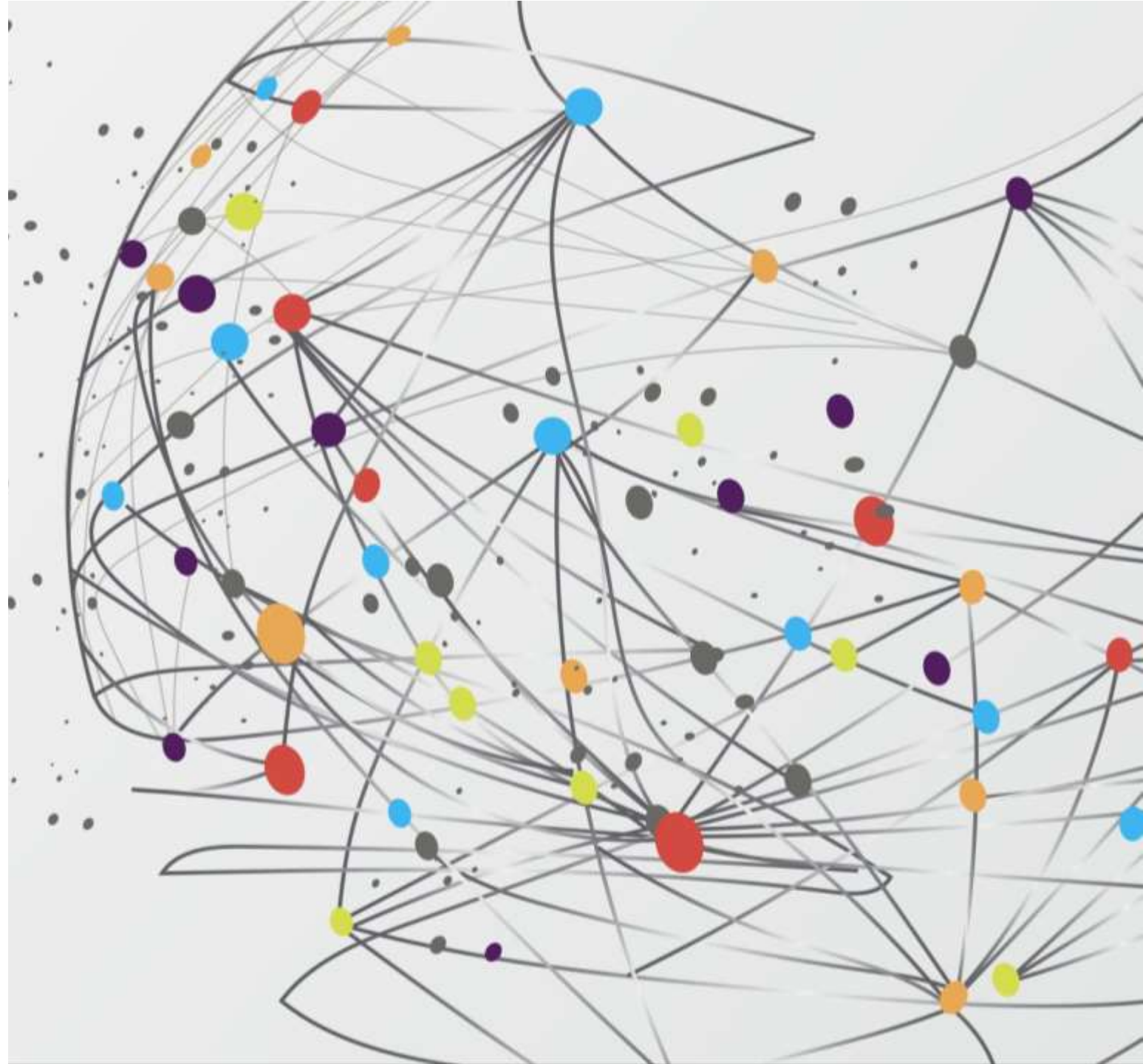

WEB TABANLI PROGRAMLAMA

BÖLÜM 11

PHP İLE PROGRAMLAMA
VERİ TİPLERİ, KOŞULLAR, DÖNGÜLER,
DİZİLER VE FONKSİYONLAR

Prof. Dr. Turgay Tugay Bilgin

turgay.bilgin@btu.edu.tr



GENEL BAKIŞ...

11.1) PHP dilinde Yorumlar

11.2) Noktalı Virgül Kullanımı

11.3) PHP Dilinde Değişkenler

11.4) PHP Değişken Adlandırma Kuralları

11.5) PHP Sabitleri

11.6) Aritmetik Operatörler

11.7) Karşılaştırma Operatörleri

11.8) Mantıksal Operatörler

11.9) Üç İşleçli Operatör

11.10) PHP Dilinde Koşul İfadeleri

11.11) PHP Dilinde “switch case” Yapısı

11.12) PHP Dilinde for Döngüleri

11.13) PHP Dilinde while Döngüleri

11.14) PHP Dilinde Dizgi (String) İşlemleri

11.14.1) Dizgi Birleştirme Operatörü

11.14.2) Dizgi Uzunluğu Bulma: strlen() Fonksiyonu

11.14.3) Dizgi İçinde Dizgi Arama: strpos () Fonksiyonu

11.15) PHP’de Diziler

11.15.1) Sayısal Dizi (Numeric Array)

11.15.2) Çağrışımsal Dizi (Associative Array)

11.16) PHP’de Fonksiyon Tanımlama

11.17) Özet

(11.1) PHP DİLİNDE YORUMLAR

- Bu ünite de bir programlama dili olarak PHP dilinin veri tipleri, koşul yapıları, aritmetik ve mantıksal operatörleri, döngü yapıları ve dizileri örneklerle açıklanmıştır. Bunlar ile birlikte fonksiyonlar konusuna da giriş yapılmıştır.
- PHP dili yazım kuralları bakımından birçok özelliğini C/C++ dillerinden almıştır. Eğer C/C++/C# veya Java dillerinden herhangi birini biliyorsanız PHP dilini öğrenmeniz çok daha kolay olacaktır.
- PHP dilinde yorum satırları eklemek C/C++ dillerindeki ile aynı olmakla birlikte (#) işareti de tek satırlı yorum olarak kullanılır. Yorum ifadeleri Tablo 11.1’de verilmiştir.

// veya #	<u>tek</u> satırlı yorum
/* */	<u>çok</u> satırlı yorum

Tablo 11.1 PHP dilinde yorum ifadeleri

```
<?php
# Burası tek satır yorum
// Burası da tek satır yorum
echo "Merhaba";
/*
    Bu yorum birden
    fazla satırdan
    oluşmaktadır.
*/
?>
```

Kod 11.1. PHP dilinde yorumlar

(11.2) NOKTALI VİRGÜL KULLANIMI

- PHP dilinde C/C++ dillerinde olduğu gibi her satır sonu noktalı virgül (;) işaretleriyle sonlandırılmalıdır.

Hatırlatma *↪ Noktalı virgül kullanmak zorunlu mudur?*

Bu kurala uyulmadığı takdirde web tarayıcı ekranında “**Parse Error**” hata mesajı görüntülenecektir.

Kod 11.2. PHP’de noktalı virgül kullanımı

```
<?php  
echo "Merhaba";  
echo "Herkes";  
?>
```

- Yandaki kod parçası ekrana MerhabaHerkes yazacaktır. Çünkü herhangi bir alt satıra geçirme komutu kullanılmamıştır.

(11.3) PHP DİLİNDE DEĞİŞKENLER

- PHP dilinde değişkenler kullanılmaya başlamadan önce tanımlanmak zorunda değildir, fakat tanımlamak faydalı bir programlama alışkanlığıdır. C/C++ dillerinden farklı olarak PHP dilinde değişken tanımlamak için **int**, **double**, **char** gibi farklı anahtar kelimeler bulunmaz, değişken tanımlarken sadece adının önüne \$ işareti eklenir.
- PHP dili değişkenlerin hangi türde olduğunu içine değer ataması yapıldığı zaman algılar. Yukarıdaki örnekte numara değişkenine 12345 değerini verirsiniz PHP bu değişkenin tipini int olarak, derece=3.16 verirsiniz tipini double olarak, adsoyad="Ali Topal" verirsiniz tipini string olarak algılayacaktır. Boolean tipinden değişken oluşturmak isterseniz TRUE veya FALSE değerini atayabilir, bir değişkene hiçbir değer atamak istemiyorsanız veya içindeki atanmış değeri silmek isterseniz NULL değerini atayabilirsiniz.

```
<?php
$numara = 12345;
$derece = 34.5;
$adsoyad = "Ali Topal";
$evli_mi = TRUE;
$maas = NULL;
?>
```

Kod 11.3. PHP'de değişkenler

Hatırlatma *☞ PHP dilinde string değişkenler tek tırnak ile mi çift tırnak ile mi tanımlanmalıdır?*

PHP dilinde string tanımlarken çift tırnak veya tek tırnak kullanmak işlem sonucunu etkiler. Eğer string değişkeni tek tırnak ile tanımlarsanız içinde başka bir değişken kullanamazsınız, string değişkeni çift tırnak ile kullanırsanız string içindeki değişkenlerin adı değil değeri otomatik olarak string içine yerleşir. Aşağıdaki örneği inceleyerek bu durumu kavrayabilirsiniz.

```
<?php
$x = "Cuma";
$mesaj = 'Bugün $x günüdür.<br>';
echo $mesaj;
$mesaj = "Bugün $x günüdür.<br>";
echo $mesaj;
?>
```

Kod 11.4. PHP’de değişkenlerde tırnak kullanımı

- Yukarıdaki kodu “c:\wamp\www” klasörüne **degisken.php** adıyla kaydediniz. Web tarayıcınızı açıp,

`http://localhost/degisken.php`

adresini yazarak çalıştırdığınızda aşağıdaki çıktı görüntülenecektir.

```
Bugün $x günüdür.
Bugün Cuma günüdür.
```

(11.4) PHP DEĞİŞKEN ADLANDIRMA KURALLARI:

- PHP dilinde değişken adlandırma kuralları tamamen C/C++ dillerindeki ile aynıdır. Özetlemek gerekirse:
 - PHP dilinde anahtar kelimeler (komutlar) değişken adı olarak kullanılamaz, Örneğin `while`, `if`, `for` gibi kelimeler değişken ismi olamaz.
 - değişkenler sayı ile başlayamazlar, bir harf veya alt çizgi (`_`) ile başlayabilirler. Örneğin `123deneme` hatalı bir değişken adı, `_123deneme` ise doğru bir değişken adıdır.
 - değişken adları büyük-küçük harf duyarlıdır.
 - değişkenler `+`, `-`, `%`, `(`, `)`, `.`, `&` gibi özel karakterler içeremezler.

(11.5) PHP SABİTLERİ

- Sabit demek, bir değer için verilen değişmez isim demektir. Bir sabitin değeri programın çalışması sırasında değiştirilemez. Sabitler büyük-küçük harf duyarlıdır. Geleneksel olarak, sabitlerin adları her zaman büyük harfle yazılır. Bir sabit tanımlamak için `define()` fonksiyonu kullanılır ve bir sabitin değerini kullanmak için sadece adını yazmak yeterlidir. PHP değişkenlerinin aksine sabitlerin başında `$` işareti kullanılmaz. Bir defa tanımlanmış olan sabitler tekrar tanımlanamaz veya değeri silinemez.

```
<?php  
define("KDV", 18);  
echo KDV;  
?>
```

Kod 11.5. PHP'de sabit kullanımı

(11.6) ARİTMETİK OPERATÖRLER

- PHP dilinde aritmetik operatörlerin kullanımı C/C++/C#/Java dillerindeki ile aynıdır. +, -, *, /, %, ++, -- operatörlerine ait bir örnek aşağıda verilmiştir.

```
<html><body>
<?php
$a = 33;
$b = 10;
//toplama operatörü
echo "$a + $b = ";
$sonuc = $a + $b;
echo $sonuc . "<br />";
//çıkarma operatörü
echo "$a - $b = ";
$sonuc = $a - $b;
echo $sonuc . "<br />";
//bölme operatörü
echo "$a / $b = ";
$sonuc = $a / $b;
echo $sonuc . "<br />";
//mod alma operatörü
echo "$a % $b = ";
$sonuc = $a % $b;
echo $sonuc . "<br />";
//arttırma operatörü
echo "++$a = ";
$sonuc = ++$a;
echo $sonuc .
"<br />";
//eksiltme operatörü
echo "--$b = ";
$sonuc = --$b;
echo $sonuc .
"<br />";
?>
</body></html>
```

-
- Yukarıdaki kodu “c : \wamp\www” klasörüne **ornek.php** adıyla kaydediniz. Web tarayıcınızı açıp,

`http://localhost/ornek.php`

adresini yazarak çalıştırdığınızda aşağıdaki çıktı görüntülenecektir.

```
33 + 10 = 43
33 - 10 = 23
33 / 10 = 3.3
33 % 10 = 3
++33 = 34
--10 = 9
```

(11.7) KARŞILAŞTIRMA OPERATÖRLERİ

- PHP dilinde karşılaştırma operatörlerin kullanımı C/C++/C#/Java dillerindeki ile aynıdır. ==, >, <, >=, <=, != operatörlerine ait bir örnek aşağıda verilmiştir.

```
<html><body>
<?php
$a = 10;
$b = 20;
// karşılaştırma operatörü
echo "($a == $b) ifadesi sonucu: ";
if ($a == $b)
    echo "Dogru";
else
    echo "Yanlis";
echo "<br />";
//küçük operatörü
echo "($a < $b) ifadesi sonucu: ";
if ($a < $b)
    echo "Dogru";
else
    echo "Yanlis";
echo "<br />";
//büyük operatörü
echo "($a > $b) ifadesi sonucu: ";
if ($a > $b)
    echo "Dogru";
else
    echo "Yanlis";
echo "<br />";
//esit operatörü
echo "($a == $b) ifadesi sonucu: ";
if ($a == $b)
    echo "Dogru";
else
    echo "Yanlis";
echo "<br />";
//büyük esit operatörü
echo "($a >= $b) ifadesi sonucu: ";
if ($a >= $b)
    echo "Dogru";
else
    echo "Yanlis";
echo "<br />";
//küçük esit operatörü
echo "($a <= $b) ifadesi sonucu: ";
if ($a <= $b)
    echo "Dogru";
else
    echo "Yanlis";
echo "<br />";
?>
</body></html>
```

- Yukarıdaki kodu "c:\wamp\www" klasörüne **ornek.php** adıyla kaydediniz. Web tarayıcınızı açıp,

<http://localhost/ornek.php>

adresini yazarak çalıştırdığınızda aşağıdaki çıktı görüntülenecektir.

```
(10 == 20) ifadesi sonucu: Yanlis
(10 < 20) ifadesi sonucu: Dogru
(10 > 20) ifadesi sonucu: Yanlis
(10 != 20) ifadesi sonucu: Dogru
(10 >= 20) ifadesi sonucu: Yanlis
(10 <= 20) ifadesi sonucu: Dogru
```

Kod 11.7. PHP'de karşılaştırma operatörleri

(11.8) MANTIKSAL OPERATÖRLER

- PHP dilinde mantıksal operatörlerin kullanımı C/C++/C#/Java dillerindeki ile aynıdır. &&, ||, ! operatörlerine ait bir örnek aşağıda verilmiştir.

```
<html><body>
<?php
$a = true;
$b = false;
// && (and) operatörü
echo "(a && b) ifadesinin sonucu: ";
if ($a && $b)
    echo "Dogru";
else
    echo "Yanlis";
echo "<br />";
```

```
// || (or) operatörü
echo "(a || b) ifadesinin sonucu: ";
if ($a || $b)
    echo "Dogru";
else
    echo "Yanlis";
echo "<br />";
// ! (DEGIL) operatörü
echo "!(a && b) ifadesinin sonucu: ";
if (!(a && b))
    echo "Dogru";
else
    echo "Yanlis";
echo "<br />";
?>
</body></html>
```

Kod 11.8. PHP'de mantıksal operatörler

-
- Yukarıdaki kodu “c : \wamp\www” klasörüne **ornek.php** adıyla kaydediniz. Web tarayıcınızı açıp,

`http://localhost/ornek.php`

adresini yazarak çalıştırdığınızda aşağıdaki çıktı görüntülenecektir.

```
(a && b) ifadesinin sonucu: Yanlis
(a || b) ifadesinin sonucu: Dogru
!(a && b) ifadesinin sonucu: Dogru
```

(11.9) ÜÇ İŞLEÇLİ OPERATÖR

- `if..else` yapısının tek satırda yazılmasını sağlayan `(?:)` operatörü C/C++/C#/Java dillerinde olduğu gibi PHP dilinde de bulunur. Kullanım biçimi aşağıdaki gibidir.

```
(koşul)? koşul_doğruysa : koşul_yanlıssa ;
```

- Bu operatör bir koşul ifadesini parametre olarak alır. Eğer koşul doğruysa “?” ile “:” arasındaki kod çalıştırılır, koşul yanlış ise “:” işaretinden satır sonuna kadar olan kod çalıştırılır.

- Aynı problemin hem `if..else` yapısı ile hem üç işleçli operatör ile çözümü aşağıdaki örnekte görülmektedir. Örnekte eğer `a` değişkeni `b` den büyük ise `sonuc` değişkenine `a` değerini, tam tersi ise `sonuc` değişkenine `b` değerini atamakta ve ekrana yazdırmaktadır.

Üç işleçli operatörlü çözüm

```
<html><body>
<?php
$a = 10;
$b = 20;
$sonuc = ($a > $b) ? 100 : 200 ;
echo $sonuc;
?>
</body></html>
```

if .. else yapısıyla çözüm

```
<html><body>
<?php
$a = 10;
$b = 20;
if( $a > $b ){
    $sonuc=100;
}else{
    $sonuc=200;
}
echo $sonuc;
?>
</body></html>
```

Kod 11.9. PHP’de üç işleçli operatör

(11.10) PHP DİLİNDE KOŞUL İFADELERİ

- PHP dilinde if..else koşul ifadelerinin kullanımı C/C++/C#/Java dillerindeki ile aynıdır. 3 farklı türde if yapısı kullanılabilmektedir:

- `if`

- `if...else`

- `if...else if...`

Bunların kullanımına ait bir örnek aşağıda verilmiştir.

Kod 11.10. PHP’de koşul ifadeleri

```
<html><body>
<?php
$plaka = 34;
if( $plaka == 17 ){
    echo "<b>Çanakkale</b>";
}else if( $plaka == 34 ){
    echo "<b>Istanbul</b>";
}else if( $plaka == 35 ){
    echo "<b>Izmir</b>";
}else{
    echo "<b>Bilinmeyen plaka!</b>";
}
?>
</body></html>
```

- Bu kod çalıştırıldığında, \$plaka değişkeninin değeri 34 olduğundan ekranda “Istanbul” yazısı görüntülenecektir.

(11.11) PHP DİLİNDE “SWITCH CASE” YAPISI

- PHP dilinde switch..case yapısının kullanımı C/C++/C#/Java dillerindeki ile aynıdır. Aşağıdaki örnekte Bölüm 11.20’deki örneğin switch..case yapısı kullanarak oluşturulmuş biçimi görülmektedir.
- Örnek çalıştırıldığında \$plaka değeri 17 olarak girilmiş olduğundan ekranda “Çanakkale” yazacaktır.

```
<html><body>
<?php
$plaka = 17;
switch ($plaka)
{
    case 17: echo "Çanakkale";
             break;
    case 22: echo "Edirne";
             break;
    case 34: echo "İstanbul";
             break;
    case 35: echo "İzmir";
             break;
    default: echo "Böyle bir plaka yok!";
}
?>
</body></html>
```

(11.12) PHP DİLİNDE FOR DÖNGÜLERİ

- PHP dilinde for döngüsünün kullanımı C/C++/C#/Java dillerindeki ile aynıdır. Aşağıdaki örnekte for döngüsünün kullanımı görülmektedir.

```
<html>
<body>
<?php
$a = 0;
$b = 0;
    for( $i=0; $i<5; $i++ )
    {
        $a += 10;
        $b += 5;
        echo ("i=$i için a=$a ve b=$b<br />" );
    }
?>
</body>
</html>
```

-
- Yukarıdaki kodu “c : \wamp\www” klasörüne **ornek.php** adıyla kaydediniz. Web tarayıcınızı açıp,

`http://localhost/ornek.php`

adresini yazarak çalıştırdığınızda aşağıdaki çıktı görüntülenecektir.

```
i=0 için a=10 ve b=5  
i=1 için a=20 ve b=10  
i=2 için a=30 ve b=15  
i=3 için a=40 ve b=20  
i=4 için a=50 ve b=25
```

(11.13) PHP DİLİNDE **while** DÖNGÜLERİ

- PHP dilinde `while` döngüsünün kullanımı da C/C++/C#/Java dillerindeki ile aynıdır. Aşağıdaki örnekte `while` döngüsünün kullanımı görülmektedir. Örnek çalıştırıldığında ekranda `<h1>` seviyesinden `<h6>` seviyesine kadar 6 farklı HTML etiketi içinde *Baslik 1*, *Baslik 2*, *Baslik 6* şeklinde ifadeler yazacaktır.

```
<html>
<body>
<?php
$i = 0;
while( $i < 6)
{
    $i++;
    echo ("<h$i>Baslik $i</h$i>" );
}
?>
</body>
</html>
```

Kod 11.13. PHP’de while döngüleri

- Yandaki örnek çalıştığında Şekil 11.1’de görülen çıktı elde edilecektir.

Baslik 1

Baslik 2

Baslik 3

Baslik 4

Baslik 5

Baslik 6

Şekil 11.1 PHP dilinde while döngüsü örneği

- Kod 11.14’de verilen örnekteki gibi koşul kontrolü süslü parantez bloğu sonuna getirilerek `while` bloğu, `do-while` döngüsü haline getirilebilir.



```
do
{
    $i++;
    echo ("<h$i>Baslik $i</h$i>" );
}while( $i < 6)
```

Kod 11.14. PHP’de do-while döngüleri

(11.14) PHP DİLİNDE DİZGİ (STRING) İŞLEMLERİ

- PHP dilinde dizgi işlemleri için onlarca farklı fonksiyon bulunmaktadır. Bu bölümde çok sık kullanılan bazı fonksiyonlar tanıtılmıştır. Tüm fonksiyonların listesine aşağıda web adresi verilen PHP'nin web sitesindeki kullanma kılavuzunda “*Dizge İşlemleri*” bölümünden erişebilirsiniz.

`http://php.net/manual/tr/ref.strings.php`

(11.14.1) DİZGİ BİRLEŞTİRME OPERATORÜ

- C/C++/Java ve Javascript dillerinden farklı olarak PHP dilinde String değişkenleri birleştirmek (arka arkaya eklemek) için nokta (.) operatörü kullanılır.

Kod 11.15. PHP'de dizgi birleştirme

```
<?php
$string1="Bugün";
$string2=" 28 Ağustos";
echo $string1 . " " . $string2;
?>
```

- Yukarıdaki örnek çalıştığında aşağıda verilen çıktı elde edilecektir.

Bugün 28 Ağustos

(11.14.2) DİZGİ UZUNLUĞU BULMA: STRLEN() FONKSİYONU

- Dizgilerin uzunluğunu bulmak `strlen()` fonksiyonu ile gerçekleştirilir. Kullanımı aşağıdaki gibidir.

Kod 11.16. PHP'de dizgi uzunluğu bulma

```
<?php  
echo strlen("Merhaba Dünya");  
?>
```

- Yukarıdaki örnek çalıştığında aşağıda verilen çıktı elde edilecektir.

13

(11.14.3) DİZGİ İÇİNDE DİZGİ ARAMA: STRPOS() FONKSİYONU

- `strpos()` fonksiyonu bir dizgi içinde başka bir dizgiyi veya karakteri aramak için kullanılır. Dizge içinde aranan ifade bulunursa, bulunan konum sıra numarası çıktı olarak üretilir. Eğer aranan ifade bulunamazsa `FALSE` değeri üretilir. Kullanımı için aşağıdaki örneği inceleyiniz:

Kod 11.17. PHP'de dizgi içinde dizgi arama

```
<?php  
echo strpos("Bu bir araba  
olabilir", "araba");  
?>
```

- Yukarıdaki örnek çalıştığında aşağıda verilen çıktı elde edilecektir.

7

(11.15) PHP'DE DİZİLER

- PHP dilinde sayısal (Numeric) diziler ve çağrışımsal (Associative) diziler olarak iki farklı türde dizi bulunmaktadır. Bunların her biri iki ayrı bölümde açıklanmıştır.

11.25.1 Sayısal Dizi (Numeric Array) : İndis değerleri sayılardan oluşan dizilerdir. Diziye her bir değer eklendiğinde indis değeri artarak büyür. Sayısal dizilerde indis değeri sıfırdan başlar. Sayısal dizi oluşturmak için `array()` fonksiyonu kullanılır. Aşağıdaki örnekte kullanımı gösterilmiştir:

```
<html>
<body>
<?php
$dizi = array( 1, 2, 3, 4, 5);
foreach( $dizi as $seleman )
{
    echo "$seleman <br />";
}
?>
</body>
</html>
```

Kod 11.18. PHP'de sayısal diziler

- Yandaki örnekte `foreach` döngüsü kullanıldığına dikkat ediniz. Dizi işlemlerinde dizinin boyutunu öğrenmeye gerek kalmadan tüm dizi elemanlarına erişim olanağı sağladığı için `foreach` döngüleri tercih edilir. `foreach` döngülerinde parantez içinde dizinin adı ve sonrasında “as” anahtar kelimesi ve sonrasında her bir dönmeye sıradaki dizi elemanını depolayacak olan geçici değişken adı tanımlanır. Bu örnekte döngünün her bir dönüşünde `$seleman` isimli değişkene, o anki dizi elemanının içeriği atanır.

- Sayısal dizilerin depoladığı değerler sayısal olmak zorunda değildir, yandaki gibi string değerler de depolayabilir.



```
<html>
<body>
<?php
$kisiler[0] = "ali";
$kisiler[1] = "metin";
$kisiler[2] = "koray";
$kisiler[3] = "izzet";
$kisiler[4] = "turgay";

foreach( $kisiler as $x ){
    echo "$x <br />";
}
?>
</body>
</html>
```

Kod 11.19. PHP'de sayısal dizilerde string değerler

- **11.25.2 Çağrışimsal Dizi (Associative Array) :** İndis değerleri `string` türden oluşan dizilerdir. Bu dizilerde dizi indislerinin artan sırada olma durumu yoktur. Değerler indis-değer çiftleri şeklinde depolanır. Çağrışimsal diziler `array()` fonksiyonu veya doğrudan anahtar-değer çifti şeklinde tanımlanabilirler.

```
<html>
<body>
<?php
/* array fonksiyonu kullanarak dizi tanımlama */
$urunler = array(
    "elma" => 2000,
    "armut" => 1000,
    "nar" => 500
);

echo "Elma fiyatı: ". $urunler['elma'] . "<br />";
echo "Armut fiyatı ". $urunler['armut'] . "<br />";
echo "Nar fiyatı ". $urunler['nar'] . "<br />";

/* deger girerek dizi tanımlama */
$plaka['canakkale'] = 17;
$plaka['edirne'] = 22;
$plaka['izmir'] = 35;

echo "Canakkale il kodu: ". $plaka['canakkale'] . "<br />";
echo "Edirne il kodu: ". $plaka['edirne'] . "<br />";
echo "Izmir il kodu: ". $plaka['izmir'] . "<br />";
?>
</body>
</html>
```

Kod 11.20. PHP’de çağrışimsal diziler

- Yukarıdaki kodu “c:\wamp\www” klasörüne **ornek.php** adıyla kaydediniz. Web tarayıcınızı açıp,

<http://localhost/ornek.php>

adresini yazarak çalıştırdığınızda aşağıdaki çıktı görüntülenecektir.

```
Elma fiyatı: 2000
Armut fiyatı 1000
Nar fiyatı 500
Canakkale il kodu: 17
Edirne il kodu: 22
Izmir il kodu: 35
```

- Çağrışımsal dizilerin çift tırnaklar içinde veya dışında kullanılması ile ilgili bir örnek Kod 11.21’da verilmiştir. Sayısal veya çağrışımsal dizileri ekrana yazdırmadan önce çift tırnakları kapatıp, nokta koyup, diziyi ekrana yazdırdıktan sonra tekrar çift tırnak açtığınızda ve sonra tekrar nokta operatörü ile birleştirdiğinizde herhangi bir sorun olmayacaktır. Bu durum aşağıdaki örnekte DOGRU (1) ve DOGRU (2) ile belirtilen satırlarda verilmiştir

```
<html>
<body>
<?php
$dizi1 = array( 1, 2, 3, 4, 5);
$dizi2 = array( "ali" => 9, "veli" => 7);

// çift tırnaklar dışında dizi elemanlarına erişim
echo "Dizi1 ilk öge: ". $dizi1[0] . "<br />"; //DOGRU(1)
echo "Dizi2 ilk öge: ". $dizi2['ali'] . "<br />"; //DOGRU(2)

// çift tırnaklar içinde dizi elemanlarına erişim
echo "Dizi1 ilk öge: $dizi1[0] <br />"; //DOGRU(3)
echo "Dizi2 ilk öge: $dizi2['ali']<br />"; //HATALI

?>
</body>
</html>
```

- Sayısal dizileri çift tırnaklı alan içinde kullandığınızda herhangi bir hata oluşmayacaktır. Bu durum DOGRU (3) satırında görülmektedir. Fakat çağrışımsal dizileri çift tırnaklı alana yazarsanız PHP derleyicisi hata verecektir. Bu durumun bir örneği HATALI şeklinde belirtilen satırda görülmektedir.

Kod 11.21. PHP’de çağrışımsal dizilerin yazdırılması

(11.16) PHP'DE FONKSİYON TANIMLAMA

- C/C++/Java dillerinde olduğu gibi PHP dilinde de fonksiyon yapıları mevcuttur. Bu sayede gereksiz kod tekrarları engellenir. PHP dilinde fonksiyon tanımlama tamamen C/C++/Java dillerindeki gibidir. Fonksiyonlar parametre alabilir ve `return` anahtar kelimesi sayesinde çağırıldıkları konuma değer döndürebilirler. Aşağıdaki örnekte parametre alan ve değer döndüren fonksiyon oluşturulmuştur.

```
<html>
<body>
<?php
function kupHesapla ($x)
{
    return $x*$x*$x;
}
$sayi=4;
echo "$sayi sayisinin küpü " . kupHesapla ($sayi) . " eder.";
?>
</body>
</html>
```

-
- Yukarıdaki kodu “c : \wamp\www” klasörüne **ornek.php** adıyla kaydediniz. Web tarayıcınızı açıp,

http://localhost/ornek.php

adresini yazarak çalıştırdığınızda aşağıdaki çıktı görüntülenecektir.

4 sayısının küpü 64 eder.

(11.17) ÖZET

- PHP dilinde yorum satırları eklemek C/C++ dillerindeki ile aynı olmakla birlikte (#) işareti de tek satırlı yorum olarak kullanılır. PHP dilinde C/C++ dillerinde olduğu gibi her satır sonu noktalı virgül (;) işaretleriyle sonlandırılmalıdır. PHP dilinde değişkenler kullanılmaya başlamadan önce tanımlanmak zorunda değildir, fakat tanımlamak faydalı bir programlama alışkanlığıdır. PHP dilinde aritmetik operatörlerin, karşılaştırma operatörlerinin ve mantıksal operatörlerin kullanımı C/C++/C#/Java dillerindeki ile aynıdır. PHP dilinde dizgi değişkenleri birleştirmek için nokta (.) operatörü kullanılır. Dizgilerin uzunluğunu bulmak `strlen()` fonksiyonu ile gerçekleştirilir. `strpos()` fonksiyonu bir dizgi içinde başka bir dizgiyi veya karakteri aramak için kullanılır. PHP dilinde sayısal diziler ve çağrışımsal diziler olarak iki farklı türde dizi bulunmaktadır. PHP dilinde fonksiyon tanımlama C/C++/Java dillerindeki gibidir. Fonksiyonlar parametre alabilir ve `return` sayesinde değer döndürebilirler.