

# WEB TASARIM I

Öğr. Gör. M. Mutlu YAPICI

Ankara Üniversitesi  
Elmadağ Meslek Yüksekokulu

# Ders İzlenesi

Hafta	Modüller/İçerik/Konular
1. Hafta	PHP Tanımı ve Sunucu Kurulumları
2. Hafta	PHP Yazım Notasyonu ve Değişkenler
3. Hafta	PHP de Karar kontrol yapıları ve Döngüler
4. Hafta	Dizi ve Dizi işlemleri
5. Hafta	Fonksiyon, Sınıf ve Nesne Kavramları
6. Hafta	HTML ve PHP
7. Hafta	PHP ile Veritabanı İşlemleri
8. Hafta	MYSQL, MYSQLİ
9. Hafta	AJAX
10. Hafta	
11. Hafta	
12. Hafta	
13. Hafta	
14. Hafta	

# Bu Ünite de Ele Alınan Konular

- PHP dilinin genel yapısı
- Yazım notasyonu
- Yazım kuralları
- Değişken ve sabit tanımlama
- Değişken veri tipleri
- Operatörler
- Karar kontrol komutları (If-else Switch-case)
- Döngü yapıları (For, while, do-While, foreach)

# Ders Kazanımları

Bu bölümü Bitirdiğimizde,

- PHP dilinin genel yapısı
- Yazım notasyonu
- Yazım kuralları
- Değişken ve sabit tanımlama
- Değişken veri tipleri
- Operatörler
- Karar kontrol komutları (İf-else Switch-case)
- Döngü yapıları (For, while, do-While)

öğrenmiş olacaksınız.

# PHP Dilinin Genel Yapısı

- PHP dilinin genel yapısında, tabii ki her dilde olduğu gibi dilin başlangıç ve bitiş noktalarını ifade eden semboller vardır. Nasıl ki HTML kodu yazarken `<HTML>` tagi ile başlayıp `</HTML>` ile bitiriyoruz ve tüm HTML kodlarını bu iki tag arasına yazıyoruz, işte PHP'de de `<?PHP .....?>` tagleri arasına PHP kodlarımızı yazmalıyız. Aksi halde kodumuz çalışmayacaktır.

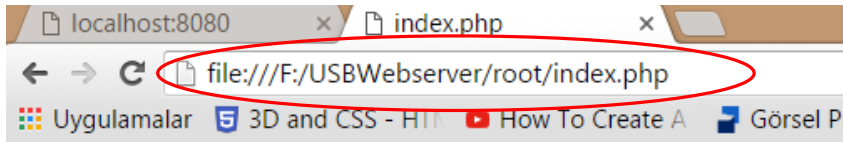
# İlk Kod

- PHP dilinde ilk kodumuzu yazalım. Eh hadi geleneği bozmayalım. Bizde ilk olarak ekrana «Merhaba Dünya» yazdıralım.
- `<?Php`
- `ECHO("Merhaba Dünya");`
- `?>`

`<?Php ?>` tagleri arasına kodumuzu yazıyoruz. ECHO() fonksiyonu PHP de ekrana bir şey yazdırmak için kullanılır. İsterseniz ECHO yerine PRINT kodunu da kullanabilirsiniz. Kod satırını ; (noktalı virgül) ile sonlandırmayı unutmayın.

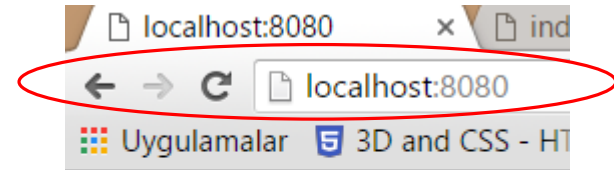
# İlk Kod

- Php kodları yazdığınız dosyanın uzantısı kesinlikle php olmalıdır.
- PHP dilinde yazdığımız kodları direkt olarak tarayıcıda çalıştıramayacağımızı daha önce söylemiştik.
- Eğer direkt olarak tarayıcıda açarsanız yazdığınız kodların çıktısı yerine kodları görürsünüz. Lokal sunucu üzerinden çalıştırmalısınız.



```
<?Php  
    echo("Merhaba Dünya");  
?>
```

**Direkt Açılınca**



Merhaba Dünya

**Sunucu Üzerinde Açılınca**

# PHP 'de Değişken Tanımlama

- Php dilinde değişken tanımlamak için değişkenin tipini yazmanıza gerek yoktur. Diğer dillere göre PHP nin güzel bir özelliğidir bu. PHP değişkeninizin içerisinde tuttuğu değere göre kendisi otomatik olarak türüne karar verir. Eğer değişken içinde 10 gibi tamsayı varsa bunun **int** türünde "ali veli" gibi bir değer varsa bunun **string** türünde ya da true false değeri varsa **boolean** türünde olduğunu anlayabilir.
- Her değişken mutlaka **\$** işaretiyle başlar.
- Örnek: **\$sayi** değişkendir **sayi** değildir.

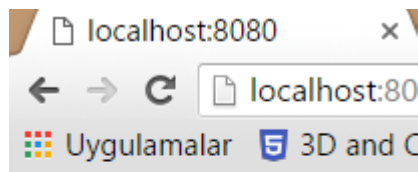


# PHP 'de Değişken Tanımlama

- Php dilinde değişken tanımlamak için değişkenin tipini yazmanıza gerek yoktur. Diğer dillere göre PHP nin güzel bir özelliğidir bu. PHP değişkeninizin içerisinde tuttuğu değere göre kendisi otomatik olarak türüne karar verir. Eğer değişken içinde 10 gibi tamsayı varsa bunun **int** türünde "ali veli" gibi bir değer varsa bunun **string** türünde ya da true false değeri varsa **boolean** türünde olduğunu anlayabilir.
- Her değişken mutlaka **\$** işaretiyle başlar.
- Örnek: **\$sayi** değişkendir **sayi** değildir.

# PHP 'de Değişken Tanımlama

- Daha önce belirttiğimiz gibi değişken tanımlarken değişken türünü belirtmemize gerek yoktur PHP kendisi değişken türünü tanıyacaktır.
- Var\_Dump() fonksiyonu PHP de değişkenlerin ve verilerin türünü öğrenmekte kullanılır.
- Var\_dump(10); \$a=true; var\_dump(\$a);
- \$a="Ali"; var\_dump(\$a);
- Ayrıca getType(\$a); /// değişken türünü verir
- Settype(\$a,double);//değişken türünü tekrar ayarlar.



```
int(10)
bool(true)
string(3) "Ali"
```

## PHP 'de Değişken Tanımlama

- İki değişken arasında bağlantı veya bir değer ile bir değişken arasında bağlantıyı C# gibi dillerde artı (+) işareti ile yapıyırduk. PHP de bu işlem için nokta (.) işareti kullanılmaktadır.

- Örnek;

```
$a="Selam"; $b="Millet"; echo $a." ".$b;
```

## PHP 'de Değişken Tanımlama

- PHP içerisinde isterseniz HTML, CSS JavaScript kodları da kullanabilirsiniz. Ancak bu kodlar PHP ye ait olmadığı için sunucuda çalışmayacaktır bu sebeple bu kodları çift tırnak içerisinde sanki bir metin yazıyormuş gibi kullanmalısınız. Tarayıcınıza ulaştığında bu kodlar tarayıcı tarafından algılanarak çalıştırılır.

# PHP 'de Değişken Tanımlama

<?Php

```
$adi="Mutlu YAPICI";
```

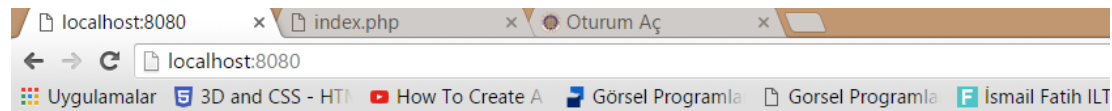
```
$yas=30;
```

```
$okul="Elmadağ Meslek Yüksekokulu";
```

```
echo "Adım : <h1>".$adi."</h1><br>". "Okul = <font  
color='red' size='12'>".$okul."</font><br>";
```

```
echo "Yaşı = ".$yas;
```

?>



Adım :

**Mutlu YAPICI**

Okul = **Elmadağ Meslek Yüksekokulu**

Yaşı = 30

# PHP 'de Değişken Tanımlama

```
<?Php
```

```
/*
```

```
  Bu ekinde  
  istediğim kadar açıklama  
  satırı  
  yazabilirim
```

```
*/
```

```
  $adi="Mutlu YAPICI"; // Açıklama satırı bu
```

```
  $yas=30;
```

```
  $okul="Elmadağ Meslek Yüksekokulu";
```

```
  ///Açıklama
```

```
  /// yazabilirim
```

```
  // Daha çok açıklama
```

```
  echo "Adım : <h1>".$adi."</h1><br>". "Okul = <font color='red' size='12'>".$okul."</font><br>";
```

```
  echo "Yaşı = ".$yas;
```

```
?>
```



Adım :

**Mutlu YAPICI**

Okul = **Elmadağ Meslek Yüksekokulu**

Yaşı = 30

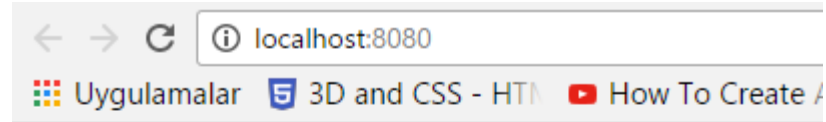
# PHP 'de Değişkenlerde İşlem

PHP de 4 işlem ve diğer matematiksel işlemler diğer programlama dillerinde olduğu gibidir. Toplama, çıkarma, çarpma bölme, mod alma işlemleri aynı şekilde gerçekleştirilir. Üs alma, karekök, trigonometri gibi işlemler içinde fonksiyonlar mevcuttur. Yine **++\$sayi** **\$sayi++** **--\$sayi** **\$sayi--** **+=** **-=** **\*** **/=** gibi işlemleri de kullanabilirsiniz.

```
<?Php
```

```
$sayi1=20; $sayi2="30";  
$sayi3=5; $sayi4=64;  
$sayi5=90; $sayi6=-190;  
Echo $sayi1+$sayi2-$sayi3;  
echo "<br>".$sayi1/$sayi3;  
echo "<br>".$sayi2*$sayi3;  
echo "<br>".$sayi4%$sayi3;  
echo "<br>".Sin($sayi5);  
echo "<br>".sqrt($sayi4);  
echo "<br>".abs($sayi6);  
echo "<br>".pow($sayi3,2);  
echo "<br>".pi();  
echo "<br>".max(3,5, 1, 56);  
echo "<br>".min(3,5, 0, 67);  
echo "<br>".floor(3.7);  
echo "<br>".ceil(3.5);  
echo "<br>".round(3.67);
```

```
?>
```



```
45  
4  
150  
4  
0.89399666360056  
8  
190  
25
```

# PHP 'de Diziler

PHP de dizi işlemleri oldukça kolaydır ve **Array** fonksiyonu ile yapılır. **Print\_r()** fonksiyonu dizinin içeriğini direk ekrana basmak için kullanılır. Dizilerin eleman sayısını **Count()** fonksiyonu ile öğrenebilirsiniz.

<?Php

```
$dizim=array(2,4,56,2,3,5);
```

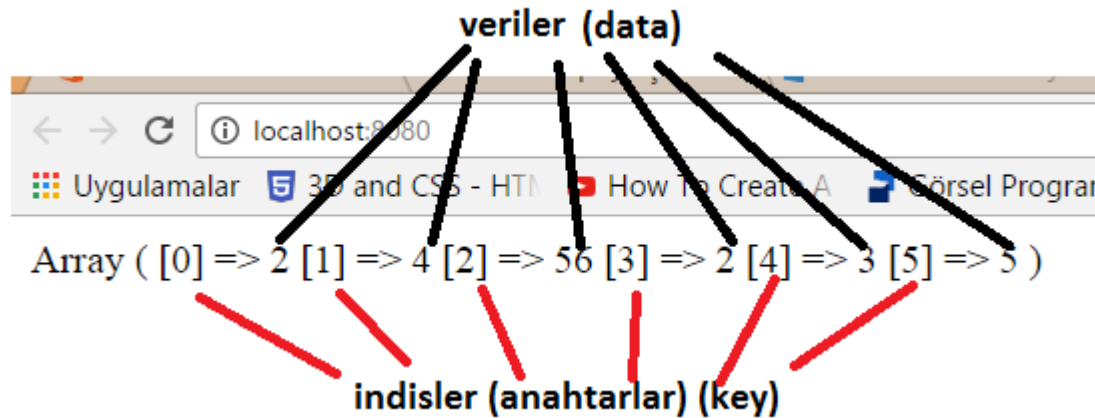
```
Echo $dizim;//ekrana ARRAY yazar
```

```
print $dizim;//ekrana ARRAY yazar
```

```
print_r($dizim);//ekrana dizinin içeriğini şablon şekilde basar
```

```
Echo count($dizim);//ekrana 6 yazar
```

?>





# PHP 'de Diziler

Dizi tanımlarken isterseniz elemanlarını string, integer, boolean yada diğer türlerden aynı anda karma şekilde tanımlayabilirsiniz.

```
<?Php
```

```
$dizim=array(2,"Ali",4.34,true,56,'C',2,"Eşşek",3,5);
```

```
Echo $dizim;//ekrana ARRAY yazar
```

```
print $dizim;//ekrana ARRAY yazar
```

```
print_r($dizim);//ekrana dizinin içeriğini şablon şekilde basar
```

```
?>
```

# PHP 'de Diziler

Dizi tanımlarken isterseniz dizilerin anahtarlarını (key) de siz tanımlayabilirsiniz.

```
<?Php
```

```
$dizim=array("can"=>2, 3=>"Ali", 0=>4.34, 'M'=>>true, 2=>56,  
4=>'C', 10=>2, "arkadasim"=>"Eşşek", 9=>3, 7=>5);
```

```
Echo $dizim["can");//ekrana 2 yazar
```

```
print $dizim["arkadasim");//ekrana Eşşek yazar
```

```
print_r($dizim);//ekrana dizinin içeriğini şablon şekilde basar
```

```
Echo count($dizim);//ekrana 10 yazar
```

```
?>
```

# PHP 'de Diziler

İsterseniz belirli bir aralıkta otomatik dizi oluşturabilirsiniz. Bunun için **Range**(başlangıç, son, adım aralığı)

```
<?Php
```

```
$dizim=range(0,100,2);
```

```
print_r($dizim);//ekrana dizinin içeriğini şablon şekilde basar
```

```
Echo count($dizim);//ekrana 51 yazar
```

```
?>
```

# PHP 'de Çok Boyutlu Diziler

PHP de çok boyutlu dizileri iç içe dizi gibi düşünün bilirsiniz ve sanki dizi içinde değer yerine başka bir dizi tanımlıyor gibi tanımlayabilirsiniz.

```
<?Php
```

```
$dizim=array(          array(1,2,3,10),  
                array(3,4,5,11,12),  
                array(6,7,8)
```

```
);
```

```
Echo $dizim[1][1];//ekrana 4 yazar
```

```
echo "<br>";
```

```
print $dizim[2][0];//ekrana 6 yazar
```

```
echo "<br>";
```

```
print_r($dizim);
```

```
echo "<br>";
```

```
print_r($dizim[1]);
```

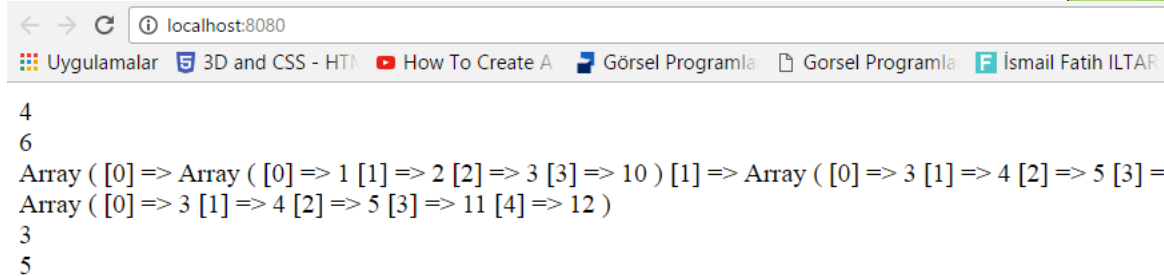
```
echo "<br>";
```

```
echo count($dizim);//ekrana 3 yazar
```

```
echo "<br>";
```

```
echo count($dizim[1]);//ekrana 5 yazar
```

```
?>
```



```
localhost:8080  
Uygulamalar 3D and CSS - HT... How To Create A Görsel Programla Gorsel Programla İsmail Fatih ILTAR  
4  
6  
Array ( [0] => Array ( [0] => 1 [1] => 2 [2] => 3 [3] => 10 ) [1] => Array ( [0] => 3 [1] => 4 [2] => 5 [3] => 11 [4] => 12 )  
3  
5
```

# Değişkenin Değişkeni

PHP de kullanılan en ilgin yapılardan biri de değişkenin değişkenidir. Değişkenin değişkeni, bir değişkenin içerisinde tutulan değer başka bir değişkenin adı olarak kullanılması mantığına dayanmaktadır.

```
<?Php
```

```
$sayi1="birsayi";
```

```
$$sayi1="Sayı içinde sayı";
```

```
Echo $sayi1;//ekrana birsayi yazar
```

```
echo "<br>";
```

```
print $birsayi;//ekrana "Sayı içinde sayı" yazar
```

```
echo "<br>";
```

```
print $$sayi1;//ekrana "Sayı içinde sayı" yazar
```

```
?>
```

# Değişkenin Referans olarak Gösterilmesi

Değişkenlerin referans olarak gösterilmesi demek birden fazla değişkenin aynı hafızayı ortak kullanması anlamına gelir. Pointer tanımlama olarakta bilinir. Dillere göre tanımlama işlemi farklılık gösterse de PHP C++ dilini taban aldığından & işareti kullanarak pointer tanımlarız. Tanımladığımız pointer ile aynı alanı tanımlı tüm değişkenlerden ortak yönetip güncelleyebiliriz.

```
<?Php
```

```
$sayi1="bir sayı";
```

```
Echo $sayi1;//ekrana birsayı yazar
```

```
$sayi2=& $sayi1; //sayi2 nin hafızası sayi1 ile aynı yapılıyor
```

```
echo "<br>";
```

```
$sayi2="içerik değişti";
```

```
echo "<br>";
```

```
Echo $sayi2;//ekrana içerik değişti yazar
```

```
echo "<br>";
```

```
Echo $sayi1;//ekrana içerik değişti yazar
```

```
?>
```

# Operatörler

- Her programlama dilinde olduğu gibi C# 'ta da işlemler yapabilmemizi sağlayan çeşitli operatörler bulunmaktadır.
- Atama, aktarma operatörleri
- Aritmetik işlem operatörleri
- Karşılaştırma operatörleri
- Mantıksal işlem operatörleri

# Atama Operatörleri

- Atama operatörleri, değişkenlere değer atamak için kullanılan sembollerdir. C# ta temel atama operatörü = dir. Diğer atama operatörleri:

Operatör	Açıklama
=	Sağdaki değeri soldaki değişkene atar.
+=	Soldakine sağdakini ekler, sonucu soldakine atar.
-=	Soldakinden sağdakini çıkarır, sonucu soldakine atar.
*=	Soldakini sağdaki ile çarpar, sonucu soldakine atar.
/=	Soldakini sağdakine böler, sonucu soldakine atar.
%=	Soldaki ile sağdakinin modula işleminin sonucunu soldakine atar.



# Atama Operatörleri

- Tüm dillerde olduğu gibi atama operatörü = (eşittir) dir. Değişkenlere değerleri = ile aktarırız. Bunun yanında hem atama hem de aritmetik işlem operatörleri de vardır.
- `$a=5;`
- `$b+=5;`
- `$c -=5;`
- `$d *=5;`
- `$e /=5;`
- `$f %=5;`

# Atama Operatörleri

İşlemlerin Sonuçlarına Göre \$a, \$b , \$c nin Değerlerini İnceleyin

\$a =10

\$b= 3

\$c=2;

\$c= \$a+ \$b

\$c= \$b--

\$a = \$c++ - -- \$b;

\$c+= \$a+ \$b

\$c=-- \$b

\$a += -- \$c + \$b++ - ++\$b;

\$c++

\$b= -\$a + \$b--

\$a += --\$c + \$b++ - ++\$d;

\$a--

\$c= \$a% \$b

\$b\*\*

\$a%= \$c

# Aritmetik İşlem Operatörleri

- Matematiksel işlemler için kullanılan sembollerdir. İşlem öncelikleri matematikteki sırayla aynıdır.

<i>Operatör</i>	<i>Açıklama</i>
+	Toplama
-	Çıkarma
*	Çarpma
/	Bölme
%	Modulo
++	1 artırma
--	1 eksiltme

# Aritmetik İşlemler

$$a = 10$$

$$b = 3$$

$$c = 2;$$

- Aşağıdaki aritmetik işlemleri gerçekleştirin
- $(a / b) : (c / d)$
- $(a + b) * c - (a - c) / 2$
- $a \% (b - c)$
- $a^2 + b^2 + c^2$
- $((a^2 - b^2) * c^2) / 2$

# PHP de Yorum Satırı Ekleme

- Yorumlar, programa etki etmeyen ancak kendimizin veya kodu inceleyen bir başkasının nerede ne yapıldığını anlamasını sağlayacak yazılardır.
- Yorumlar tek satırlık ve çok satırlık olmak üzere ikiye ayrılır
- PHP de, iki tane slash'ten (//) sonra gelen aynı satırdaki yazıları yorum sayar.

- `$a=0; //Burada a'yı tanımladık ve sıfır değerini verdik.`
- `$b=5; //Burada b'yi tanımladık ve beş değerini verdik.`

- PHP de, /\* ve \*/ arasına yazılan her şeyi yorum sayar. Dolayısıyla da çok satırlık yorumlar yazılabilir.

```
echo("deneme"); /*Burası birinci satır,  
burası ikinci satır,  
işte burada bitti.*//
```

# İlişkisel (Karşılaştırma) Operatörleri

- Programlamada iki veya daha fazla değişkeni kıyaslamak için kullanılan sembollerdir. İki farklı türü vardır birincisi **eşittir** ikincisi ise **özdeştir** şeklinde kullanılır.

Operatörler	C# 'deki karşılığı	C# 'de örneği	C#'de anlamı
<i>Eşitlik Operatörleri</i>			
=	==	<b>x == y</b>	<b>x eşittir y</b>
≠	!=	<b>x != y</b>	<b>x eşit değildir y</b>
<i>Karşılaştırma Operatörleri</i>			
>	>	<b>x &gt; y</b>	<b>x büyüktür y</b>
<	<	<b>x &lt; y</b>	<b>x küçüktür y</b>
>=	>=	<b>x &gt;= y</b>	<b>x büyüktür yada eşittir y</b>
<=	<=	<b>x &lt;= y</b>	<b>x küçüktür yada eşittir y</b>

# İlişkisel (Karşılaştırma) Eşittir

```
<?php
    $x=10;
    $y=5;

    echo var_dump($x<$y);/// false
    echo var_dump($x>$y);/// true
    echo var_dump($x<=$y);/// false
    echo var_dump($x>=$y);/// true
    echo var_dump($x == $y);/// false
    echo var_dump($x<>$y);/// true
    echo var_dump($x!=$y);/// true
```

?>

# İlişkisel (Karşılaştırma) Eşittir

PHP nin esnek yapısı sayesinde biz string bir değişkenle sayısal bir değişkenide kıyaslasak sonuç alabiliriz. Örneğin aşağıda `$x="5`; metinsel 5 değerini tutuyor `$y=5`; sayısal 5 değerini tutuyor. Eşittir karşılaştırması ile bunların aynı olduğu sonucuna varıldığını görüyoruz. Ancak bir sonraki slaytta özdeşlik karşılaştırmasında aynı sonucu almayacağız.

```
<?php
    $x="5";
    $y=5;
    // biri string olmasına rağmen kıyaslama sonucu doğru çıktı
    echo var_dump($x<$y);/// false
    echo var_dump($x>$y);/// false
    echo var_dump($x<=$y);/// true
    echo var_dump($x>=$y);/// true
    echo var_dump($x == $y);/// true
    echo var_dump($x<>$y);/// false
    echo var_dump($x!=$y);/// false
?>
```



# İlişkisel (Karşılaştırma) Özdeşir

PHP de özdeşir karşılaştırması eğer değişkenlerin **türleri de aynı** ise true sonucunu verir. Bir önceki sunumda türler farklı olmasına rağmen eşittir karşılaştırması true değerini vermişti.

```
1 <?php
2     $x="5";
3     $y=5;
4     // Özdeşir karşılaştırması değişkenlerin türleri de aynı
5     // olursa TRUE verir
6     echo var_dump($x === $y);/// false
7     echo var_dump($x !== $y);/// true
8     ?>
```

```
<?php
    $x="5";
    $y=5;
    // Eşittir karşılaştırması değişkenlerin türlerine BAKMAZ
    echo var_dump($x == $y);/// true
    echo var_dump($x != $y);/// false
    ?>
```

# Mantıksal Operatörler

- Bu operatörler true veya false sabit ya da değişkenleri mantıksal ve, veya, değil işlemine sokarlar. Bunlar && (ve), || (veya), XOR ve ! (değil) operatörleridir.

Operatör	Türü	C# teki kullanımı	Örnek
&& (VE)	Mantıksal	&&	$\$a \ \&\& \ \$b = \text{true/false}$
AND	Mantıksal	AND	$\$a \ \text{AND} \ \$b = \text{true/false}$
(Veya)	Mantıksal		$\$a \    \ \$b = \text{true/false}$
OR	Mantıksal	OR	$\$a \ \text{OR} \ \$b = \text{true/false}$
! (Değil)	Mantıksal	!	$! \ \$b = \text{true/false}$
XOR (Özel Veya)	Mantıksal	XOR	$\$a \ \text{XOR} \ \$b = \text{true/false}$

# Mantıksal Operatörler

```
<?php
    $x=6;
    $y=5;
    $z=10;
    echo var_dump($x == $y && $z > $y);/// false
    echo var_dump($x == $y || $z > $y);/// true
    echo var_dump($x == $y AND $z > $y);/// false
    echo var_dump($x == $y OR $z > $y);/// true

    echo var_dump(!($x == $y && $z > $y));/// True
    echo var_dump($x == $y || !($z > $y));/// false

    echo var_dump($x == $y XOR $z > $y);/// true
    echo var_dump($x > $y XOR $z > $y);/// false
?>
```

# Bitsel Operatörler

- Bu operatörler bitsel karşılaştırma ve işlem için kullanılan operatörlerdir. . Bunlar & (ve), | (veya) ve ! (değil) operatörleridir.
- Bitsel operatörler sayıların kendisi yerine sayıların bitleriyle ilgilenirler. Diğer bir deyişle sayıları ikilik sisteme dönüştürüp öyle işlem yaparlar.
- Bu operatörler yalnızca tam sayı sabit, değişken ya da ifadelerle kullanılabilirler.
- Eğer bitsel operatörler **bool** türünden değişken, sabit ya da ifadelerle kullanılırsa mantıksal operatörlerin gördüğü işin aynısını görürler

# Bitsel Operatörler

Operatör	Türü	C# teki kullanımı	Örnek
& (Bitsel VE)	Bitsel	&	\$a & \$b
(Bitsel Veya)	Bitsel		\$a   \$b
~ (Bitsel Değil )	Bitsel	~	\$a ~ \$b
^ (Bitsel Özel Veya)	Bitsel	^	\$a ^ \$b

```

<?php
$a=6;
$b=5;

echo $a & $b; /// 110 & 101 =100 (4)
echo $a | $b; /// 110 | 101 =111 (7)
echo $c= ~$b; /// 110 & 101 =100 (-6)
echo $c= ~$a; /// 110 & 101 =100 (-7)
echo $a ^ $b; /// 110 & 101 =011 (3)

```

# Bitisel Kaydırma ( Shift )Operatörleri

Operatör	Türü	C# teki kullanımı	Örnek
<< (Sola Kaydırma)	Bitisel	<<	\$a << 1 (1 bit)
>> (Sağa Kaydırma)	Bitisel	>>	\$a >>2 (2 bit)

```
<?php
$a=6;
$b=5;
//kaydırma operatörleri
echo $a>>2;/// 110>>001 (1)
echo $b>>1;/// 101 >> 010 (2)
echo $b<<2;/// 101 << 10100 (20)
echo $a<<1;/// 110 << 1100 (12)
?>
```

# Metin Operatörü

- PHP de metinleri birbirlerine bağlama için . (nokta) operatörü kullanılmaktadır. Hatırlarsanız C# ta + ile bağlıyorduk.

```
<?php
    $a=6;
    $b=" Merhaba Millet";
    $c= "Ne Mutlu";
    $x ="Türküm Diyene";

    Echo "yaşım = ". $a; /// yaşım = 6
    echo $b." ".$c;// Merhaba Millet Ne Mutlu
    echo $c." , ".$x;// Ne Mutlu , Türküm Diyene
?>
```

# İS Operatörü

- Verilen değişken, sabit ya da ifadenin **türünü kontrol** eder. Eğer söz konusu değişken, sabit ya da ifade verilen türle uyumluysa **true** değilse **false** değeri üretir. Eğer söz konusu değişken, sabit ya da ifadenin türü her zaman true ya da false üretiliyorsa derleyici uyarı verir, ancak bu uyarı derlemeye engel değildir.



# is Operatörü

```
<?php
    $a=6;
    $b=5.0;
    $c=true;
    $x;
    //kayıdırma operatörleri
    echo var_dump(is_int($a));///true
    echo var_dump(is_bool($c));///true
    echo var_dump(is_double($b));///true
    echo var_dump(is_array($a));///false
    echo var_dump(is_string($a));///false
    echo var_dump(isset($x));//false
```

```
?>
```

# Karar Kontrol Yapıları

Program yazarken bazı noktalarda belirli koşullar altında gerçekleşmesini istenilen durumlar olabilir. Bu bölümde anlatılan **if-else** ve **switch-case** deyimleri ile bu tür kapsamlı programlar geliştirilebilir.

Genel anlamda programlama dilinde kullanılan koşul yapıları iki çeşittir.

Bunlar;

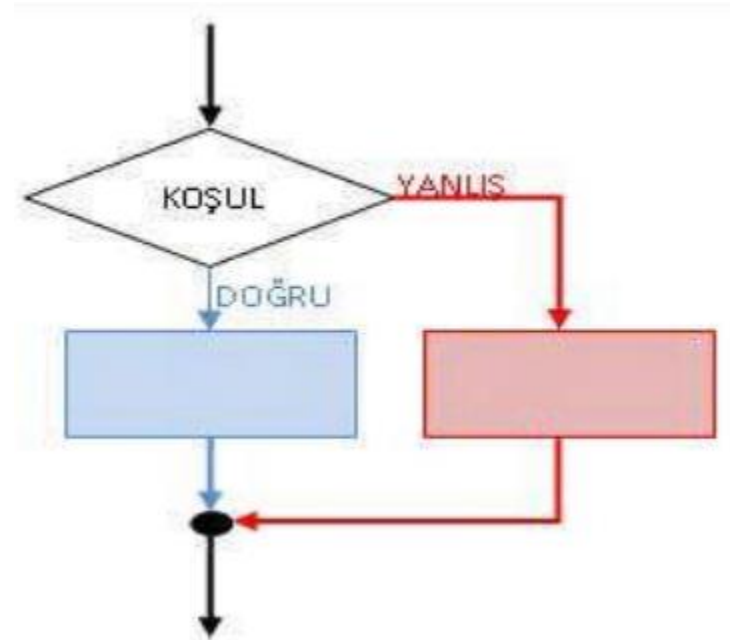
- if-else deyimi
- switch-case deyimi dir.

# If - Else Yapısı

If deyimi bir programın akışını kontrol etmek için kullanılır. Belirli bir şarta göre yapılması istenilen işlemler, If-Else deyimi kullanılarak gerçekleştirilir. If-Else deyiminin kullanımı ve akış diyagramları ile gösterimi ise şu şekildedir.

Kullanımı:

```
if(koşul)
{
    Koşul doğruysa
    yapılacak işlemler;
}else
{
    Koşul yanlışsa
    yapılacak işlemler;
}
```



# Program Örnekleri

Klavyeden girilen sayının 50 den büyük olup olmadığını bulan programı yazınız? Büyükse 50 den büyük " Küçükse "50 den Küçük" yazacak.

**Çözüm:**

# Program Örnekleri

**Klavyeden girilen bir sayının tek mi, çift mi olduğunu bulan c# kodlarını yazınız?**

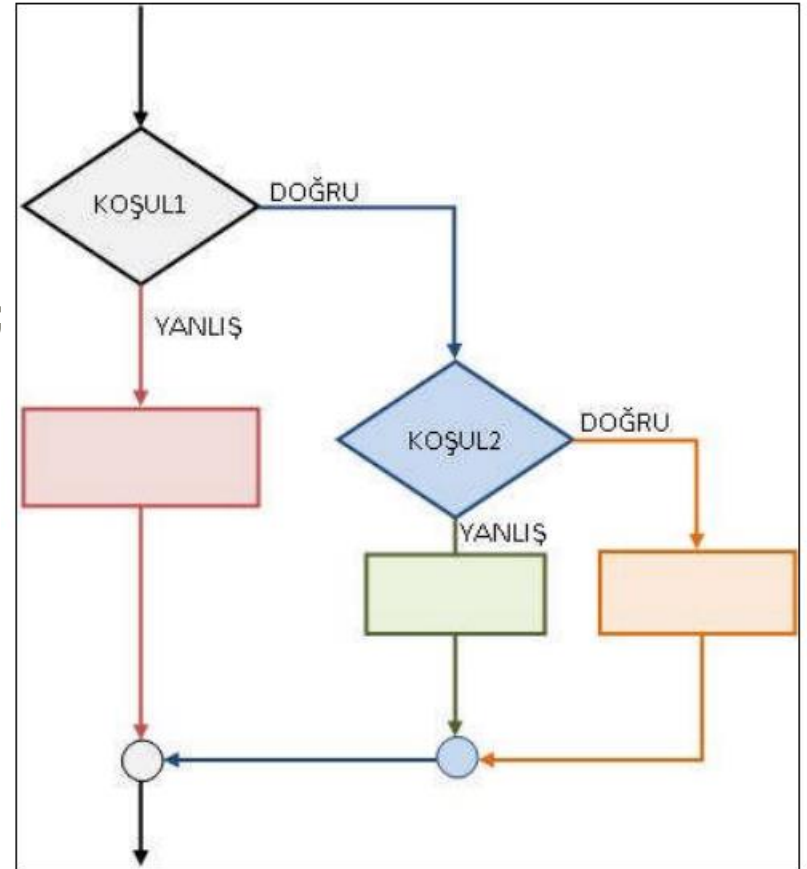
**Çözüm:**

# İç İçe İF-ELSE Yapısı

Örneğin bir koşulun sağlanması durumunda başka koşullara göre işlem yapılması istenilen durumlarda yine iç-içe If ifadeleri kullanılırlar. Bu durumdaki iç-içe If ifadelerinin kullanımları ve akış diyagramlarıyla gösterimi şu şekildedir;

Kullanım Şekli:

```
if(koşul1)
{
    if(koşul2)
    {
        koşul2 doğruysa yapılacak işlemler;
    }
    else
    {
        koşul2 yanlışsa yapılacak işlemler;
    }
}
else
{
    koşul1 yanlışsa yapılacak işlemler;
}
```



Şekil 0-2. İç-içe If İfadesi

# Program Örnekleri

Bir öğrencinin klavyeden girilen vize, final notlarına göre not ortalamasını bulan (**Not Ortalaması =vize'nin %40'ı ile final' in %60'nın toplamıdır**) ve not ortalaması **50 ve üzeri ise Ekrana "Bu Öğrenci GEÇTİ"**, değilse **"Bu Öğrenci KALDI"** yazan PHP program kodlarını yazınız?

Çözüm:

# Program Örnekleri

**Klavyeden alınacak bir tam sayının, 18 ile 65 arasında olup olmadığını kontrol eden bir program yazınız**

**Çözüm:**



# Program Örnekleri

**Klavyeden Girilen Bir Sayı Sıfır mı, Pozitif bir sayı mı, negatif bir sayı olup olmadığını bulan programın PHP kodlarını yazınız?**

**Çözüm:**

# Program Örnekleri

Girilen kenar uzunluklarına göre bir üçgenin türü hakkında bilgi veren PHP programını yazınız(Mantıksal Operatörler ile)

**Çözüm:**

# Program Örnekleri

Klavyeden 2 adet notu girilen bir öğrencinin notu 0-24 arası ise “Not Sıfır”, 25-44 arası ise “Not Birdir”, 45-54 arası ise “Not İkidir”, 56-69 arası ise “Not Üçtür”, 70-84 arası ise “Not Dörttür”, 85-100 arası ise “Not Beştir” şeklinde notu harflendiren PHP kodları yazınız?

**Çözüm:**

# Program Örnekleri

Klavyeden girilen saat bilgisine göre;

->Saat negatif bir sayı girildiğinde “Girdiğiniz Saat Negatif Olamaz”

->Saat 05-12 arasında ise “Günaydın”

->Saat 12-16 arasında ise “İyi Günler”

->Saat 16-24 arasında ise “İyi Akşamlar”

->Saat 24-05 arasında ise “İyi Geceler”

Mesajı veren PHP kodlarını yazınız?

Çözüm:

# ?: Operatörü

- ?: operatörü C#'ta olfduğu gibi PHP'de de üç operand alan tek operatördür. Verilen koşula göre verilen değerlerden (object türünden) birini tutar. Kullanımı şöyledir:
- koşul?doğruysa\_değer:yanlışsa\_değer
- `echo $sonuc=1453>120?"Büyük":"Küçük";` // Büyük

# ?: Operatörü

- Girilen sayı 50 den küçük ise b değişkenine 10 değilse b değişkenine 20 değerini atayan ?: operatörü ile yazılmış program

## **Break Yapısı**

**Programın çalışması esnasında kullanıldığı noktadan itibaren işlemin bırakılmasını sağlar. Karar ve kontrol yapılarında kullanırsa kontrol işlemi bırakılır. Döngü yapılarında kullanılırsa döngü işlemi sonlandırılır.**

**Örnek;**

# Switch-Case Yapısı

Bu deyim bir değişkenin içeriğine bakarak, programın akışını bir çok seçenekten birine yönlendirir. case (durum) deyiminden sonra değişkenin durumu belirlenir ve takip eden gelen satırlar (deyimler) işleme konur. Bütün durumların aksi söz konu olduğunda gerçekleştirilmesi istenen deyimler default deyiminden sonraki kısımda bildirilir. Switch-Case deyimi genellikle karmaşık if-else bloklarının yerine, daha okunabilir oldukları için tercih edilmektedir.

Genel yazım biçimi:

```
switch(değişken) {  
    case sabit1:  
        ...  
        deyimler;  
        ...  
    Break;  
    case sabit2:  
        ...  
        deyimler;  
        ...  
    Break;  
    case sabitn:  
        ...  
        deyimler;  
        ...  
    Break;  
    default:  
        ... hata deyimleri veya varsayılan deyimler; .  
    Break;  
}
```



# Switch-Case Yapısı

Switch-Case yapısının çalışmasına bir göz atalım;

- Önce switch parantezleri içerisindeki ifade hesaplanır.
- Programın akışı, hesaplanan ifade ile aynı case sabitinin bulunduğu satıra gelir.
- Eğer hesaplanan ifade, mevcut case sabitlerinden herhangi birisi ile eşleşmiyorsa **default** anahtar sözcüğünün bulunduğu yere gelir ve program buradan devam eder.

Her case satırı içerisindeki işlemlerimiz tamamlandıktan sonra, ilgili case satırının sonuna geldiğimizi belirtmek için **break** komutu kullanılır.

# Switch-Case Yapısı

## Switch-Case Yapısı İle İlgili Önemli Kurallar:

- Case anahtar sözcüğünün yanındaki ifadeler sabit olmak zorundadırlar. Bu ifadeler içerisinde değişken bulunamaz.
- Case ifadeleri herhangi bir tam sayı sabiti, karakter veya string sabiti olabilir.
- Default durumunu istediğimiz yere yazabiliriz. Aynı şekilde case ifadelerini de istediğimiz sırada yazabiliriz.
- Bir switch bloğunda iki veya daha fazla sayıda aynı değere sahip case ifadesi bulunamaz.
- Bir switch bloğunda default case olmak zorunda değildir.
- Akış herhangi bir case ifadesine geldiğinde, akış farklı bir case ifadesine yönlendirilmek istenirse goto anahtar sözcüğü kullanılır. Örnek **goto case 'elma'**

# Switch-Case Yapısı Örnek

Klavyeden girilen 1-12 arasındaki sayı değerine göre o sıradaki ayın ismini veren programı yazınız

**Çözüm:**

```
$ay=5;
switch ($ay)
{
case 1:
echo("$ay. ay OCAK ayıdır.");
break;
case 2:
echo("$ay. ay ŞUBAT ayıdır.");
break;
case 3:
echo("$ay. ay MART ayıdır.");
break;
case 4:
echo("$ay. ay NİSAN ayıdır.");
break;
case 5:
echo("$ay. ay MAYIS ayıdır.");
break;
```

# Switch-Case Yapısı Örnek

**Klavyeden girilen değer ile seçimi yapılan şeklin alanını veya çevresini bulan programı yazınız**

# Switch-Case Yapısı Örnek

Klavyeden sırasıyla birinci sayının , işlem türünün ve ikinci sayının girilmesini isteyen ve işlem türüne göre işlemi yapıp sonucu ekrana yazdıran programı yazınız.

İşlemler :

- Toplama
- Çıkartma
- Çarpma
- Bölme
- Mod

# Goto Yapısı

Program içerisinde her hangi bir adıma gidilmesi veya tanımlı bir yere dönülmesi için kullanılır. Kullanabilmek için gidilecek yer etiketlenerek belirlenmelidir.

**<?Php**

**goto** a;

**echo** 'EMYO';

a:

**echo** 'ANKARA';

**?>**

# Goto Yapısı Örnek

**Program içerisinde her hangi bir adıma gidilmesi veya tanımlı bir yere dönülmesi için kullanılır. Kullanabilmek için gidilecek yer etiketlenerek belirlenmelidir.**

**Örnek : Klavyeden 1- ile 10 arasında sayı girildiğinde sayının adını yazan, 10 dan büyük sayılar için sayının 1 ile 10 arasında olmasını isteyerek tekrar girilmesini isteyen programı switch-case ve goto deyimini ile yazınız. (Duruma uygun 5 sayı girildiğinde işlemi sonlandırın)**

# KAYNAKLAR

- İnternet ortamı
- PHP ve AJAX Haydar TUNA
- A'dan Z'ye PHP Rıza ÇELİK

