

Bölüm 7

FONKSİYONLAR VE DİZİLER

7.1. Fonksiyonlar

Bu bölümde öncelikler nasıl fonksiyon yazılacağını göreceğiz. Fonksiyonların bir amacı uzun programları mantıklı alt bölmelere ayırmaktır. Her fonksiyon programın bir alt bölümünü oluşturur. Böylece program karmaşıklığı azaltılmaya ve program anlaşılabilirliği artırılmaya çalışılır.

Fonksiyonların ikinci bir amacı da yazılan bir kodun tekrar tekrar kullanılabilmesidir. Web sitenizin birçok sayfasında aynı problemi değişik girdiler için çözeniz gerekebilir. İşte fonksiyonlar aynı kodun web sitenizin birçok noktasında tekrar edilmesinden sizi kurtarır. Yazılan bir fonksiyon ihtiyaç duyulan her noktadan çağırılabilir. Böylece kod tekrarı azaltılmış olur.

Bir web sitesinde aynı kodun birçok sayfada tekrarlanması bir sakıncası site üzerinde değişiklik yapılmasını zorlaştırmasıdır. Eğer aynı kod birçok yerde varsa, bir değişiklik ihtiyacı olduğunda bütün yerlerdeki kodların değişmesi gerekir. Eğer bir fonksiyon yazılmış ve bu fonksiyon birçok noktadan çağırılmışsa, bu durumda sadece fonksiyonun içindeki kodu değiştirmek yeterli olur. Fonksiyon içindeki programların normal bir sayfa içindeki program parçasından şu şekilde bir farkı vardır. Fonksiyon kodu sadece fonksiyon çağırıldığı zaman işlem görür. Fonksiyon çağırılmazsa fonksiyon içindeki kod işlem görmez. Ama bir php sayfasındaki diğer program kodları sırayla işlem görür. Sayfanın basından işlem baslar ve sırayla php komutları islenir. Fonksiyon tanımlama kısımlarına gelindiğinde ise fonksiyon gövdeleri işlem görmez. O kısımlar sadece fonksiyon çağırıldığı zaman işlem görür.

7.1.1 Fonksiyon çağırma

Fonksiyon içeren programlar geliştirmenin iki aşaması vardır. İlk aşama fonksiyonların yazılmasıdır. İkinci aşama ise fonksiyonların ihtiyaç olan yerlerden çağırılmasıdır. Su ana kadar biz php kütüphanesindeki bazı fonksiyonları kullandık ama hiç fonksiyon yazmadık. Şimdi önce php kütüphanesindeki bir diğer fonksiyonu çağırma örneği yapalım. Sonra da yeni fonksiyon yazmayı görelim. Fonksiyonların birer ismi vardır. Fonksiyon yazılırken isimleri belirlenir. Her değişkenin ve php sayfasının nasıl bir ismi varsa her fonksiyonun da bir ismi vardır. Fonksiyon çağırma için bu isim kullanılır. Eğer fonksiyon parametre alıyorsa çağırılırken parametreleri de gönderilir.

PHP kütüphanesinde `phpinfo` isimli bir fonksiyon vardır. Bu fonksiyon php'nin kurulu olduğu sistem hakkında bilgiler verir. Şimdi bu fonksiyonu çağıralım;

Video16

Fonksiyon _simleri Harf Kipine Duyarlı Degildir

PHP dilinde değişkenler harf kipine duyarlıdır. Fakat fonksiyon isimleri harf kipine duyarlı değildir. Dal ayısıyla `phpinfo` fonksiyonu aşağıdaki her hangi bir şekilde yazılabilir.

```
phpinfo();  
Phpinfo();  
PHPinfo();  
PHPINFO();
```

7.1.2 Fonksiyon Yazmak

Bu bölümde php dilinde bir fonksiyonun nasıl yazılacağını göreceğiz. Bunun için bir dakika ekleme uygulaması yapılacaktır. O anki zamana dakika ekleme yapan bir php sayfası yazılmıştır. Dakika ekleme kısmı farklı bir php dosyasında bir fonksiyon içinde yazılmıştır. Dolayısıyla dakika ekleme fonksiyonu her sayfadan çağırılabilir.

O anki zamana gireceğimiz dakika miktarını ekleyen bir PHP sayfası oluşturalım.

Video17

Ekle.php

```
<?php
// verilen $zaman degiskenine
// $miktar kadar dakika ekleyen bir fonksiyon
function dakikaEkle($zaman, $miktar){
    $zaman = $zaman + $miktar*60;
    return $zaman;
}

?>
```

PHP dilinde fonksiyonlar `function` anahtar kelimesiyle başlar. Bu kelimenin hemen ardından ise fonksiyon ismi gelir. Bu örnekte fonksiyona `dakikaEkle` ismi verilmiştir. Fonksiyon isimleri de değişken isimleri gibi verilirler. Değişken isimleri için geçerli olan kurallar fonksiyon isimleri için de geçerlidir. Değişken isimlerinden farklı olarak fonksiyon isimleri harf kipine duyarsızdır. Değişken isimleri ise harf kipine duyarlıdır.

Fonksiyon isminden sonra parantez vardır. Parantez içine bu fonksiyonun parametre olarak alacağı değerler yazılır. Bu örnekteki fonksiyon iki tane parametre almaktadır: `$zaman` ve `$miktar`. Parantezden sonra küme parantezi işaretiyle fonksiyon gövdesi başlar. Bu örnekte fonksiyon gövdesinde iki satırlık bir kod vardır. Daha sonra da fonksiyonun bittigini gösteren kapanış küme parantezi gelir.

Bu örnekte zaman değeri saniye cinsinden tutulmaktadır. Biz ise dakika eklemek istiyoruz. Onun için ekleme yapılmadan önce dakika değeri 60 ile çarpılarak saniyeye çevrilir. Bu şekilde saniye olarak eklenir.

return anahtar kelimesi

Bir fonksiyondan değer döndürmek için `return` anahtar kelimesi kullanılır. Bu örnekte eklemeyen sonra elde edilen zaman değeri döndürülmektedir. Bu değeri döndürmek için `return` anahtar kelimesi kullanılmıştır.

```
Cağır.php
<?php
include("ekle.php");//az önce oluşturmuş olduğumuz php dosyasını bu sayfaya
çağırıyoruz.
$dm = 100;
$z1 = mktime();
$z2 = dakikaEkle($z1, $dm);

echo "<hr>";
echo "Su anki zaman: " . date('H:i') . "<br>";
echo "Eklenen dakika miktarı: $dm <br>";
echo "Eklenince: " . date('H:i', $z2) . "<br>";

?>
```

İnclude

PHPde harici dosyaları programa eklemek için include komutunu kullanırız. include ettiğimiz (include ile programa eklediğimiz) dosyaların içeriği programda eklediğimiz yere eklenir. Bunu sık kullandığımız fonksiyon, kod veya yazılar için kullanabiliriz.

mktime ve date fonksiyonları

PHP dilinde zamanı göstermek ve zamanla ilgili işlemler yapmak için birçok fonksiyon vardır. Bu fonksiyonlardan iki tanesi bu sayfada kullanılmıştır.

mktime: Bu fonksiyon o anki zamanı saniye cinsinden döndürür. Döndürülen saniye miktarı 1 ocak 1970'ten bu yana geçen süredir. Bu şekilde zaman tutma birçok dilde ve işletim sisteminde yaygın olarak kullanılır. Bu zamana eklemek istenen süre de yine saniye olarak eklenmelidir. Bu örnekte `mktime` fonksiyonunun döndürmüş olduğu değer `$z1` değişkenine atanmıştır. Daha sonra bu değere ekleme yapılacaktır.

date: Bu fonksiyon zamanı yazdırmak için kullanılır. Parametre olarak verilen string zamanın yazılacağı formatı belirler. Diğer parametre ise yazdırılacak zaman değeridir. İkinci parametre verilmezse `date` fonksiyon o anki zamanı yazdırır. `Date` fonksiyonu bu sayfada iki defa kullanılmıştır. İlk kullanılışında bu fonksiyona tek parametre gönderilmiştir. Dolayısıyla o anki zamanı göstermektedir. Sayfa içinde şu şekilde kullanılmıştır:

```
echo "Su anki zaman: " . date('H:i') . "<br>";
```

Bu kullanımda `date` fonksiyonuna `'H:i'` stringi parametre olarak gönderilmiştir. Bu stringde kullanılan `H` harfi saatin yazılacağını `i` ise dakikanın yazılacağını gösterir. Bu iki harf arasında bulunan iki nokta ise saat ile dakika arasına iki nokta konulacağını belirtir. Gerçekten de sonuçta o anki zaman, tarayıcı içinde saat ve dakika olarak gösterilmiştir. Aralarına da iki nokta konulmuştur.

`Date` fonksiyonu tarihi yazdırmak için de kullanılır. Örneğin o anki tarihi gün/ay/yıl olarak yazdırmak istersek bu fonksiyonu şu şekilde çağırırız:

```
echo date('d/m/Y');
```

Bu komutun 10 şubat 2009 tarihindeki çıktısı şöyle olur:

```
10/02/2009
```

Bu kullanımda `d` harfi önce günün yazdırılacağını gösterir. Ortadaki `m` harfi ikinci olarak ayın yazılacağını gösterir. Sondaki `Y` harfi ise yılın yazdırılacağını gösterir. Bu üç harfin arasındaki taksim işareti ise tarih yazılırken araya taksim işaretinin konulacağını gösterir.

7.2. Diziler

7.2.1.Dizi değişkenlerin tanımı

Dizi değişkenleri bütün programlama dillerinde bir değişkene sınırsız sayıda düzenli olarak veri yüklememize yarayan bir fonksiyondur.

PHP dilinde dizi işlemleri mümkün olduğunca kolay olacak şekilde tasarlanmıştır. Dizilerin sayısal indisleri olabildiği gibi string tipinde indisleri de olabilir. Sayısal indisler de sırayla olmak zorunda değildir.

En basit anlamda dizi kullanımını aşağıdaki gibidir;

```
$dizi[ ]="murat";
```

```
$dizi[ ]="şenol";
```

```
$dizi[ ]="hasibe";
```

Burada \$dizi[] değişkenine üç veri yerleştirdik. Peki bu verileri nasıl ekrana basarız? Bu işlemden önce bir hususu belirtmekte yarar vardır. Programlama dillerinde sayı sayma işlemi 0 dan başlar. Bu durumda yukarıda ki dizi değişkenlerini sıralamak için en üstte bulunan dizinin değişkenin sıra numarası 0, 3. dizinin sıra numarası ise 2 dir. Ekrana yazdırma işlemine gelince;

```
echo $dizi[1];
```

burada 1 numaralı dizimizi basacaktır. İşlem sırasına göre 1 numaralı dizide “şenol” verisi vardır.

String tipinde indislerde verilebilir. Yukarıdaki örnekte otomatik olarak sayısal indisler verildi.

```
$deneme['a']="murat";
```

```
$deneme['b']="ali";
```

```
$deneme['c']="şenol";
```

```
$deneme['d']="hasibe";
```

```
echo $deneme['a'];
```

Bu komut sizinde tahmin edeceğiniz üzere ekrana **murat** yazacaktır.Burada dikkat edilmesi nokta string indislerin tırnak içinde yazılmasıdır.

Diziler başka bir yoldan da tanımlayabiliriz. Yukarıda ki örneği aşağıdaki gibide tanımlayabiliriz.

```
$deneme = array("a" => "murat", "b" => "şenol", "c"=> "Ali", "b" => "hasibe");
```

```
echo $deneme['a'];
```

Aynı sonucu vericektir.

Şimdi aşağıdaki örneğimizden diziler konusu kavrayalım;

```
<?php
/* array anahtar
kelimesiyle dizi oluşturmak */
$nolar = array(234, 435, 456);//bu dizi sayılardan oluşuyor.
$adlar = array("Ali", "şaban", "murat");//bu dizi ise stringlerden
// dizi elemanlarını yazdırmak
echo "numaralar: " . $nolar[0] . " " . $nolar[1] . " " . $nolar[2];
echo "<br><br>";
echo "isimler: " . $adlar[0] . " " . $adlar[1] . " " . $adlar[2];
echo "<br><br>";
// indisi string olan bir dizi oluşturmak
$nfs = array("adana" => 123456, "ankara" => 387872, "izmir"=> 2323466);
echo "nufuslar: " . $nfs["adana"] . " " . $nfs["ankara"] . " "
.$nfs["izmir"];
echo "<br><br>";
// indisi string olan bir diziyi array kullanmadan oluşturmak
$nfs2["adana"] = 123456;
$nfs2["ankara"] = 387872;
$nfs2["izmir"] = 2323466;
```

```
$nfs2["Kayseri"] = 233445;
// indis verilmeden ve array kullanmadan dizi olusturmak
$nolar2[] = 123;
$nolar2[] = 325;
$nolar2[] = 903;
$nolar2[] = 233;
echo "nolar2: " . $nolar2[0]." " . $nolar2[1] . " " . $nolar2[2];
echo "<br><br>";
?>
```

Video 18

Count fonksiyonu

Yukarıdaki örneği for döngüsü kullanarak da listeleyebiliriz. Ancak dizinin eleman sayısı bilmemiz gerekir For döngüsü dizideki eleman sayısı kadar dönmelidir. Dizinin her dönmesinde bir tane eleman yazdırılır. Bir dizideki eleman sayısını bulmak için count fonksiyonu kullanılır. Bu php kütüphanesindeki bir fonksiyondur. Count kullanarak örneğimizi değiştirecek olursak en son 2 satır yerine;

```
for($i=0; $i < count($nolar2); $i++){
echo "nolar2[$i]=".$nolar2[$i]. "<br><br>";
}
```

Yazmamız yeterli olacaktır.Bu şekilde çok elemanlı bir diziyi hızlı bir şekilde yazdırabiliriz.

7.2.2.Foreach döngüsünün

PHP dilinin desteklediği bir diğer döngü şekli foreach döngüsüdür. Bu döngü bir dizideki elemanlara erismek için kullanılır. Bu örnekte sıralanmış olan \$nfs dizisini

```
<?php
$nfs = array("adana" => 123456, "ankara" => 387872, "izmir"=> 2323466);

foreach($nfs as $anahtar=>$deger )
echo "$anahtar ile kayıtlı olan değişkenin verisi= $kelime ";
?>
```

yazdırmak için kullanılabiliriz.

foreach döngüsünün her dönmesinde dizideki bir elemanın indisini as ile tanımlanan \$anahtar değişkenine değerini ise \$deger ile tanımlanan değişkene atar.Bu şekilde dizinin hem indisi hem de değeri elimizde olur. atanır. Bu örnekte döngünün her dönmesinde dizinin sıradaki elemanı \$deger değişkenine atanır. Bu döngü dizinin bütün elemanlarına sırayla erişmek için kullanılır. For döngüsünün diğer haline göre daha kısadır. Ayrıca dizinin boyunu hesaplamaya da gerek yoktur.