bir fonksiyonu varsayılan bir parametre deęeri atayarak da kullanabilirsiniz. Fonksiyona bir parametre gönderirseniz son gönderdiğiniz deęer aktif olacak, parametresiz bir çağrı yaparsanız da varsayılan deęer devreye girecektir.

```
1. <?php
2.
3.     function yukseklikAyarla($yukseklik=40) {
4.         echo "Yükseklik: $yukseklik";
5.     }
6.
7.     yukseklikAyarla(10); // Ekran çıktısı -> Yükseklik: 10
8.     yukseklikAyarla(); // Ekran çıktısı -> Yükseklik: 40
9.     yukseklikAyarla(100); // Ekran çıktısı -> Yükseklik: 100
10.
11. ?>
```

Parametrelİ bir fonksiyonda birden çok deęer de gönderebilirsiniz. Dikkat edilmesi gereken şey her bir deęerin virgüllerle ayrılarak yazılmasıdır.

```
1. <?php
2.
3.     function topla($sayi1, $sayi2) {
4.         $toplam = $sayi1 + $sayi2;
5.         echo "Toplam: $toplam";
6.     }
7.
8.     topla(8, 16); // Ekran çıktısı -> Toplam: 24
9.
10. ?>
```

topla() fonksiyonu ile gönderilecek iki parametre sayi1 ve sayi2 deęişkenleri ile yakalanacak daha sonra toplam deęişkenine bu iki sayının toplamı atanacaktır. Fonksiyon sonunda ise toplam deęeri ekrana yazdırılacaktır.

Değer Döndürme

Bazı durumlarda fonksiyonlar içerisinde bir *echo* komutu kullanarak ekran çıktısı almak istemeyebilirsiniz. Bu durumlarda fonksiyonda üretilen sonucu bir *return* ifadesi ile ekrana yazdırmadan döndürebilirsiniz. Dönen değer üzerinde dileğiniz işlemleri -ekrana yazdırma, değerini değiştirme, format dönüştürme- yapabilirsiniz.

```
1. <?php
2.
3.     function toplama($sayi1, $sayi2) {
4.         $toplam = $sayi1 + $sayi2;
5.         return $toplam;
6.     }
7.
8.     toplama(8, 16); // Ekran çıktısı olmayacaktır
9.
10.    echo toplama(8, 16); // Ekran çıktısı -> 24
11.
12.    echo "Sonuç: ".(toplama(8, 16) + 3); // Ekran çıktısı -> Sonuç: 27
13.
14. ?>
```

Fonksiyonlarda global ifadesi

Fonksiyonlar varsayılan olarak kendi kapsamlarında tanımlanan değişkenleri ya da parametre ile gelen değişkenleri tanırlar. Bu kapsamlar dışında tanımlanan değişkenler ise tanınmazlar. Aynı zamanda kendi kapsamında tanımlanan değişkenleri de fonksiyon dışında kullanırlamazlar.

```
1. <?php
2.
3.     function isimOlustur() {
4.         $isim = 'Gazi';
5.     }
6.
7.     isimOlustur();
8.     echo $isim; // isim adında bir değişkenin tanımlanmadığına yönelik hata mesajı görülecektir
9.
10.    $deger = "Beltek";
11.
12.    function degerYazdir() {
13.        echo $deger;
14.    }
15.
16.    degerYazdir(); // deger adında bir değişkenin tanımlanmadığına yönelik hata mesajı görülecektir
17.
18. ?>
```

BELTEK-04-3-970 | İnternet Programcılığı [PHP&MySQL]

Hafta 3 – Ders 3 | Fonksiyonlar

Değişkenlerin tanımlanmadığına yönelik bu hataları almamak için *global* ifadesinin değişkenlerle birlikte bir kez kullanılması gerekir. Bu sayede herhangi bir yerde tanımlanan bir değişken hem fonksiyon içinde ve hem de fonksiyon dışında geçerlik kazanmış olacaktır.

```
1. <?php
2.
3.     function isimOlustur() {
4.         global $isim; // isim değişkeni fonksiyonun dışında da kullanılabilir hâle geldi
5.         $isim = 'Gazi';
6.     }
7.
8.     isimOlustur();
9.     echo $isim; // Ekran çıktısı -> Gazi
10.
11.     $deger = "Beltek";
12.
13.     function degerYazdir() {
14.         global $deger; // deger değişkeni fonksiyonun içinde de kullanılabilir hâle geldi
15.         echo $deger;
16.     }
17.
18.     degerYazdir(); // Ekran çıktısı -> Beltek
19.
20. ?>
```

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. 140-143). Addison-Wesley Professional.

