

**T.C.
MİLLÎ EĞİTİM BAKANLIĞI**

DENİZCİLİK

İNTERNET PROGRAMCILIĞI 2

ANKARA 2013

- Bu modül, mesleki ve teknik eğitim okul/kurumlarında uygulanan Çerçeve Öğretim Programlarında yer alan yeterlikleri kazandırmaya yönelik olarak öğrencilere rehberlik etmek amacıyla hazırlanmış bireysel öğrenme materyalidir.
- Millî Eğitim Bakanlığınca ücretsiz olarak verilmiştir.
- **PARA İLE SATILMAZ.**

İÇİNDEKİLER

AÇIKLAMALAR	ii
GİRİŞ	1
ÖĞRENME FAALİYETİ 1	3
1. DEĞİŞKENLER	3
1.1. Değişken Tipleri.....	4
1.2. Sabitler	7
1.3. Operatörler	8
UYGULAMA FAALİYETİ	12
ÖLÇME VE DEĞERLENDİRME	13
ÖĞRENME FAALİYETİ 2	14
2. PROGRAM DENETİMİ	14
2.1 Kontrol Yapıları	14
2.2 Döngüler	19
UYGULAMA FAALİYETİ	22
ÖLÇME VE DEĞERLENDİRME	23
MODÜL DEĞERLENDİRME	24
CEVAP ANAHTARLARI	27
KAYNAKÇA	28

AÇIKLAMALAR

ALAN	Denizcilik
DAL/MESLEK	
MODÜLÜN ADI	İnternet Programcılığı 2
MODÜLÜN TANIMI	Web programlamada değişken kullanabilme ve program denetimi yapabilme ile ilgili öğrenme materyalidir.
SÜRE	40/32
ÖN KOŞUL	İnternet Programcılığı 1 modülünü başarmış olmak
YETERLİK	Programlama dilinin komut yapısını tanıyarak temel programlama elemanlarını kullanmak ve programlama dilinin komut yapısını tanıyarak temel programlama elemanlarını kullanmak
MODÜLÜN AMACI	Genel Amaç Öğrenci bu modül ile gerekli ortam sağlandığında programlama komutları yardımıyla değişken ve denetim uygulamaları hazırlayabileceksiniz. Amaçlar <ol style="list-style-type: none">1. Programlama içinde değişken kullanımını öğrenerek, uygulamalar yapabileceksiniz.2. Programlama içinde değişken kullanımını öğrenerek, uygulamalar yapabilecek, program denetimi işlemlerini gerçekleştirebileceksiniz.
EĞİTİM ÖĞRETİM ORTAMLARI VE DONANIMLARI	Web programlama yazılımlarını çalıştırabilecek yeterlikte bilgisayar, internet bağlantısı
ÖLÇME VE DEĞERLENDİRME	Her faaliyet sonrasında o faaliyetle ilgili değerlendirme soruları ile kendi kendinizi değerlendireceksiniz. Öğretmen modül sonunda size ölçme aracı (uygulama, soru-cevap) uygulayarak modül uygulamaları ile kazandığınız bilgi ve becerileri ölçerek değerlendirecektir.

GİRİŞ

Sevgili Öğrenci

İnternet Programcılığı 1 modülünde PHP dilinin kullanılması için gereken yazılımların kurulumlarını öğrendiniz.

Programınızın her adımını, değişkenlerini, döngülerini ve hatalarını kontrol etmek programınızın kalitesini ve kullanılabilirliğini artırır. Değişkenler, programların temel taşlarıdır. Yeri geldiği zaman bu taşların içeriği değişmekte ve program içinde farklı anlamlar kazanmaktadır. If deyimi, switch deyimi, for döngüsü gibi yapıları, programınızı kontrol eden mekanizmalar olarak düşünebilirsiniz.

Unutmayınız ki yapabileceğiniz programlar, sizin hayal gücünüzle sınırlıdır. Hayal gücünüzdekileri gerçekleştirebilmeniz için programlarınızda kullanacağınız değişken, döngü ve kontrol yapılarını bilmek zorundasınız.

Bu modülde PHP dilinde kullanılan değişkenler ve kontrol yapıları anlatılacaktır. Bu modülü tamamladığınızda PHP içinde kullanılan değişkenler, döngüler ve kontrol yapılarını öğrenecek ve programlarınızda kolay bir şekilde kullanabileceksiniz.

ÖĞRENME FAALİYETİ-1

AMAÇ

Programlama içinde değişken kullanımını öğrenerek uygulamalar yapabileceksiniz.

ARAŞTIRMA

- Değişkenler program içinde hangi amaçla kullanılabilir, düşününüz.
- Program içinde sabit kullanılması hangi durumlarda gerekebilir, araştırınız.
- Hangi türlerde değişken tipleri vardır arkadaşlarımızla tartışınız.

1. DEĞİŞKENLER

Değişkenler, verileri bellekte tutmak için kullanılır. Her tanımlanan değişken, bellekte belli bir yer ayrılmasına neden olur. Değişkenin içerisine programın herhangi bir yerinde farklı değerler atanabilir. İçerisinde değer bulunduran bir değişkene yeni bir değer atandığında eski değer kaybolur, yerine bu yeni değer tutulur. 'Değişken' adını da bu yüzden alır.

Örnek olarak boş bir bardak düşünün. İçindeki madde su, meyve suyu veya kola olabilir. Önemli olan bardağın kendisidir. İçindeki ise bardağın içerdiği maddedir. Yani bardağı bir değişken içindeki maddeyi de değişkenin değeri olarak düşünebiliriz.



Değişkenler tanımlanırken değişken isminin başına "\$" (dolar işareti) konulur. Değişkenler (ç, ğ, ü, ö, ş, İ gibi Türkçe karakterler hariç) bir harf ya da '_' işareti ile başlayabilir. Değişken adı içerisinde rakam bulunabilir fakat değişken adı rakamla başlayamaz. Ayrıca değişken içerisinde '_' işaretinden başka işaret kullanılmaz.

Değişkenlere atama yapılırken daha sonra göreceğiniz atama operatörü olan "=" (eşittir) işareti kullanılır. Değişken tipi belirlemeden değişkene direkt olarak değer atayabiliriz.

➤ Doğru Tanımlama Örnekleri

- \$sayi_1 = 453;

- \$isim = "Hasan";
- \$boy = 1.78;

➤ Yanlış Tanımlama Örnekleri

- \$1sayi= 453; (değişkenler rakamla başlayamaz)
- \$isim% = "Hasan"; (değişkenler içinde '_' işaretinden başka işaret kullanılmaz)
- \$maaş=1000; //'ş' (Türkçe karakter kullanılamaz)
- \$Soyad=Ünlü (değişken içine Türkçe karakter atanabilir fakat bütün karakter değişkenler çift tırnak içine yazılmalıdır. "Ünlü" gibi.)

1.1. Değişken Tipleri

PHP'de de diğer dillerde olduğu gibi değişken tipleri (veri türü) bulunmaktadır. Tutulacak verilerin özelliğine (türü, boyutu gibi) göre, kullanıcı veri türlerinden birini seçer. Bu veri türlerini sırayla inceleyelim.

➤ Tam Sayı (Integer) Veri Türü

Negatif veya pozitif ondalık kısmı bulunmayan sayılardır. 34, 567, 2343, -65, 56306 gibi sayılar örnek verilebilir.

Örnek Kod	Ekran Çıktısı
<code><body> <? \$i=65; echo (\$i); ?>
 <? \$x=-345; echo (\$x); ?>
 <? \$y=937278; echo (\$y); ?> </body></code>	65 -345 937278

➤ Kayan Noktalı (Real) Veri Türü

Ondalıklı kısmı bulunan sayılardır. Örneğin 564.4, -23.9 gibi sayılardır. Bu veri türüne değer atanırken sayının tam kısmı ve ondalık kısmı arasına "." (nokta işareti) konulur.

Örnek Kod	Ekran Çıktısı
<code><body> <? \$i=65.6; echo (\$i); ?>
 <? \$x=-345.76; echo (\$x); ?>
 <? \$y=937278.94;echo(\$y); ?> </body></code>	65.6 -345.76 937278.94

➤ Karakter Dizileri (String) Veri Türü

İsminden de anlaşılacağı gibi karakterlerden (harf, rakam, işaret) oluşan veri türüdür. Örneğin “internet”, “Sayı 11/102#”, “30 Nisan 2007” ifadeleri birer string örneğidir. String veri türleri çift tırnak (“) ve tek tırnak (‘) olmak üzere iki şekilde tanımlanmaktadır. Bu iki tanımlamanın arasındaki fark çok önemlidir.

- **Çift Tırnak ile Tanımlanan Karakter Dizileri**

Bu tanımlamada çift tırnak içinde başka bir değişken varsa o değişkenin değeri aktarılır.

Örnek Kod	Ekran Çıktısı
<pre><body> <? \$ad="Bilişim"; \$sonuc="\$ad Teknolojileri"; echo(\$sonuc); ?> </body></pre>	Bilişim Teknolojileri

- **Tek Tırnak ile Tanımlanan Karakter Dizileri**

Bu tanımlamada tek tırnak içinde başka bir değişken varsa o değişkenin değeri alınmaz. Değişken adı \$ işareti ile birlikte aktarılır.

Örnek Kod	Ekran Çıktısı
<pre><body> <? \$ad="Bilişim"; \$sonuc='\$ad Teknolojileri'; echo(\$sonuc); ?> </body></pre>	\$ad Teknolojileri

- **Dizi Tipindeki Değişkenler**

Hazırlanmış fonksiyonlarla kullanıldıklarında birçok işlem kolayca yapılabilir. Bu fonksiyonları ileriki konularda göreceksiniz. Dizi tipindeki değişkenlerin nasıl tanımlandığını inceleyelim.

- **\$dizi[anahtar] = değer ;**

Dikkat: Dizi oluştururken kullanılan anahtar nümerik (sayısal) ve karakter dizisi olabilir. Fakat negatif değer alamaz.

Örnek Kod
<pre><? \$gun[0]= "Pazartesi"; \$gun[1]= "Salı"; \$gun[2]= "Çarşamba"; \$gun[3]= "Perşembe"; \$gun[4]= "Cuma"; ?></pre>

Yukarıdaki örnekte tek boyutlu dizi tanımlaması yapılmıştır. Tek boyutlu dizilerin yanı sıra iki, üç veya daha fazla boyutta diziler tanımlamak da mümkündür.

İki Boyutlu Dizi Tanımlaması
<pre><? \$araba ["BMW"] ["renk"] = "kırmızı"; \$araba ["BMW"] ["model"] = 2007; \$araba ["Fiat"] ["renk"] = "beyaz"; \$araba ["Fiat"] ["model"] = 2005; ?></pre>

Dizilerde kullanılan anahtarlar karakter dizisi kullanıldığında mutlaka çift tırnak içine alınmalıdır. `$araba [BMW] [renk] = "kırmızı";` tanımlaması yanlış bir tanımlama olur.

➤ Nesnelere (Objects)

Herhangi bir sınıf (class) yazıldıktan sonra o sınıfa ait tanımlanan değişkenlerdir. Tanımlandıkları sınıfların özellik ve fonksiyonlarına bu değişkenler üzerinden ulaşılabilir.

Örnek Kod	Ekran Çıktısı
<pre><? class araba { function mesaj () { echo "motorlu taşıt";} } \$mercedes = new araba; \$mercedes->mesaj(); ?></pre>	<p>motorlu taşıt</p>

Yukarıdaki programda öncelikle araba adında bir sınıf (class) tanımlaması yapılmıştır. Bu sınıf mesaj adında bir fonksiyona sahiptir. Bu fonksiyonla ekrana "motorlu taşıt" cümlesi yazdırılabilir. Sınıf tanımlaması bittikten sonra `$mercedes = new araba;` satırı ile araba sınıfı türünden \$mercedes isimli yeni bir nesne (object) değişkeni tanımlanmıştır. Bu aşamadan sonra \$mercedes isimli nesnemiz araba sınıfının sahip olduğu bütün özellik ve fonksiyonları kullanabilecektir. Nitekim `$mercedes->mesaj();` satırı ile mesaj fonksiyonu çalıştırılmıştır.

➤ Mantıksal (Boolean) Değişkenler

Bazı kaynaklara göre, gerçekte PHP'de mantıksal değişken türü yoktur. Ancak diğer PHP değişkenleri mantıksal değişkene çevrilebilir. Böylece geriye 1 yani true (doğru) veya 0 yani false (yanlış) ya da null (boş) olarak değer döndürülür. Aşağıdaki tabloda veri türlerinin mantıksal türe dönüştürüldüğü zaman ortaya çıkan sonuçlar gösterilmiştir.

Mantıksal tipe dönüştürülen değişken	Mantıksal sonucu
String (içi boş ise)	0 veya null(boş)
String (içi dolu ise)	1
Integer (içi sıfır ise)	0 veya null(boş)
Integer (negatif veya pozitif ise)	1
Double(içi sıfır ise)	0 veya null(boş)
Double (negatif veya pozitif ise)	1

Not: Değişkenler arasında tür ataması settype() fonksiyonu yardımıyla gerçekleştirilebilir. Değişkenin türünü öğrenmek için ise gettype() fonksiyonu kullanabilirsiniz.

Örnek Kod	Ekran Çıktısı
<pre><? \$i=4543; Settype(\$i,'boolean'); echo(\$i); ?>
 <? \$sayi=0; Settype(\$sayi,'boolean'); echo(\$sayi); ?>
 <? \$s="ali"; Settype(\$s,'boolean'); echo(\$s); ?></pre>	<pre>1 1</pre> <div style="border: 1px solid black; padding: 5px; margin-top: 10px;"><p>2. satırda döndürülen mantıksal değer null (boş) olduğu için bu satırda boşluk yazılacaktır</p></div>

1.2. Sabitler

Sabitler önceden tanımlanır ve değer atanır ve program içinde değeri değiştirilemez. Diğer tiplerdeki değişkenlere programın her yerinde istenilen değer atanabilir. Fakat sabit bir değere yeni bir değer ataması yapılırsa hata alınır.

Sabitlerin kullanım amacı programda herhangi bir değerın değişmesini engellemektir. Örneğin web sayfamızın boyutlarını sabit değerde tutmak istiyor, kullanıcının değiştirmesini

istemiyorsak sabitleri kullanabiliriz. Ya da herhangi bir matematiksel problem içinde değeri 3.14 olan pi sabitini kullanabiliriz. Aşağıda sabitlerin nasıl tanımlandığı gösterilmiştir.

Örnek Tanımlama
Define ("sabitin ismi", değeri); Define ("pi",3.14);

Örnek Kod	Ekran Çıktısı
<? Define ("pi",3.14); echo(pi); ?>	3.14

Hatalı Kod
<? Define ("pi",3.14); pi=54; echo(pi); ?>

Dikkat: Sabitler kullanılırken "\$" işareti kullanılmaz.

1.3. Operatörler

Operatörler, değerler ve değişkenler üzerinde işlem yaparak onları düzenlememizi sağlayan özel karakterlerdir. Bu karakterlerin neler olduğunu ve nasıl kullanıldıklarını sırayla inceleyelim

➤ Atama Operatörü

Atama operatörü "=" (eşittir işareti) dir. Program içerisinde herhangi bir değişkene herhangi bir değeri atamak amacıyla kullanılır. Atama işlemi sağdan sola doğru gerçekleştirilir.

Örnek Kod	Açıklama
\$isim="Ali";	\$isim adlı değişkene Ali karakter dizisi aktarılıyor
\$sayı=249;	\$sayı değişkenine 249 sayısı aktarılıyor

➤ Aritmetik Operatörler

Temel matematiksel işlemlerin yapılmasında kullanılan operatörlerdir.

Operatör	Anlamı	Kullanımı
+	Toplama	$x+y$
-	Çıkarma	$x-4$
*	Çarpma	$-2*x$
/	Bölme	$10/2$
%	Mod (Kalan)	$a\%4$

Örnek olarak 10 ve 2 sayılarını birer değişkenlere atayıp aralarında toplama, çıkarma, çarpma, bölme ve kalan bulma işlemlerini yapan PHP kodunu yazalım.

```
<?PHP
```

```
$sayi1=10; //birinci sayımıza 10 değerini aktardık.  
$sayi2=2; //ikinci sayımıza 2 değerini aktardık.  
$toplam = $sayi1+$sayi2; //birinci ve ikinci sayımızı topladık ve sonucu $toplam değişkenine  
//aktardık.  
$fark = $sayi1-$sayi2; //Çıkardık ve sonucu $fark değişkenine aktardık.  
$carpim = $sayi1*$sayi2; //çarptık ve sonucu $carpim değişkenine aktardık.  
$bolum = $sayi1/$sayi2; // böldük ve sonucu $bolum değişkenine aktardık.  
$bolum_kalan = $sayi1%$sayi2; // birinci sayımızın ikinci sayımıza bölümünden  
//kalanı bulduk ve sonucu $bolum_kalan değişkenine aktardık.  
echo $toplam; // toplamlarını yazdırdık.  
echo "<BR>"; // bir alt satıra geçtik.  
echo $fark; // farklarını yazdırdık.  
echo "<BR>"; // bir alt satıra geçtik.  
echo $carpim; // çarpımlarını yazdırdık.  
echo "<BR>"; // bir alt satıra geçtik.  
echo $bolum; //bölümlerini yazdırdık.  
echo "<BR>"; // bir alt satıra geçtik.  
echo $bolum_kalan; // bölümden kalanı yazdırdık.  
?>
```

Ekran çıktımız ise aşağıdaki gibi olacaktır.

```
12  
8  
20  
5  
0
```

➤ String Operatörleri

String, birleştirme amaçlı olarak kullanılan operatör “.” (nokta) operatörüdür.

Örnek Kod	Ekran Çıktısı
<pre><? \$ad="Ali"; \$soyad="MERT"; \$adsoyad=\$ad." ".\$soyad; echo \$adsoyad; ?></pre>	Ali MERT

➤ Bileşik Atama Operatörleri

İki değişken arasında yapılan işlem ve atamaları tek operatörle gerçekleştirmek için kullanılır.

Operatör	Kullanımı	Eşdeğeri
+=	\$a += \$b	\$a = \$a + \$b
-=	\$a -= \$b	\$a = \$a - \$b
*=	\$a *= \$b	\$a = \$a * \$b
/=	\$a /= \$b	\$a = \$a / \$b
.=	\$ad .= \$soyad	\$ad = \$ad.\$soyad

➤ Artırma ve Azaltma Operatörleri

Artırma ve azaltma operatörlerini iki farklı şekilde kullanabiliriz.

- **Önden Artırma veya Azaltma**

++\$a veya --\$a şeklinde kullanılır. Herhangi bir işlem yapılmadan önce değişkenin değeri bir artırılır veya azaltılır.

Örnek Kod	Ekran Çıktısı
<pre><? \$a=10; echo ++\$a; ?></pre>	11

- **Arkadan Artırma veya Azaltma**

++\$a veya --\$a şeklinde kullanılır. Herhangi bir işlem yapıldıktan sonra değişkenin değeri bir artırılır veya azaltılır.

Örnek Kod	Ekran Çıktısı
<? \$a=10; echo \$a++; ?>	10

Bu işlemde \$a değişkeninin ilk değeri 10 olduğu için önce ekrana 10 sayısı yazılır bu işlemden sonra \$a değişkeninin değeri 1 artırılarak 11 olur.

- **Karşılaştırma Operatörü**

Operatör	Anlamı	Kullanışı
==	eşittir	\$a == \$b
===	denktir	\$a === \$b
!=	eşit değil	\$a != \$b
<>	eşit değil	\$a <> \$b
<	küçüktür	\$a < \$b
>	büyüktür	\$a > \$b
<=	küçük veya eşit	\$a <= \$b
>=	büyük veya eşit	\$a >= \$b

- **Mantıksal Operatörler**

Mantıksal operatörler, çoklu mantıksal sorgular yapmakta kullanılır.

Operatör	Adı	Kullanışı	Sonuç
!	NOT	!\$a	\$a 'nın değerini mantıksal tersine çevirir.
&&	AND	\$a && \$b	\$a ile \$b nin ikisinde true ise true döndürür.
	OR	\$a \$b	\$a yada \$b true ise true döndürür.
And	AND	\$a and \$b	&& ile aynı görevi yapar.
Or	OR	\$a or \$b	ile aynı görevi yapar.

UYGULAMA FAALİYETİ

Programlama içinde değişken kullanımı ile ilgili uygulamalar yapınız.

İşlem Basamakları	Öneriler
➤ İki adet integer tipindeki değişken tanımlayıp bu değişkenlere 80 ve 5 değerlerini atayınız.	➤ Değişken atamasına bakınız.
➤ Bu iki değişken arasında aritmetik 4 işlemi (toplama, çıkarma, çarpma, bölme) yapınız ve sonuçları ekrana yazdırınız	➤ Aritmetik operatörleri kullanınız. Echo komutunu kullanınız.
➤ Taban genişliği ve yüksekliği belli olan bir üçgenin alanını hesaplayan programı yazınız.	➤ Üçgenin alanı formülü = Taban genişliği*yükseklik/2
➤ “İnternet” ve “Programcılığı” sözcüklerini iki farklı değişkene atayınız	➤ Atama operatörünü kullanınız.
➤ Bu iki sözcüğü birleştirip tek bir değişkene atayınız ve ekrana yazdırınız	➤ String birleştirme operatörünü kullanınız.
➤ Tek boyutlu bir dizinin içine haftanın günlerini atayıp, ekrana alt alta yazdırınız	➤ Alt alta yazdırmak için tagını kullanınız.
➤ “pi” adında ve değeri 3.14 olan bir sabit tanımlayınız	➤ Sabit tanımlama konusundan yardım alabilirsiniz.
➤ Yarıçapı 3 cm olan bir çemberin alanını bulunuz.	➤ Çemberin alan formülü= Pi sayısı * çemberin yarıçapının karesi
➤ İnteger tipinde bir değişken tanımlayıp 10 değerini atayınız.	➤ Atama operatörünü kullanınız.
➤ 10 sayısını birer birer azaltıp her azaltmada ekrana değerini alt alta yazdırınız.	➤ Azaltma operatörünü kullanınız. Alt satıra geçmek için tagını kullanınız.
➤ İki boyutlu bir dizi tanımlayınız	➤ İki boyutlu dizi örneğine bakınız.
➤ Bu dizinin elemanlarına farklı değerler atayınız.	➤ Dizi elemanlarına değer atamak için atama operatörlerine bakınız.
➤ Bir kenarı 9 cm olan karenin çevresini hesaplayan ve çıktısı “bu karenin çevresi 36 cm’dir” şeklinde olan programı yazınız.	➤ Aritmetiksel operatörleri ve string birleştirme operatörlerini kullanınız.
➤ \$isim adlı bir değişkenin içine “Ali” değerini atayınız ve bu değişken değerinin boş olmadığını gösteriniz.	➤ Settype() komutunu kullanınız.
➤ Vize notunun %40 ‘ı ile final notunun %60 ını toplayan programı yazınız.	➤ Aritmetiksel operatörleri kullanınız.

ÖLÇME VE DEĞERLENDİRME

Aşağıdaki soruları dikkatlice okuyarak doğru seçeneği işaretleyiniz.

1. Aşağıdakilerden tanımlamalardan hangisi doğrudur?
A) \$1.sayi 15;
B) \$sayi_1= 15;
C) \$sayi 1 = 15;
D) \$sayı =15;
2. 6.15 değeri atanacak bir değişken hangi türde olmalıdır?
A) integer(tamsayı)
B) string
C) double
D) define
3. define (“sayı”,5); ifadesinden sonra aşağıdaki satırlardan hangisi yazılabilir?
A) sayi=sayi+1;
B) sayi--;
C) \$degisken=sayi+1;
D) sayi=\$degisken+1;
4. \$a *=\$b; komutunun anlamı nedir?
A) \$a=\$a*\$b;
B) \$b=\$a*\$b;
C) \$a=\$b*\$b;
D) \$b=\$a*\$a;
5. Aşağıdakilerden hangisi bir mantıksal operatördür?
A) =>
B) ++
C) &&
D) ==

DEĞERLENDİRME

Cevaplarınızı cevap anahtarıyla karşılaştırınız. Yanlış cevap verdiğiniz ya da cevap verirken tereddüt ettiğiniz sorularla ilgili konuları faaliyete geri dönerek tekrarlayınız. Cevaplarınızın tümü doğru ise bir sonraki öğrenme faaliyetine geçiniz.

ÖĞRENME FAALİYETİ-2

AMAÇ

Program denetimi işlemlerini gerçekleştirebileceksiniz.

ARAŞTIRMA

- Belirli bir şarta göre çalışan deyimler nelerdir, araştırınız.
- Döngüleri hangi amaçlar için kullanıyoruz, arkadaşlarımızla tartışınız.
- PHP de kullanılan döngüler nelerdir, araştırınız.

2. PROGRAM DENETİMİ

Programın çalışması esnasında birkaç farklı alternatif durum arasından seçim yapmamız, bazı komutları birden fazla tekrarlatmamız, mantıksal kararlar vermemiz ya da istenmeyen hataları önlemek amacıyla programımıza müdahale etmemiz gerekebilir. Bu gibi durumları kontrol altına almak amacıyla kontrol yapılarını ve döngüleri kullanıyoruz. Şarta bağlı olarak işlem yapmak için if ve switch ifadelerini, döngüler için ise for ve while ifadelerini kullanıyoruz. Şimdi bunları sırayla ve örneklerle inceleyelim

2.1 Kontrol Yapıları

➤ if Deyimi

Türkçe anlamı "eğer" demektir. Anlamından da anlaşıldığı gibi if deyimi program içinde belirlenen şartın doğru (true) veya yanlış (false) olmasına göre belirli komutların çalışmasını sağlar.

if deyiminin kalıbı aşağıdaki şekildedir;

If (şart)

Komut ifadesi;



Eğer şartımız sağlanıyorsa if deyiminden sonraki komut ifadesinde yer alan işlemler yapılır. Sağlanmıyorsa program komut ifadesini atlar ve bir alt satıra geçer.

if deyimini basit bir örnekle inceleyelim

```
<?  
$sonuc=100/2;  
if ($sonuc == 50)  
echo "Sonuç 50 dir."  
?>
```

Şart bölümünde \$sonuç değişkeni ile 50 sayısı karşılaştırılıyor. Bu karşılaştırmadan çıkan sonuç doğru (true) ise ekrana “Sonuç 50 dir.” İfadesi yazdırılıyor.

Şimdi bu ifadeye ek olarak **else** bölümünü inceleyeceğiz. If ...else olarak adlandırılan bu kalıpta şart sağlanıyorsa belirli komut ifadesi sağlanmıyor ise başka bir komut ifadesi çalıştırılmaktadır.

“if else” deyiminin kalıbı aşağıdaki şekildedir

```
If (şart)  
  
Birinci komut ifadesi; // şart doğru ise  
  
else  
  
İkinci komut ifadesi; //şart yanlış ise
```

Örnek: Yaşı 18’ den küçük olanların girmesi yasak olan bir web sayfası için kontrol programı yazalım. Ziyaretçimizin yaşı 18’ den küçükse ekrana “Yaşınız 18’ den küçük.” ve “Bu siteye girmeniz yasaktır.” yazılmasını, 18’ den büyük ise “Web sayfasına hoş geldiniz.” yazılmasını istiyoruz.

Program kodu

```
<?  
if ($yas < 18)  
{  
echo "Yaşınız 18 den küçük";  
echo "<BR>";  
echo "Bu siteye girmeniz yasaktır";  
}  
else  
echo "Web sayfasına hoş geldiniz"  
?>
```

Şartın gerçekleşmesi ya da gerçekleşmemesi durumunda birden fazla komutun işletilmesini istiyorsak komutlarımızı { } işaretleri arasına koymalıyız

Programın ihtiyaçlarına göre birden fazla şartımız olabilir. Bu gibi durumlarda şart bölümünde mantıksal operatörleri kullanarak şart deyimlerimizi sıralayabiliriz.

Örnek: Sınavdan aldığı puanı 160 ya da daha yüksek olan ve diploma notu 3.50 ya da daha yüksek olan öğrenciler için ekrana “Başarılı” diğerleri için “Başarısız” yazan gerekli kodları yazalım.

Program Kodu

```
<?  
if ($sinavnotu>=160 && $diplomanotu>=3.50)  
echo "Başarılı";  
else  
echo "Başarısız";  
?>
```

Ayrıca iç içe if yapısı da kullanılabilir. Örneğin işe alınacak bir eleman için cinsiyetinin erkek, boyunun en az 1.80, kilosunun ise en fazla 90 olması isteniyor. Bu değerlerin kontrolünü yapan ve sonunda bize “aranılan eleman bulundu” sonucunu yazan programı oluşturalım.

```
If ( $cinsiyet="erkek" )  
  If ( $boy>=1.80 )  
    If ( $kilo<=90 )  
      Echo "aranılan eleman bulundu";
```

➤ Switch Deyimi

Switch, seçilmiş bir değişkenin belli şartlardan hangisine uygun olduğuna bakıp o şart için belirlenmiş işlemleri gerçekleştirir.

Doğru şartı sağlayan gruba ait işlemlerin yapılmasından sonra "Break" deyimi ile switch yapısından çıkarılır. Break deyimi kullanılmırsa diğer şartlar kontrol edilmeksizin komutları sırayla çalıştırılır.

Switch deyiminin kalıbı aşağıdaki şekildedir.

```
Switch (değişken)  
{  
  Case şart1;  
  Yapılacak işlemler  
  Break;  
  
  Case şart2;  
  Yapılacak işlemler  
  Break;  
  
  Case şart3;  
  Yapılacak işlemler  
  Break;  
  
  Default:  
  Hiçbir şart sağlanmıyorsa yapılacak işlemler  
}
```

```
Örnek kod  
<?  
$takim="Galatasaray";  
Switch ($takim) {  
case "Galatasaray";  
echo "Tuttuğunuz takım Galatasaray";  
break;  
  
case "Fenerbahçe";  
echo "Tuttuğunuz takım Fenerbahçe";  
break;  
  
case "Beşiktaş";  
echo "Tuttuğunuz takım Beşiktaş";  
break;  
  
case "Trabzonspor";  
echo "Tuttuğunuz takım Trabzonspor";  
break;  
  
default:  
echo "yazdığınız takım listede yok"; }  
>
```

Ekran Çıktısı

Tuttuğunuz takım Galatasaray

Not: Eğer \$takim="Galatasaray" satırında Galatasaray, Fenerbahçe, Beşiktaş veya Trabzonspor yerine başka bir takım yazılıysa Default bölümdeki komutlar çalışacak ve ekrana "yazdığınız takım listede yok" şeklinde bir metin yazılacaktı.

Switch deyimine başka bir örnekle devam edelim. Daha önce belirlediğimiz iki sayı ve matematiksel dört işlemi aritmetiksel operatörleri kullanarak gerçekleştirilelim.

```
<?
$sayi1=10; // birinci sayımıza 10 değerini verdik.
$sayi2=5; // ikinci sayımıza 5 değerini verdik.
$islem="çarpma"; //yapılacak işlemi çarpma işlemi olarak belirledik.
Switch ($islem) { //switch deyimindeki $islem değişkeni işleme alınıyor.

Case "toplama"; //yazılan işlem toplama işlemi ise aşağıdaki işlemler yapılacak.
$sonuc=$sayi1+$sayi2; // değişkenler toplanıp $sonuc değişkenine aktarılıyor.
Echo $sonuc; //sonuç ekrana yazdırılıyor.
Break; // switch deyiminden çıkılıyor.

Case "çıkarma"; //yazılan işlem çıkarma işlemi ise aşağıdaki işlemler yapılacak
$sonuc=$sayi1-$sayi2; // değişkenler çıkartılıp $sonuc değişkenine aktarılıyor
Echo $sonuc; //sonuç ekrana yazdırılıyor
Break; // switch deyiminden çıkılıyor

Case "çarpma"; //yazılan işlem çarpma işlemi ise aşağıdaki işlemler yapılacak.
$sonuc=$sayi1*$sayi2; // değişkenler çarpılıp $sonuc değişkenine aktarılıyor.
Echo $sonuc; //sonuç ekrana yazdırılıyor.
Break; // switch deyiminden çıkılıyor.

Case "bölme"; //yazılan işlem bölme işlemi ise aşağıdaki işlemler yapılacak.
$sonuc=$sayi1/$sayi2; // değişkenler bölünüp $sonuc değişkenine aktarılıyor.
Echo $sonuc; //sonuç ekrana yazdırılıyor.
Break; // switch deyiminden çıkılıyor.

Default: //eğer yazılan işlem dört işlem dışında ise
Echo "yazdığınız işlem geçersiz"; //ekrana işlemin geçersiz olduğu yazılıyor.
}

?>
```

2.2. Döngüler

➤ For Döngüsü

Komut ya da komutların istenilen sayıda tekrarlanmasını sağlar. Sayaç mantığıyla çalışır. Döngüyü istediğimiz sayıdan başlatır istediğimiz sayıda sonlandırabiliriz. Döngünün değişme değerini her işlemde istediğimiz miktarda artırabilir ya da azaltabiliriz. Örneğin döngümüz için başlangıç değerini 1, bitiş değerini 100 olarak belirleyebilir. Artış miktarını ise 2 olarak ayarlayabiliriz. Bu durumda döngü içindeki komutlar 50 defa tekrar tekrar çalıştırılacaktır.

For döngüsünün kalıbı aşağıdaki gibidir

```
for (döngü değişkeni=başlangıç değeri; şart; değişim miktarı)  
{  
yapılacak işlemler  
}
```

Basit bir örnekle for döngüsünün yapısını ve çalışmasını inceleyelim. 1' den 10' a kadar olan sayıları alt alta yazdıralım.

```
Program Kodu  
?  
for ($i=1;$i<=10;$i++) //başlangıç değerimizi 1, bitiş  
//değerimizi 10, ve değişim miktarını +1 olarak belirledik.  
{  
echo $i; //değişken değeri ekrana yazdırılıyor.  
echo "<br>"; //bir satır alta geçiliyor.  
}  
?>
```

Ekran Çıktısı

```
1  
2  
3  
4  
5  
6  
7  
8  
9  
10
```

Bu örnekteki döngümüzde \$i değişkenimizin başlangıç değeri 1'dir. Döngü başladığında değişkenimizin değeri ekrana yazılmakta ve bir satır alta geçilmektedir. Değişkenin değeri 10' a eşit olmadığı için değişken değeri bir artırılarak tekrar döngüye

girmektedir. Aynı işlemler tekrarlanmaktadır. Bu işlem değişken değeri 10 oluncaya kadar devam etmektedir.

➤ While Döngüsü

While döngüsü, içerisinde tanımlanan koşul sağlandığı sürece belirli komut ifadesi grubunun çalıştırılmasını sağlar. Şart ifadesinin doğruluğu her döngünün başında kontrol edilir, buna göre döngünün içerisindeki komut ifadeleri çalıştırılmaya devam eder ya da döngü sona erer. Eğer şart sağlanmıyorsa while hiç çalıştırılmaz

while döngüsünün kalıbı aşağıdaki gibidir:

```
while(şart)
{
Yapılacak işlemler
}
```

Örnek bir program üzerinde while döngüsünü inceleyelim. Daha önce for döngüsü ile yaptığımız 1' den 10' a kadar olan sayıları yazdıran programı while döngüsü ile yazalım. Böylece hem while döngünün kullanımını hem de iki döngü arasındaki farklılıkları göreceksiniz

Örnek Kod

```
<?
$i=1; //başlangıç değerini belirledik.
while ($i<=10) //10 a kadar olma şartımızı belirledik.
{
echo $i; //değişken değerini yazdırdık.
echo "<br>"; //bir alt satıra geçtik.
$i++; //değişken değerimizi 1 artırdık.
}
?>
```

Yukarıdaki programda ilk olarak tanımlanan bir "i" değişkenine 1 değeri atanmaktadır. While döngüsü, "i", 10' dan küçük olduğu sürece çalışacak şekilde kurulmuştur. Programın 5 ve 6. satırlarında i' nin değeri ekrana yazdırılmakta ve bir satır alta inilmektedir. \$i++; komutu önemlidir çünkü "i" değeri bu komutla değiştirilmektedir. Döngümüz i değerine bağlı olarak kurulmuştur. i değerinin sabit kalması durumunda (\$i<=10) ifadesi (1<=10) olarak kalacağından sürekli true sonucunu döndürecektir.

While (True) →Döngüye gir
While (False)→Döngüye girme

Anlamına gelmektedir.

Dikkat edilmesi gereken bir başka nokta ise $i++$ ifadesi (bu deęer deęişiklięi komutunu farklı artış ve azalmalar sağlayacak şekilde de yazabilirsiniz). Döngümüzün sonunda olduęu için döngümüzün son çalıştığı seferde ($i=10$ durumunda) son bir kez daha 1 artırılarak 11'e eşit olacaktır. $11 \leq 10$ ifadesi false sonucu döndürdüğünden döngü tekrar çalışmayacaktır.

UYGULAMA FAALİYETİ

Programlama içinde değişken kullanımıyla ilgili uygulamalar yaparak, program denetimi işlemlerini gerçekleştiriniz.

İşlem Basamakları	Öneriler
➤ 100 tam puan üzerinden notları harf notuna çeviren Php kodunu yazınız.	➤ Bu uygulamayı hem if deyimi ile hem de switch deyimi ile ayrı ayrı yapınız.
➤ Önceden belirlenen 3 sayıdan en büyüğünü bulan programı yazınız.	➤ İç içe if yapısını kullanınız.
➤ Önceden belirlenen bir sayının 2' ye kalansız olarak bölünüp bölünemediğini belirleyen php kodunu yazınız.	➤ Mod operatörünü ve if ifadesini kullanabilirsiniz.
➤ Maaşı 500 YTL' den fazla olanlardan % 25 vergi kesintisi yapan, maaşı 500 YTL' den az olanlardan hiç kesinti yapmayan php kodunu yazınız.	➤ If ifadesini kullanabilirsiniz.
➤ 1' den 10 'a kadar olan sayıları aralarına + işareti koyarak yazan ve toplamlarını = işareti ile yazan php kodunu yazınız.	➤ For döngüsünü kullanınız. ➤ Ekran çıktısı ➤ 1+2+3+4+5+6+7+8+9+10=55
➤ X ➤ XX ➤ XXX ➤ XXXX	➤ Yandaki şekli ekrana for döngüsü ile yapınız. ➤ İç içe for döngüsü kullanabilirsiniz.
➤ Değerleri önceden girilen 10 elemanlı tek boyutlu bir dizinin elemanlarını toplayan ve ortalamasını alan php kodunu yazınız.	➤ While deyimini kullanınız.
➤ 1' den 1000'e kadar olan ve 5'e veya 7'ye kalansız bölünen sayıların toplamını bulan php kodunu yazınız.	➤ Bu uygulamayı hem for deyimiyle hem de while deyimiyle yapınız.
➤ Değerleri daha önceden girilmiş 5 elemanlı tek boyutlu bir dizinin elemanlarını büyükten küçüğe sıralayan Php kodunu yazınız.	➤ Bu uygulamada istediğiniz ifadeyi kullanabilirsiniz.
➤ 1' den 20' ye kadar olan çift sayıları ve karelerini ekrana yandaki gibi yazdıran Php kodunu yazınız.	➤ Sayı Karesi ➤ 4 ➤ 4 16
➤ 18 yaşından küçük kızlar ve 18 yaşından küçük erkekler için "giremezsiniz" yazısını yazdıran Php kodunu yazınız.	➤ Mantıksal operatörleri kullanabilirsiniz
➤ Çarpım tablosunu yapan programı yazınız.	➤ Birlerden onlara kadar tüm sayıların tüm çarpım tablosunu hazırlayınız.
➤ $1 + \frac{1}{2} + \frac{1}{3} + \frac{1}{4} + \frac{1}{5} + \dots$ dizisinin ilk 50 elemanının toplamını bulan Php kodunu yazınız.	➤ Sonuç: 4,499205

ÖLÇME VE DEĞERLENDİRME

Aşağıdaki soruları dikkatlice okuyarak doğru seçeneği işaretleyiniz.

1. Aşağıdakilerden deyimlerden hangisi şarta bağlı işlem yapar?
A) If Else
B) For
C) until
D) repeat
2.

```
<? $y=1;  
for ($i=1;$i<=10;$i++)  
$y=$y+$i;  
echo $y; ?>
```

kodunun çıktısı nedir?
A) 10
B) 55
C) 56
D) 99
3. `if ($yas>18) && ($cinsiyet="erkek")` kodunun anlamı nedir?
A) Yaşı 18' den büyük ve cinsiyeti erkek ise
B) Yaşı 18' den büyük veya cinsiyeti erkek ise
C) Yaşı 18' ve cinsiyeti erkek değil ise
D) Yaşı 18' den küçük ve erkek ise
4. Aşağıdaki ifadelerden hangisi doğrudur?
A) if deyiminden sonra yalnızca tek bir komut kullanılabilir.
B) for döngüsünün sınırlarını belirleyemeyiz.
C) break komutu döngüye devam etmeyi sağlar
D) iç içe birden fazla if deyimi kullanılabilir.
5. **For** döngüsü için aşağıdakilerden hangisi yanlıştır?
A) Başlangıç değerini belirleyebiliriz.
B) Bitiş değerini belirleyebiliriz.
C) Artış değeri her zaman 1' dir.
D) İstedığımız yerde döngüyü sonlandırabiliriz.

DEĞERLENDİRME

Cevaplarınızı cevap anahtarıyla karşılaştırınız. Yanlış cevap verdiğiniz ya da cevap verirken tereddüt ettiğiniz sorularla ilgili konuları faaliyete geri dönerek tekrarlayınız. Cevaplarınızın tümü doğru ise "Modül Değerlendirme"ye geçiniz.

MODÜL DEĞERLENDİRME

Aşağıdaki cümlelerin başında boş bırakılan parantezlere, cümlelerde verilen bilgiler doğru ise D, yanlış ise Y yazınız.

1. () 4.15 değerini integer tipindeki bir değişkene atayabiliriz.
2. () Diziler en fazla iki boyutlu olabilir.
3. () \$a && \$b ifadesi için \$a ile \$b' nin ikisi de true ise true döndürür.
4. () **if** deyimindeki şart sağlanmıyorsa **else**' den sonraki komutlar çalışır.
5. () Switch değer verdiğimiz bir değişkenin belirlediğimiz şartlardan hangisine uygun olduğuna bakıp gerekli işlemi yapar.
6. () Sabitler önceden tanımlanır ve değer atanır ve program içinde değeri değiştirilebilir.
7. () String birleştirme amaçlı olarak en sık kullanılan operatör "." (nokta) operatörüdür.
8. () "**Break**" komutu PHP script dilinde döngülerde işlemi durdurup işlemi sonlandırmamızı sağlar.

DEĞERLENDİRME

Cevaplarınızı cevap anahtarıyla karşılaştırınız. Yanlış cevap verdiğiniz ya da cevap verirken tereddüt ettiğiniz sorularla ilgili konuları faaliyete geri dönerek tekrarlayınız. Cevaplarınızın tümü doğru ise bir sonraki modüle geçmek için öğretmeninize başvurunuz.

UYGULAMALI TEST

Bu faaliyet kapsamında aşağıda listelenen davranışlardan kazandığınız beceriler için **Evet**, kazanamadıklarınız için **Hayır** kutucuklarına (X) işareti koyarak kontrol ediniz.

Değerlendirme Ölçütleri	Evet	Hayır
1. Tüm veri türlerinde değişken tanımlayıp değer atadınız mı?		
2. İki integer tipindeki değişken arasında aritmetik 4 işlemi yaptınız mı?		
3. String birleştirme işlemini yaptınız mı?		
4. Tek boyutlu bir dizinin elemanlarına 1' den 10' a kadar olan sayıları atadınız mı?		
5. Pi sayısını sabit olarak tanımlayarak çemberin alanını hesaplayan php kodunu yazdınız mı?		
6. 100 tam puan üzerinde notları harf notuna çeviren php kodunu yazdınız mı?		
7. 1' den 1000' e kadar olan ve 5'e veya 7'ye kalansız bölünen sayıların toplamını bulan php kodunu yazdınız mı?		
8. Tek boyutlu bir dizinin içine haftanın günlerini atayıp for döngüsü ile ekrana alt alta yazan php kodunu yazdınız mı?		
9. Önceden belirlenen bir sayının 2' ye kalansız olarak bölünüp bölünemediğini belirleyen php kodunu yazdınız mı?		
10. Çarpım tablosunu yapan programı yazdınız mı?		
11. $1 + \frac{1}{2} + \frac{1}{3} + \frac{1}{4} + \frac{1}{5} + \dots$ dizisinin ilk 50 elemanının toplamını bulan php kodunu yazdınız mı?		
12. Değerleri önceden girilen 10 elemanlı tek boyutlu bir dizinin elemanlarını toplayan ve ortalamasını alan php kodunu while deyimi ile yazdınız mı?		
13. X XX XXX XXXX XXXXX şeklini oluşturan programı yazdınız mı?		

DEĞERLENDİRME

Değerlendirme sonunda “Hayır” şeklindeki cevaplarınızı bir daha gözden geçiriniz. Kendinizi yeterli görmüyorsanız öğrenme faaliyetini tekrar ediniz. Bütün cevaplarınız “Evet” ise bir sonraki modüle geçmek için öğretmeninize başvurunuz.

CEVAP ANAHTARLARI

ÖĞRENME FAALİYETİ 1'İN - CEVAP ANAHTARI

1	B
2	C
3	C
4	A
5	C

ÖĞRENME FAALİYETİ 2'NİN - CEVAP ANAHTARI

1	A
2	C
3	A
4	D
5	C

MODÜL DEĞERLENDİRME'NİN CEVAP ANAHTARI

1	YANLIŞ
2	YANLIŞ
3	DOĞRU
4	DOĞRU
5	DOĞRU
6	YANLIŞ
7	DOĞRU
8	DOĞRU

KAYNAKÇA

- http://www3.itu.edu.tr/~sariel/course_notes/vtys2003/PHP1_2_slide.pdf
(16.05.2013)