

2021-2022 EĞİTİM-ÖĞRETİM YILI

PYTHON DERS NOTLARI

Tatvan Anadolu Lisesi



PRESENTED TO
Mustafa BALDIR

Bilişim Teknolojileri ve Yazılım





Python Uygulama Kodları 1

İfade ve Kavram	Açıklama								
Bilgisayar Veriyi Nasıl Saklar?	Bilgisayar veriyi hafızada saklar. Her bir değişken için hafızada belirli bir alan ayrılır ve bu alan her seferinde tek bir değer saklayabilir. Kullanıcı, var olan değer yerine yeni bir değer atadığında eski değer silinir. Hafızada bu konumlar geçicidir. Programın çalışması bittiğinde ya da bilgisayar kapatıldığında bu veriler silinir. Verilerin daha sonra tekrar kullanılması gerekiyorsa sabit disk gibi kalıcı bir konuma kaydedilmeleri gerekir. Bu şekilde kaydedilen verilere dosya adı verilir.								
Değişken Nedir?	Yazılan programlar bilgisayarın belleği üzerinde çalışır program içerisinde kullanılan her veri için bellekte bir yer ayrılır. Her bir bellek bölgesinin bir adresi vardır. Ancak adreslerle işlem yapmak zordur. Bu nedenle bellekteki verilere erişmek için, verinin tutulduğu bellek bölgesine program içerisinde sembolik isimler verilir. Bir bellek bölgesine işaret eden bu sembolik isimlere değişken adı verilir. Değişkenler programın akışı içerisinde farklı zamanda farklı değerler alabilir. Örneğin yandaki atamalardan sonra a=15 şeklinde bir atama daha gerçekleştirilirse, bellekteki a hücresinin değeri değiştirilerek 15'e ayarlanır.								
Değişken Adlandırma Kuralları	<table><tr><th colspan="2">BELLEK</th></tr><tr><td>a</td><td>3</td></tr><tr><td>b</td><td>3214.5987</td></tr><tr><td>c</td><td>Ali</td></tr></table> a=3 b=3214.5987 c="Ali" a=15 print(a) 15 1-Değişken adları sayı ile başlamaz. 3_kilo_elma = "10 tl" ✗ kilo_elma_3 = "10 tl" ✓ 2-Değişken adları özel sembol içermez (_ altçizgi hariç) gelir?= "500 TL" ✗ kullanici_adi= "admin" ✓ 3- Değişken adlarında boşluk olmaz. kullanici adi = "admin" ✗ kulllanici_adi = "admin" ✓ 4- Değişken adlarında bazı özel anlam ifade eden kelimeler kullanılmaz. Not: Python'da özel anlam ifade eden kelimeleri görmek için aşağıdaki kodları yazın. ['False', 'None', 'True', 'and', 'as', 'assert', 'break', 'class', 'continue', 'def', 'del', 'elif', 'else', 'except', 'finally', 'for', 'from', 'global', 'if', 'import', 'in', 'is', 'lambda', 'nonlocal', 'not', 'or', 'pass', 'raise', 'return', 'try', 'while', 'with', 'yield'] 5-Python dili büyük küçük harf duyarlı bir dildir. sayi=10 Sayi=20 print(sayi+30) 40 6-Değişkenler adlandırılırken İngiliz alfabesindeki karakterler kullanılmalıdır.	BELLEK		a	3	b	3214.5987	c	Ali
BELLEK									
a	3								
b	3214.5987								
c	Ali								



Python Uygulama Kodları 1

Değişkenlere Değer Atama

Not: Değişkenlere değer ataması yaparken = operatörü kullanılır.

Örnek:

```
x=3
```

Not: Bir değere iki veya daha fazla değişken atayabiliriz.

Örnek:

```
>>>a=b=c=4 → a,b ve c değişkenlerinin hepsine birden 4 değeri atandı.
```

```
>>>print(a*b+c) → 4*4+4
```

```
20
```

Not: Bir değişkene defalarca farklı değerler atayabiliriz.

Örnek:

```
>>>x = 10
```

```
>>>x = 20
```

```
>>>x = 30 → Değişkenin en son değeri
```

```
>>>print(x) → Değişkenin en son değeri geçerlidir.
```

```
30
```

Çoklu atama:

Örnek:

```
>>> x, y, z = 4,3,2 → x değişkenine 4, y değişkenine 3, z değişkenine ise 2 değeri atandı.
```

```
>>> print(x*y-z) → 4*3-2
```

```
10
```

Değişken Takası:

Örnek:

```
>>> x,y,z=4,3,2
```

```
>>> x,y,z=y,z,x → x artık y, y artık z, z artık x oldu.
```

```
>>> print(z**y*x) → 4**2*3
```

```
48
```

Değişken İptali: Değişken iptali için komut penceresini kapatıp açabiliriz ya da del komutunu kullanabiliriz.

Örnek:

```
>>>a=2
```

```
>>>print(a)
```

```
2
```

```
>>>del a → a değişkeni artık yok yani iptal edildi.
```

```
>>>print(a) → a değişkenini iptal ettiğimiz için kod hata verir. ✗
```



Python Uygulama Kodları 1

ÖRNEK 1.

```
a=6
print(a)
b=3.14159265359
print(b)
c="Merhaba Dünya"
print(c)
```

ÖRNEK 2.

```
3boy=177
boy3=189
print(boy3)
```

ÖRNEK 3.

```
ad-soyad="Ali Can"
ad_soyad="Ali Can"
print(ad_soyad)
```

ÖRNEK 4.

```
x=y=z=3
print(x*y+z)
```

ÖRNEK 5.

```
k,l,m=4,5,2
print(k*l+m)
```

ÖRNEK 6.

```
d,e,f=1,2,3
d,e,f=f,d,e
print(f-d*e)
```

ÖRNEK 7.

```
m="Ali YILMAZ"
print(m)
del m
print(m)
```



Python Uygulama Kodları 2

İfade ve Kavram	Açıklama	Örnek
print	Ekrana yazdırma fonksiyonudur. Metinleri(string) yazdırırken tırnak işareti içerisine almamız gerekir. NOT: Sayıları tırnak içerisine alınmaz.	<pre>print("Merhaba Dünya") print(35) print(3.14) a=4 b=5 print(a+b) 9</pre>
print komutunda tırnak kullanımı	Üç farklı tırnak kullanımı mevcuttur. 1. Tek tırnak → print('Python programlama dilinin adı "piton" yılanından gelmez') 2. Çift tırnak → print("Elazığ'ın 5 günlük hava durumu tahmini") 3. Üç tırnak → print("""Hello world Merhaba Dünya.""")	

ÖRNEK 1.

```
print("Merhaba Dünya")
print(35)
print(3.14)
print("5")
```

ÖRNEK 2.

```
print(5+6)
print("5+6")
```

ÖRNEK 3.

```
print("bilgi"+"sayar")
print("Ali","Ayşe","Can")
```

ÖRNEK 4.

```
simitf=1.25
print("Simit'in fiyatı",simitf)
```

ÖRNEK 5.

```
a=3
b=2
print(a*b)
```

ÖRNEK 6.

```
c="4"
d="yıldız"
print(c,d)
```

ÖRNEK 7.

```
print("GS"*4, "*" * 4)
```

ÖRNEK 8.

```
print("1"*4)
```

ÖRNEK 9.

```
print(('6'+ '5')*3)
```

ÖRNEK 10.

```
print('''Dört sene üst üste
Şampiyon oldun
Avrupa Fatih'i Galatasaray
*****''')
```



Python Uygulama Kodları 3

İfade ve Kavram	Açıklama	Örnek
Veri Türleri	Python dilinin üç temel veri türü vardır. Bunlar: 1. Sayısal Türler (Tamsayı(int), ondalıklı sayı(float), karmaşık(complex) 2. Karakter dizileri(string) 3. Mantıksal veri türü(bool (True-False) doğru ya da yanlış)	
int	Tam sayılardır. Pozitif, negatif ya da sıfır değeri alabilir. Kesirli değer içermez.	100,12,0,-5
float	Reel(ondalıklı) sayılardır. Kayan noktalı sayılar da denir. Ancak burada virgül yerine nokta kullanmamız gerekir.	4.5,6.789,-2.7453,5.0
complex	Karmaşık sayılardır.	4+7j, -2+3j
str	Karakter dizileri bir ya da daha çok karakterin birleşimi ile oluşan sözcük yapılarıdır. Tek, çift ya da üç tırnak içerisinde alınarak gösterilir.	str1="merhaba" str2='günaydın' a="5"
bool	Mantıksal veri türü içerisinde doğru ya da yanlış ifadeleri saklanabilir. Doğru True ifadesi ile yanlış False ifadesi ile gösterilir. Mantıksal bir değişkenin alabileceği iki değerden biri True ise diğeri False'tur. Mantıksal bir değişkenin bu iki değer dışında bir değer alamaz.	m=2<3 True n=5>8 False
type()	Verilerin türünü sorgular.	type(100)→<class 'int'> type(3,14)→ <class 'float'> type("Ali")→<class 'str'> type(True)→<class 'bool'>
input() fonksiyonu :	Kullanıcıdan bilgi almak için kullanılır. input ile okunan tüm veriler str(karakter dizisi) türü olarak gelir. Kullanıcıdan sayısal bir veri girmesi istenmişse matematiksel işlem yapabilmek için veriyi sayısal bir türe çevirmemiz gerekir.	ad=input("Adınız :") yas=int(input("Yaşınız :"))
Veri türü dönüşümleri	Python veri türlerinin, belirli durumlarda farklı türlere dönüştürülmesi gerekir. Örneğin; bir karakter dizisinin sayıya dönüşümü ya da bir ondalıklı sayının bir tamsayıya dönüşümü gibi. 1. int("65")→65 Karakter dizisini tam sayıya çevirir. 2. str(12)→"12" Sayısal türdeki veriyi karakter dizisine çevirir. 3. float(5)→5.0 Sayısal türdeki veriyi ondalıklı sayıya çevirir.	



ÖRNEK 1.

```
int(28.9)
int(5+4)
int(5/3)
int(5*0.4)
int(0.8+0.9)
```

ÖRNEK 2.

```
str(4)
str(4.0)
str(10+2)
```

ÖRNEK 3.

```
float(5)
float(-5)
float(3+4)
float(3.5+4)
```

ÖRNEK 4.

```
type("6")
type(6)
type(6.89)
type(3.0)
type(6+7)
type("6+7")
type(Merhaba)
type("Merhaba")
```

ÖRNEK 5.

```
sayi1=7.897
int(sayi1)
sayi2=9
float(sayi2)
```

ÖRNEK 6.

```
m=2<3
type(m)
```

Python Uygulama Kodları 3

ÖRNEK 7.

```
n=5>8
type(n)
```

ÖRNEK 8.

```
ad=input("Adınızı öğrenebilir miyim")
print("Adınız ",ad)
yas=input("Yaşınızı öğrenebilir miyim")
print("Yaşınız ",yas)
```

ÖRNEK 9.

```
kenar=input("Karenin kenar uzunluğunu giriniz")
cevre=4*kenar
print("Karenin çevresi ",cevre)
```

BİLGİ : Sonuç neden yanlış çıktı ? Çünkü kenar uzunluğunu sayıya çevirmedik.
Doğrusu alttaki örnekte.

ÖRNEK 10.

```
kenar=int(input("Karenin kenar uzunluğunu giriniz"))
cevre=4*kenar
print("Karenin çevresi ",cevre)
```

ÖRNEK 11.

```
yaricap=float(input("dairenin yarıçapını giriniz"))
pi=3.14
alan=pi*yaricap*yaricap
```

ÖRNEK 12.

```
uzunkenar=int(input("Dikdörtgenin uzun kenar uzunluğunu giriniz"))
kisakenar=int(input("Dikdörtgenin kısa kenar uzunluğunu giriniz"))
cevre=2*(uzunkenar+kisakenar)
print("Dikdörtgenin çevresi ",cevre )
```



Python Uygulama Kodları 4

İfade ve Kavram	Açıklama	Örnek
\n parametresi:	Bu parametreye newline adı verilir. print() fonksiyonu içerisinde kullanıldığında ilgili yerden bir alt satıra geçiş yapar.	<pre>print("bilgisayar\nbilimi")</pre> bilgisayar bilimi
\t parametresi:	print() fonksiyonu içerisinde kullanıldığında ilgili yerden bir tab kadar boşluk bırakır	<pre>print("Ocak\tŞubat\tMart")</pre> Ocak Şubat Mart
* parametresi:	stringi parçalara böler. Varsayılan olarak karakterlerin arasına boşluk konuluyor.	<pre>print(*"Linux")</pre> L i n u x
sep parametresi:	print() fonksiyonu içerisinde kullanıldığında yazdığımız değerlerin arasına istediğimiz karakterlerin yerleştirilmesini sağlar. Eğer bu parametreyi kullanmazsak değerlerin arasına varsayılan olarak boşluk yerleştirir.	<pre>print("bir ","iki ","üç ","dört ","on dört " ,sep="mumdur ")</pre>
end parametresi:	print() içerisinde kullanılır. Yazdırılmak istenen ifadelerin sonuna hangi karakterin geleceğini belirler. Varsayılan olarak "\n" parametresi ile birlikte gelir. Yani yazılan ifade bitince bir alt satıra geçer.	<pre>print("bir ","iki ","üç ","dört ","ondört " ,sep="mumdur ",end="mumdur\n")</pre>

ÖRNEK 1.

```
print("bilgisayar\nbilimi")
```

ÖRNEK 2.

```
print("Ocak\tŞubat\tMart")
```

ÖRNEK 3.

```
print(3,4,5,6)  
print(3,4,5,6,sep = ".")
```

ÖRNEK 4.

```
print(*"Python")
```

ÖRNEK 5.

```
print(*"Python",sep = "\n")
```

ÖRNEK 6.

```
print(*"TBMM",sep = ".")
```

ÖRNEK 7.

```
print("1","2","3",sep="X")
```

ÖRNEK 8.

```
print("Türk;","öğün","çalış","güven",sep="\n")
```

ÖRNEK 9.

```
print("Türk;","öğün","çalış","güven",sep="\t")
```

ÖRNEK 10.

```
print("bir ","iki ","üç ","dört ","on dört " ,sep="mumdur ")
```

ÖRNEK 11.

```
print("bir ","iki ","üç ","dört ","ondört " ,sep="mumdur",end="mumdur\n")
```




Python Uygulama Kodları 5

İfade ve Kavram		Açıklama		
Operatörler	Veriler üzerinde aritmetik, mantıksal, karşılaştırma gibi işlemleri yerine getiren sembollere operatör adı verilir. Örneğin; toplama işleminde kullanılan “+” işareti bir operatördür. Operatörlerin üzerinde işlem yaptığı değerlere ise operand adı verilir. “3+4” işleminde 3 ve 4 değerleri birer operand'dır.			
Aritmetik Operatörler	İŞLEV	GÖREVİ	ÖRNEK	SONUÇ
	+	Toplama	5 + 3	8
	-	Çıkarma	5 – 3	2
	*	Çarpma	5 * 3	15
	/	Ondalıklı Bölme	5 / 3	1.666667
	**	Üs alma	5 ** 3	125
	//	Bölümün tamsayı kısmını verir.	15//2	7
	%	Mod alır. Kalanı verir	15 % 2	1
Aritmetik Operatörlerin Öncelik Sırası	Herhangi bir matematiksel işlem dizisi içerisindeki operatörlerin öncelik sıralaması, matematikteki kabuller ile aynıdır. Yani çarpma, bölme işlemleri toplama ve çıkarma işlemlerine göre önceliğe sahiptir. İşlem öncelikleri parantezler ile grupta yapılarak değiştirilebilir. a=3*2+1→7 iken a=3*(2+1) →9 olur.			

ÖRNEK 1.

```
5+6
4.5+6.7
9+3.0
"9"+"6"
"9"+6
"7"+str(8)
2+int("3")
```

ÖRNEK 2.

```
15-9
4.3-2.7
-7--9
```

ÖRNEK 3.

```
7*8
-7*-9
1.5*2.5
"w"*3
```

ÖRNEK 4.

```
x=4
y=3
3*x+2*y-5
```

ÖRNEK 5.

```
21/3
int(21/3)
21/0
```

ÖRNEK 6.

```
24//7
3//8
4.5//1.2
2.1//1
```

ÖRNEK 7.

```
3**2
4**3
5**-1
```

```
16**1/2
16**(1/2)
```

ÖRNEK 8.

```
25%7
22%11
```

ÖRNEK 9.

```
print(5*3+2)
print(5*(3+2))
```

ÖRNEK 10.

```
5+4*3/3-9
15-2**4/2+(-2-2)
100**0.5/10*2
39//12+12/2
```



Python Uygulama Kodları 6

İfade ve Kavram		Açıklama			
İşlemleri Atama Operatörleri		Program yazarken bir değişkenin değerini değiştirdikten sonra yine kendi üzerine atadığımız işlemler vardır. Örneğin bir ağacın boyunun her yıl 2 metre uzadığı söyleniyorsa ağacın şimdiki boyu geçen yılki boyunun 2 metre fazlası olacaktır. Her geçen yıl için ayrı ayrı boy değişkenleri tanımlanmadan ağacın mevcut boyu artırılarak ortaya çıkan sonuç yine şu anki boy değeri olarak belirlenir. boy=boy+2 yada boy+=2 Aşağıdaki tablo için a değişkeninin mevcut değerinin 8 olduğunu varsayarsak:			
	Operatör	İşlem	Eşdeğeri	a'nın yeni değeri	GÖREVİ
	+=	a = a +2	a +=2	10	Toplama işlemi yapıp, ortaya çıkan değeri tekrar aynı değişkene atar.
	-=	a = a -2	a -=2	6	Çıkarma işlemi yapıp, ortaya çıkan değeri tekrar aynı değişkene atar.
	/=	a = a /2	a /=2	4.0	Bölme işlemi yapıp, ortaya çıkan değeri tekrar aynı değişkene atar.
	*=	a = a *2	a *=2	16	Çarpma işlemi yapıp, ortaya çıkan değeri tekrar aynı değişkene atar.
	%=	a = a %2	a %=2	0	Bölme işleminden kalan sayıyı tekrar aynı değişkene atar.
	**=	a = a **2	a **=2	64	Bir sayının kuvvetini hesapladıktan sonra çıkan değeri tekrar aynı değişkene atar.
	//=	a = a //2	a //=2	4	Bir sayının bölme işleminin sonucunu tekrar aynı değişkene atamaktır
Karşılaştırma Operatörleri		İki değer birbirlerine göre durumlarını (eşitlik, farklılık, büyüklük, küçüklük) tespit etmek için karşılaştırma operatörleri kullanılır. Karşılaştırma işlemlerinin sonucunda mantıksal doğru (True) ya da yanlış (False) değerleri elde edilir. a değişkeninin 8 , b değişkeninin 10 olduğunu varsayarak aşağıdaki tabloyu inceleyelim.			
Opr.	Anlamı	İşlem	Sonuç	Görevi	
==	Eşit midir?	a==b	False	İki değer birbirine eşitse True, değilse False değer döner.	
!=	Farklı mıdır?	a!=b	True	İki değer birbirine eşit değilse True, diğer durumda False döner.	
>	Büyük müdür?	a>b	False	Soldaki değer sağdaki değerden büyükse True, değilse False döner.	
<	Küçük müdür?	a<b	True	Soldaki değer sağdaki değerden küçükse True, değilse False döner.	
>=	Büyük yada eşit midir?	a>=b	False	Soldaki değer sağdaki değerden büyükse veya sağdaki değere eşitse True, değilse False döner.	
<=	Küçük yada eşit midir?	a<=b	True	Soldaki değer sağdaki değerden küçükse veya sağdaki değere eşitse True, değilse False döner.	
Mantıksal Operatörler		Mantıksal operatörler doğru yada yanlış değeri taşıyan mantıksal ifadeleri çeşitli şekillerde bir araya getirmek, düzenlemek için kullanılır. Temel 3 adet mantıksal işlem vardır. Bunlar VE(AND) , VEYA(OR) , DEĞİL(NOT) işlemleridir.			
Operatör		Görevi			
and Operatörü		Bağlanan karşılaştırma işlemlerinin hepsinin kendi içinde sonucu True ise genel sonuç True, diğer durumlarda ise sonuç False çıkar.			
or Operatörü		Bütün karşılaştırma işlemlerinin sonuçlarından en az birinin True olmasına bakar. Bağlanan karşılaştırma işlemlerinin en az birinin True olmasında genel sonuç True, diğer durumlarda ise sonuç False çıkar.			
not Operatörü		Bu operatör sadece bir mantıksal değeri veya karşılaştırma işleminin tam tersi sonuca çevirir. Yani, not operatörü True olan bir sonucu False, False olan bir sonucu True sonucuna çevirir.			



Python Uygulama Kodları 6

a=23 b=10	OPERATÖR	İŞLEM	SONUÇ	ÖRNEK	SONUÇ
	and	True and True	True	a == 23 and b == 10	True
	and	True and False	False	a == 23 and b == 56	False
	and	False and False	False	a == 20 and b == 56	False
	or	True or True	True	a == 23 or b == 10	True
	or	True or False	True	a == 23 or b == 56	True
	or	False or False	False	a == 20 and b == 56	False
Aitlik Operatörü	not			not a==23	False
	Herhangi bir elemanın bir koleksiyona ait olup olmama durumunu kontrol eden operatörlerdir. Örneğin herhangi bir harfin bir karakter dizisi içinde yer alıp almadığını kontrol ederken kullanılır. İki adet aitlik operatörü vardır. in ve not in				
	Operatör	Görevi	Örnek	Sonuç	
	in Operatörü	İlk operandla verilen değer, ikinci operand ile verilen dizi içerisinde varsa True yoksa False sonucunu üretir.	print("a" in "Merhaba") print("abc" in "Merhaba")	True False	
	not in Operatörü	İlk operandla verilen değer, ikinci operand ile verilen dizi içerisinde varsa False yoksa True sonucunu üretir.	print("a" not in "Merhaba") print("abc" not in "Merhaba")	False True	

ÖRNEK 1.

```
a = 23
a = a + 5
print(a)
a += 5
print(a)
```

ÖRNEK 2.

```
b = 30
b = b - 5
print(b)
b -= 5
print(b)
```

ÖRNEK 3.

```
c = 27
c = c/3
print(c)
c /= 3
print(c)
```

ÖRNEK 4.

```
d = 10
d = d*2
print(d)
d*= 2
print(d)
```

ÖRNEK 5.

```
e = 40
e%= 3
print(e)
```

ÖRNEK 6.

```
f = 4
f**= 3
print(f)
```

ÖRNEK 7.

```
g = 45
g=g//3
print(g)
g//=3
print(g)
```

ÖRNEK 8.

```
1 < 2 and "Murat" == "Murat"
2 > 3 and "Murat" == "Murat"
1 < 2 or "Murat" != "Murat"
2 > 3 or "Murat" != "Murat"
2 > 3 or "Murat" != "Murat" or 3.14 < 4.32
2 == 2 and 3.14 < 2.54 and "Elma" != "Armut"
not 2 == 2
not "Python" == "Php"
not (2.14>3.49or(2!=2and"Murat" == "Murat"))
```

ÖRNEK 9.

```
dersNotu = 65
print(dersNotu > 60)
```

ÖRNEK 10.

```
cinsiyet = "erkek"
print(cinsiyet != "kadın")
```

ÖRNEK 11.

```
adi = "Ali"
print(adi == "Veli")
```

ÖRNEK 12.

```
isim="Ali Can"
"C" in isim
"Al" in isim
"d" in isim
"Can" not in isim
"e" not in isim
```



Python Uygulama Kodları 7

İfade ve Kavram	Açıklama																				
Algoritma	Bilgisayara hangi işlemi hangi sırada yapması gerektiğini söyleyen yönergeler bütünüdür.																				
Akış şeması	Algoritmanın görsel gösterimidir. Programcı, oluşturulan algorithmadan grafiksel gösterimler oluşturur. Akış şemalarını oluşturmak için kullanılan <i>evrensel simgeler</i> ve bu her bir simgenin anlamı vardır. <table><tr><th>Simge</th><th>İşlev</th></tr><tr><td></td><td>Başla/Bitir</td></tr><tr><td></td><td>Giriş</td></tr><tr><td></td><td>Atama/İşlem</td></tr><tr><td></td><td>Denetim (Karar)</td></tr><tr><td></td><td>Çıkış</td></tr><tr><td></td><td>Döngü</td></tr><tr><td></td><td>Akış Yönü</td></tr><tr><td></td><td>Bağlaç</td></tr><tr><td></td><td>Önceden Tanımlı İşlem/Fonksiyon</td></tr></table>	Simge	İşlev		Başla/Bitir		Giriş		Atama/İşlem		Denetim (Karar)		Çıkış		Döngü		Akış Yönü		Bağlaç		Önceden Tanımlı İşlem/Fonksiyon
Simge	İşlev																				
	Başla/Bitir																				
	Giriş																				
	Atama/İşlem																				
	Denetim (Karar)																				
	Çıkış																				
	Döngü																				
	Akış Yönü																				
	Bağlaç																				
	Önceden Tanımlı İşlem/Fonksiyon																				
Elips	Başla ve Bitir adımları için kullanılır. Akış şemasının başlangıç ve bitiş noktasında yer alır.																				
Paralelkenar	Giriş ya da Çıkış işlemleri için kullanılır. Örneğin; klavyeden bir sayı girilmesi istenmesi veya ekrana işlem sonucunun yazdırılması gibi.(Oku, Yaz)																				
Dikdörtgen	Hesaplama ya da Değişkene Değer Atama işlemleri için kullanılır. Örneğin; iki sayıyı topla veya girilen ilk sayıyı A olarak kabul et.																				
Eşkenar Dörtgen	Karşılaştırma, kontrol ya da Karar Verme işlemleri için kullanılır. Örneğin; girilen sayı 5'ten büyük mü?																				
Altıgen	Bir işlemin belli bir sayıda veya belli bir koşul doğru olduğu sürece tekrar edilmesini sağlayan döngü komutunu gösteren şekildir.																				
Kırık Dörtgen	Ekrana veya yazıcıya bilgi/veri yazdırmak için kullanılır.																				



Python Uygulama Kodları 7

ÖRNEK 3.

ÖRNEK 1.

Algoritma	Akış Şeması	Sözde Kod
1. Başla. 2. Notları Oku. 3. Ortalamayı Hesapla. 4. Ortalamayı Yaz. 5. Bitir.		1. Başla. 2. Oku <i>not1, not2</i> 3. $ort = (not1 + not2)/2$ 4. Yaz <i>ort</i> 5. Bitir.

ÖRNEK 2.

Algoritma	Akış Şeması	Sözde Kod
1. Başla. 2. Notları oku. 3. Ortalamayı hesapla. 4. Eğer ortalama ≥ 50 ise "Geçti" yaz. Değilse "Kaldı" yaz. 5. Bitir.		1. Başla. 2. Oku <i>not1, not2</i> 3. $ort = (not1 + not2)/2$ 4. if ($ort \geq 50$) then Yaz "Geçti" else Yaz "Kaldı" 5. Bitir.

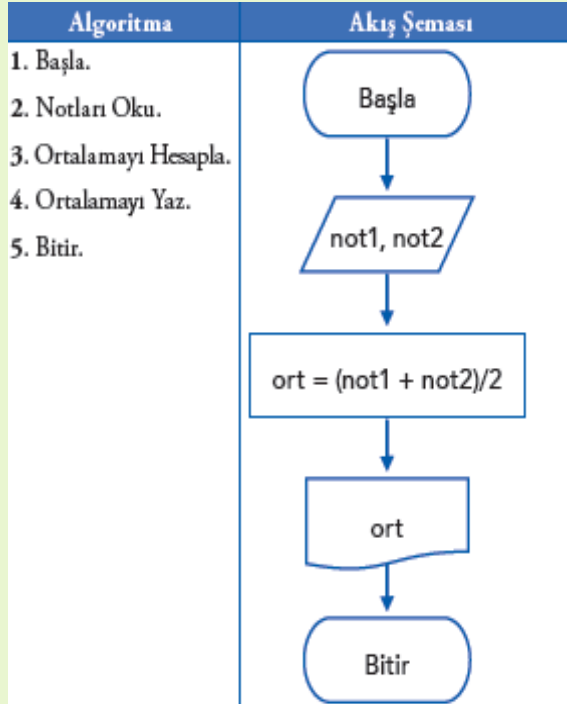
Algoritma	Akış Şeması	Sözde Kod
1. Başla. 2. Sınıf ortalamasını 0'a eşitle 3. Döngü $i \leq 20$ olana kadar dön, $i > 20$ olunca 7. adıma git. 4. Notları oku. 5. Ortalamayı hesapla. 6. Öğrenci ortalamasını toplama aktar. 7. i yi 1 artır. 8. Sınıf ortalamasını hesapla. 9. Sınıf ortalamasını yaz. 10. Bitir.		1. Başla. 2. $topOrt = 0$ 3. for ($i=1$; $i \leq 20$; $i++$) 4. Oku <i>not1, not2</i> 5. $ort = (not1 + not2)/2$ 6. $topOrt = topOrt + ort$ 7. $i++$ 8. $sınıfOrt = topOrt/20$ 9. Yaz <i>sınıfOrt</i> 10. Bitir.



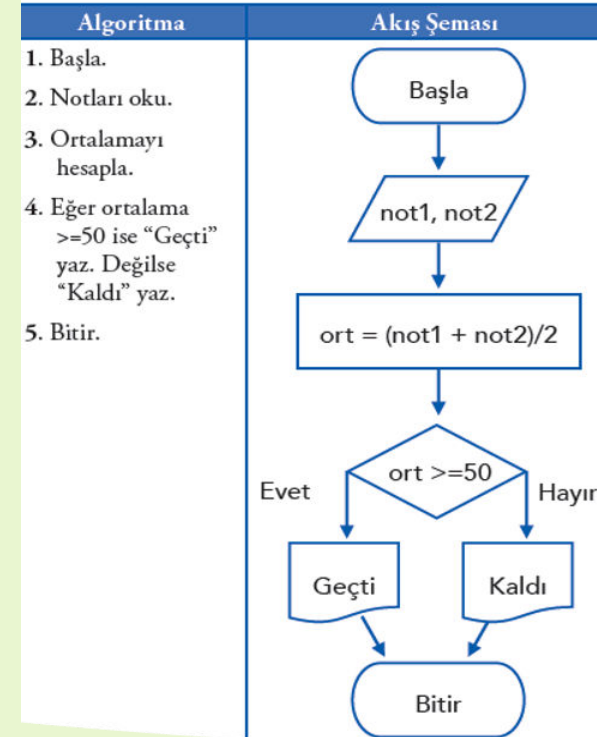
Python Uygulama Kodları 8

İfade ve Kavram	Açıklama
Göstergeler	Bilgisayarlar; problemleri çözmek, işlerimizi kolaylaştırmak, daha hızlı ve etkili çözümler üretmek için kullanılır. Yeterli çözümler üretebilmek için aşağıdaki göstergeleri önemsemek gerekir. - Bütünü, her biri anlamlı işlemler içeren parçalara bölünüz, modülleri kullanınız. - Farklı satırlar arasında bağlantı kurmak yerine mantıksal yapıları kullanınız
Mantıksal Yapılar	Doğrusal yapı, işlemleri sıra ile çalıştırır. Örnek olarak iki sınavın ortalamasını bulmak, Üçgenin, dairenin vb. alanını ve çevresini hesaplamak gösterilebilir. Karar yapısı, iki olasılıktan birini seçmek ve ona göre devam etmek için kullanılır. Örnek olarak iki sınav puanının ortalamasını bularak öğrencinin dersten geçip geçmediğini kontrol etmek gösterilebilir. Döngüsel yapı, bir dizi işlemi tekrarlamak için kullanılır. Örnek olarak 20 öğrencilik bir sınıfın bir dersten aldığı iki not üzerine sınıf ortalamasını hesaplayan yapı görülmektedir. Durumsal yapı, belirli bir duruma göre farklı işlemlerin yapılmasına olanak sağlar. Örnek olarak klavyeden girilen sayıya göre haftanın gününü yazan yapı görülmektedir.

ÖRNEK 1.

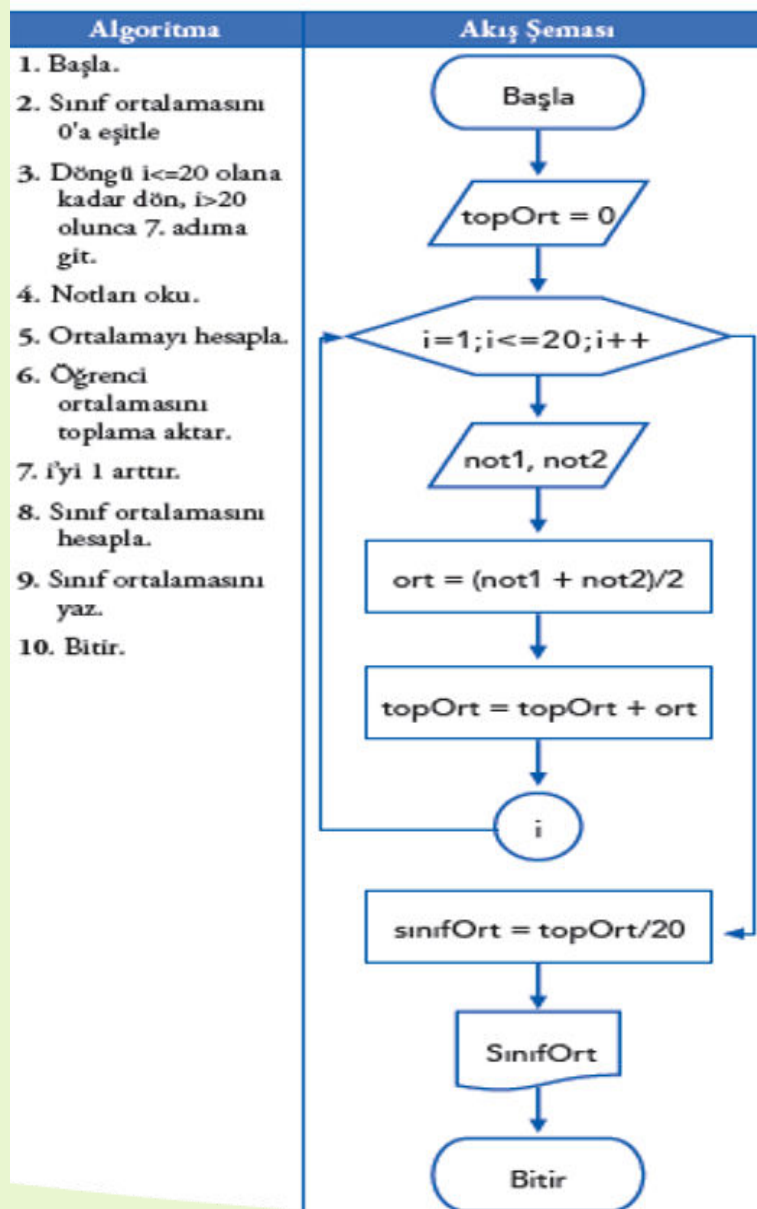


ÖRNEK 2.



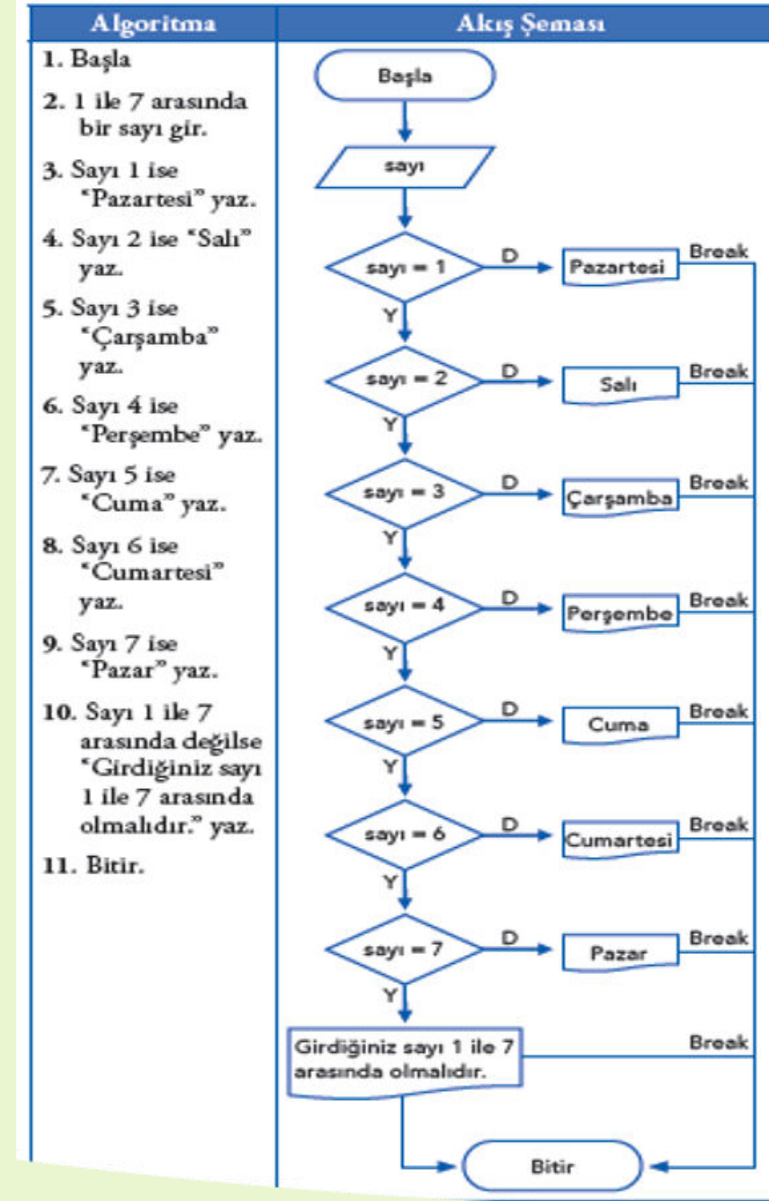


ÖRNEK 3.



Python Uygulama Kodları 8

ÖRNEK 4.





Python Uygulama Kodları 9-Örnekler

- ÖRNEK 1.** Bir internet satış sitesi alışveriş tutarı 50 TL'nin üzerinde ise kargoyu bedava yapıyor. 50 TL altı sipariş tutarına ise 7 TL ilave edilerek, ekrana kargo ücretini ve toplam ücreti yazdırılıyor. Bu problemin algoritmasını yazınız, akış şemasını çizin ve Python kodunu oluşturunuz.
- ÖRNEK 2.** Klavyeden, girilen sıcaklık derecesine göre suyun katı/sıvı/gaz hâllerinden hangisinde olduğunu yazan programın algoritmasını yazınız, akış şemasını çizin ve Python kodunu oluşturunuz.
- ÖRNEK 3.** Bir kredi kartından 2000 TL üzeri alışveriş yapıldığında 100 TL hediye puan, 1000 TL üzeri alışveriş yapıldığında 50 TL hediye puan ,1000 TL ve altında alışveriş yapıldığında 10 TL hediye puan kazanılmaktadır. Bu problemin algoritmasını yazınız, akış şemasını çizin ve Python kodunu oluşturunuz.
- ÖRNEK 4.** Kullanıcıya boyunun kaç cm olduğunu soran ve eğer boyu 170 cm'den az ise "Boyunuz kısa", boyu 170 cm ile 180 cm arasında ise "Boyunuz normal", 180 cm'den fazlaysa "Boyunuz uzun" yazan programın algoritmasını yazınız, akış şemasını çizin ve Python kodunu oluşturunuz.
- ÖRNEK 5.** Kullanıcıdan ortalamasını aldıktan sonra eğer ortalaması 85 ve üzerindeyse "5" , 70 ve üzerindeyse "4" , 55 ve üzerindeyse "3" , 45 ve üzerindeyse "2" , 45'in altındaysa "KALDI" yazan programın algoritmasını yazınız, akış şemasını çizin ve Python kodunu oluşturunuz.
- ÖRNEK 6.** Tiyatro bileti alırken bilet fiyatı yaşa göre değişmektedir. Yaşı 18'den küçük olanlar için bilet ücreti 15 TL; yaşı 18'den büyük ve 65'ten küçük olanlar için 20 TL ve yaşı 65'ten büyük olanlar için 10 TL olarak belirlenmiştir. Bu durumda tiyatro seyircilerinin yaşlarına göre bilet almalarına olanak sağlayan programın algoritmasını yazınız, akış şemasını çizin ve Python kodunu oluşturunuz.
- ÖRNEK 7.** Bir kargo şirketi
- 2 kilograama kadar olan gönderilerin kilogramını 6 TL'ye
 - 2 kilogram ve 6 kilogram arasında olan gönderilerin kilogramını 5 TL'ye
 - 6 kilogram ve 10 kilogram arasında olan gönderileri kilogramını 4 TL'ye
 - 10 kilogram üzerinde olan gönderileri kilogramını 3TL'ye taşımaktadır.
- Buna göre kullanıcıdan gönderinin ağırlığını alıp kilograama göre taşıma fiyatını belirleyen algoritmasını yazınız, akış şemasını çizin ve Python kodunu oluşturunuz.
- ÖRNEK 8.** Bir kişiye ait cinsiyet, boy ve kilo bilgilerini alarak aşağıdaki şartlara göre kişinin ideal kiloda olup olmadığını, ideal kiloda değilse kaç kilo alması veya kaç kilo vermesi gerektiğini bulan algoritmasını yazınız, akış şemasını çizin ve Python kodunu oluşturunuz.
- İdeal Kilo Hesabı: Bayanlarda: Boy-110, Erkeklerde: Boy-108



Python Uygulama Kodları 9-Örnekler

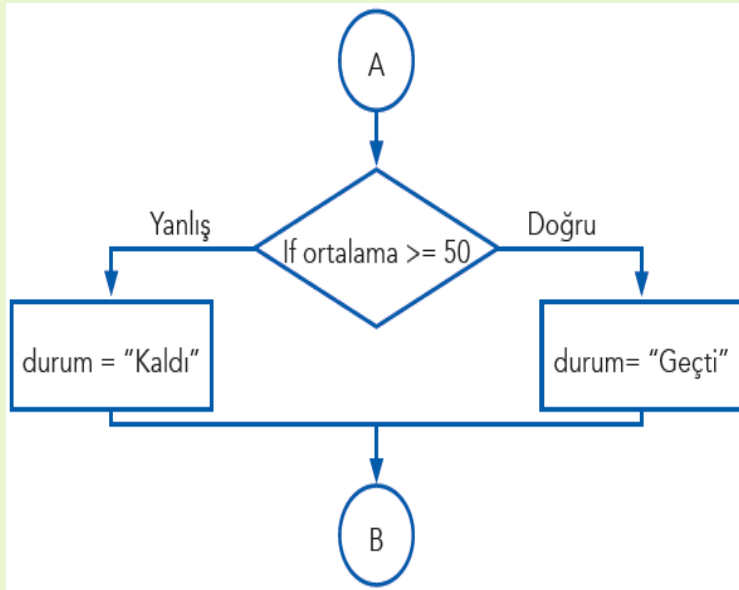
- ÖRNEK 9.** Kullanıcıya Almanca ya da İngilizce ve ofis programları bilip bilmediğini soran, Almanca ya da İngilizceden birini biliyorsa ve ofis programları biliyorsa “İşe başlayabilirsiniz”, değilse “üzgünüz” mesajı veren algoritmasını yazınız, akış şemasını çizin ve Python kodunu oluşturunuz.
- ÖRNEK 10.** Bir mağaza müşterilerine yaptıkları alışveriş tutarına göre indirim yapmaktadır. 200 TL ye kadar olan alışverişler için %10, 200-400 TL arası olan alışverişler için %15, 400 TL den fazla olan alışverişler için %20 olacak şekilde indirim yapıp alışveriş tutarını yazan algoritmasını yazınız, akış şemasını çizin ve Python kodunu oluşturunuz.
- ÖRNEK 11.** Emekli olabilmek için erkekler en az 7800 gün prim yatırmalı ve 58 yaşını tamamlamış olmalıdır. Kadınlar ise en 7200 gün prim yatırmalı ve 55 yaşını tamamlamış olmalıdır. Buna göre kullanıcıya cinsiyet, yaş ve prim gün sayısını girdirip emekli olabilecekse $\text{yaş} * 1000$ formülü ile alacağı ikramiyeyi hesaplayıp emekli olabilirsiniz yazan, olamayacak ise emekli olmayacağını belirten algoritmasını yazınız, akış şemasını çizin ve Python kodunu oluşturunuz.
- ÖRNEK 12.** Bir otoparkın ücret tarifesi şöyledir:
- 0 – 3 saat: 3 TL
 - 3 – 7 saat: 4 TL
 - 7 – 12 saat: 5 TL
 - 12 ve üzeri: 6 TL’dir.
- Buna göre girilen saate göre otoparka ödenecek ücreti hesaplayıp ekrana yazan algoritmasını yazınız, akış şemasını çizin ve Python kodunu oluşturunuz.
- ÖRNEK 13.** Klavyeden, uzunlukları girilen üç doğru parçasının bir üçgen oluşturup oluşturamayacağını hesaplayan algoritmasını yazınız, akış şemasını çizin ve Python kodunu oluşturunuz. (Bir üçgenin iki kenarının uzunluğu toplamı, üçüncü kenardan büyük olmalıdır.)
- ÖRNEK 14.** Bir öğrencinin ortalama ve devamsızlık bilgisine göre geçme/kalma durumunu kontrol eden ve ekrana yazdıran algoritmasını yazınız, akış şemasını çizin ve Python kodunu oluşturunuz.
- ÖRNEK 15.** Klavyeden, girilen ay bilgisine göre kuzey yarım kürede hangi mevsimin yaşandığını ekrana yazdıran algoritmasını yazınız, akış şemasını çizin ve Python kodunu oluşturunuz.



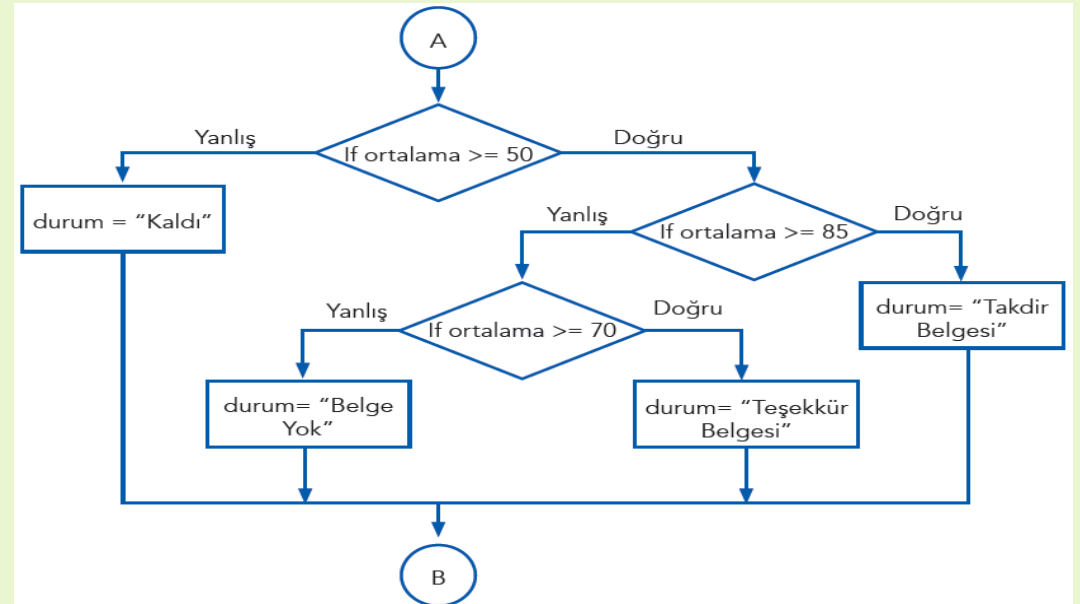
Python Uygulama Kodları 9

İfade ve Kavram	Açıklama
Karar Yapıları	Karar yapıları, bilgisayara <i>iki ya da daha fazla seçenek arasında seçim yapma</i> hakkı tanıyan bir mantık yapısıdır. Eğer karar yapıları olmasaydı bilgisayarlar hızlı bir hesap makinesi olmanın ötesine gidemezdi.
Tek Koşullu Yapılar	Koşulun sağlanıp sağlanmaması durumuna göre programın akışı değişir ve program, karara uygun yönergelerle çalışmaya devam eder. Hiçbir zaman <i>üçüncü bir seçenek olamaz</i> çünkü karar sembolünden yalnızca iki olasılık çıkabilir. Diğer bir ifade ile belirtilen durum <i>ya doğrudur ya da yanlıştır</i> . Örneğin bir öğrencinin puan ortalamasına bakarak Geçme/Kalma durumu.
Çok Koşullu Yapılar	<i>Birden fazla karar içeren</i> algoritmaları yazmak için kullanılacak üç tür karar yapısı vardır: Düz Mantık, Pozitif Mantık ve Negatif Mantık.
Düz Mantık Yapısı	Düz mantık ile çalışan kararlarda <i>bütün koşullar test edilir</i> . Bir koşulun test edilmesi, “Doğru” ya da “Yanlış” sonuç elde etmek için durumun işlenmesidir. Düz mantık çözümlerin içinde en yetersizi, çözüm olarak nitelendirilebilir çünkü bütün koşulların test edilmesi, programın çalışmasını da uzatır.
Pozitif Mantık Yapısı	Düşünme biçimimize en çok benzeyen yapı olması nedeni ile pozitif mantık kullanımı en kolay yapıdır. Bu yapı; kullanıldığında genellikle bilgisayardan, <i>koşulun doğru olması durumunda işlem yapması</i> , yanlış olması durumunda farklı bir karar vermesi beklenir. Böylece daha az adımda karar verilebilir.
Negatif Mantık Yapısı	Negatif mantık kullanıldığında bilgisayardan, <i>koşulun doğru olması durumunda farklı yönergeleri takip etmesi</i> beklenir. Negatif mantık kullanmak, kontrol edilecek koşul sayısını azalttığından programı daha anlaşılır kılarak geliştirir.
İç içe Karar Yapıları	Çoklu karar yapıları içeren algoritmalarda eğer koşullarını iç içe yazmamız gerekebilir. Bu durumda pozitif ve negatif mantık yapıları kullanılabilir; düz mantık yapısı kullanılmaz.

ÖRNEK 1. (TEK KOŞULLU KARAR YAPISI)

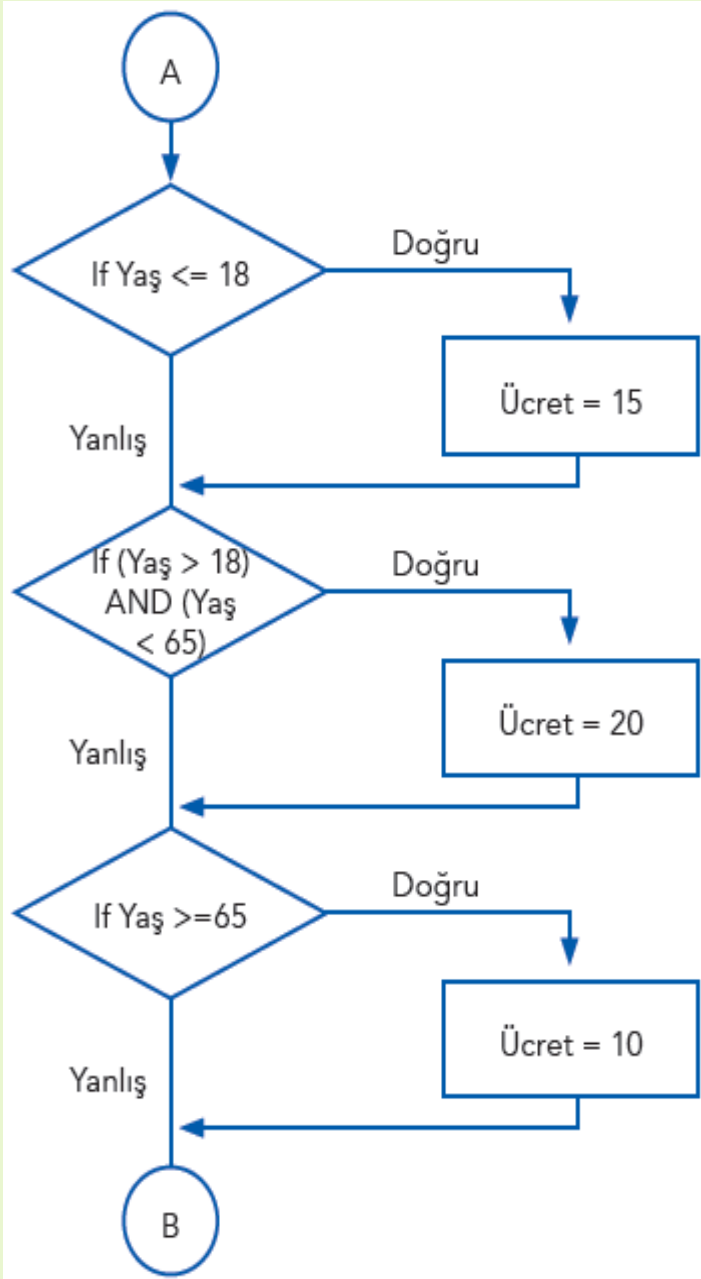


ÖRNEK 2. (İÇ İÇE KARAR YAPILARI)



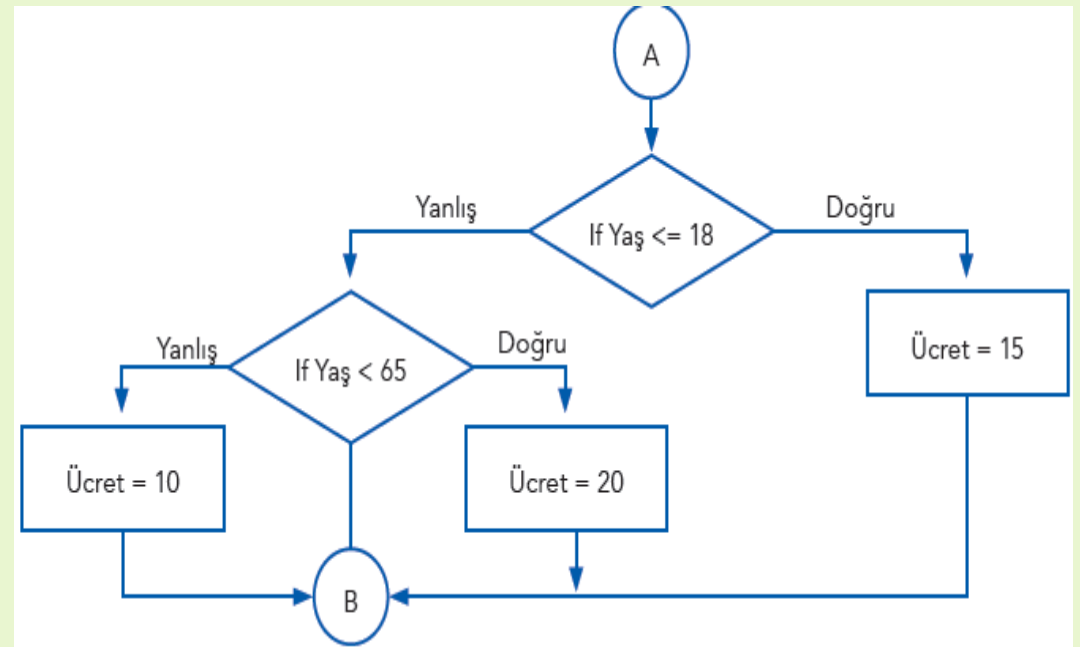


ÖRNEK 3. (DÜZ MANTIK KULLANIMI)

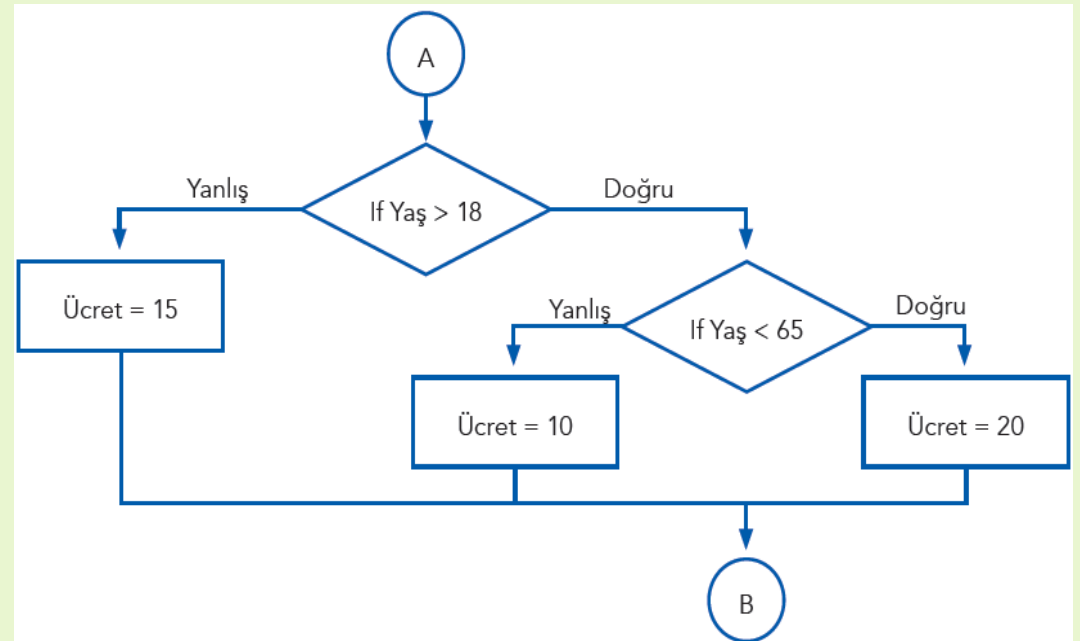


Python Uygulama Kodları 9

ÖRNEK 4. (POZİTİF MANTIK KULLANIMI)

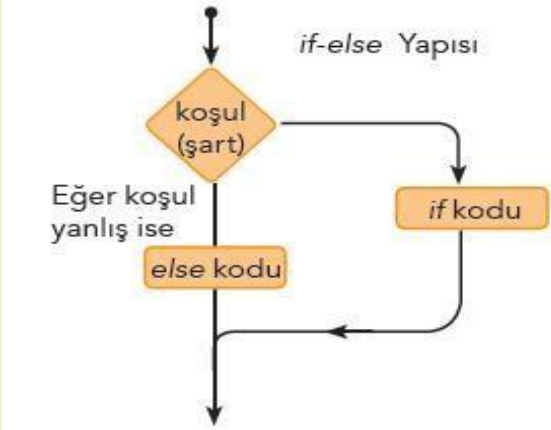


ÖRNEK 5. (NEGATİF MANTIK KULLANIMI)





Python Uygulama Kodları 10

İfade ve Kavram	Açıklama	Örnek
"if else" İfadesi	if Türkçede EĞER anlamına gelir, adından da anlaşılacağı üzere, bir koşula bağlı durumları kontrol etmek amacıyla kullanılır. else Türkçe' de aksi halde anlamına gelir ve şartın gerçekleşmediği durumlarda kullanılır.  if/else ifadesinin yapısı: <pre>if koşul: ifadeler else: ifadeler</pre>	<pre>a=int(input("a sayısı giriniz : ")) b=int(input("b sayısı giriniz : ")) if a<b: print("a sayısı b'den küçüktür") else: print("a sayısı b'den büyüktür")</pre>

ÖRNEK 1.Kullanıcının yaşını isteyerek girdiği değer 18 ve üzeriyse "Reşitsiniz", değilse "Reşit değilsiniz" yazan algoritma, akış şeması ve Python kodunu oluşturunuz.

ÖRNEK 2.Kullanıcının yaşını isteyerek girdiği değer 18 ve üzeriyse "Ehliyet alabilirsiniz", değilse "Ehliyet alamazsınız" yazan algoritma, akış şeması ve Python kodunu oluşturunuz.

ÖRNEK 3.Kullanıcıdan adını alıp, eğer adı 7 karakterden uzunsa "İsminiz uzun", değilse "İsminiz kısa" yazan algoritma, akış şeması ve Python kodunu oluşturunuz.

ÖRNEK 4.Kullanıcıdan bir sayı girmesini isteyerek girdiği sayı çiftse "Girdiğiniz sayı çift", tekse "Girdiğiniz sayı tek" yazan algoritma, akış şeması ve Python kodunu oluşturunuz.

ÖRNEK 5.Kullanıcıdan bir sayı girmesini isteyerek girdiği sayı pozitifse "Girdiğiniz sayı pozitif", negatifse "Girdiğiniz sayı negatif" yazan algoritma, akış şeması ve Python kodunu oluşturunuz.

ÖRNEK 6.Kullanıcıdan bir sıcaklık değeri girmesini isteyerek girdiği değer 25'ten büyükse "Hava Sıcak", 25'ten küçükse "Hava Soğuk" yazan algoritma, akış şeması ve Python kodunu oluşturunuz.

ÖRNEK 7.Kullanıcıdan su sıcaklık değeri girmesini isteyerek girdiği değer 0'dan büyükse "Sıvı", 0'dan küçükse "Katı" yazan algoritma, akış şeması ve Python kodunu oluşturunuz.

ÖRNEK 8.Kullanıcıdan a ve b adında iki sayı alıp bu sayıları karşılaştıran algoritma, akış şeması ve Python kodunu oluşturunuz.



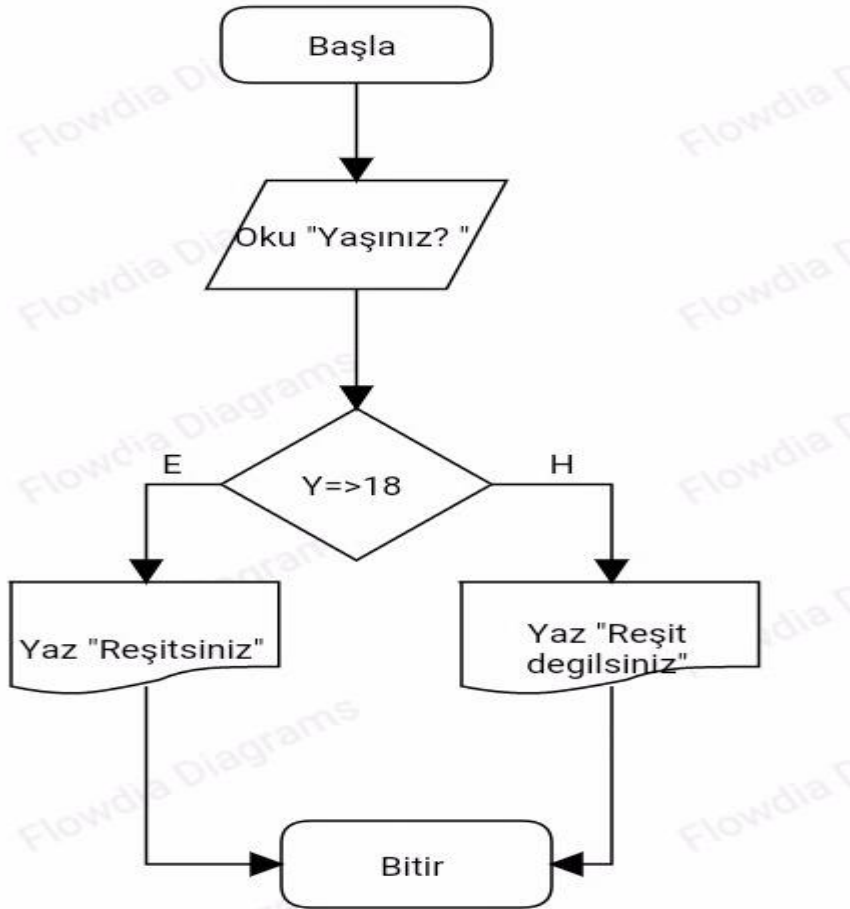
Python Uygulama Kodları 10-Cevaplar

ÖRNEK1) Kullanıcının yaşını isteyerek girdiği değer 18 ve üzeriyse "Reşitsiniz", değilse "Reşit değilsiniz" yazan algoritma, akış şeması ve Python kodunu oluşturunuz.

ALGORİTMA)

- 1) Başla
- 2) Oku "Yaşınız? " (Y)
- 3) Eğer
- 4) ($Y \geq 18$) yaz "Reşitsiniz"
- 5) (DEĞİLSE) yaz "Reşit değilsiniz"
- 6) Bitir

(AKIŞ ŞEMASI)



PYTHON KODU

```
yas=int(input("Lütfen yaşıınızı giriniz : "))
if yas>=18:
    print("Reşitsiniz")
else:
    print("Reşit değilsiniz")
```



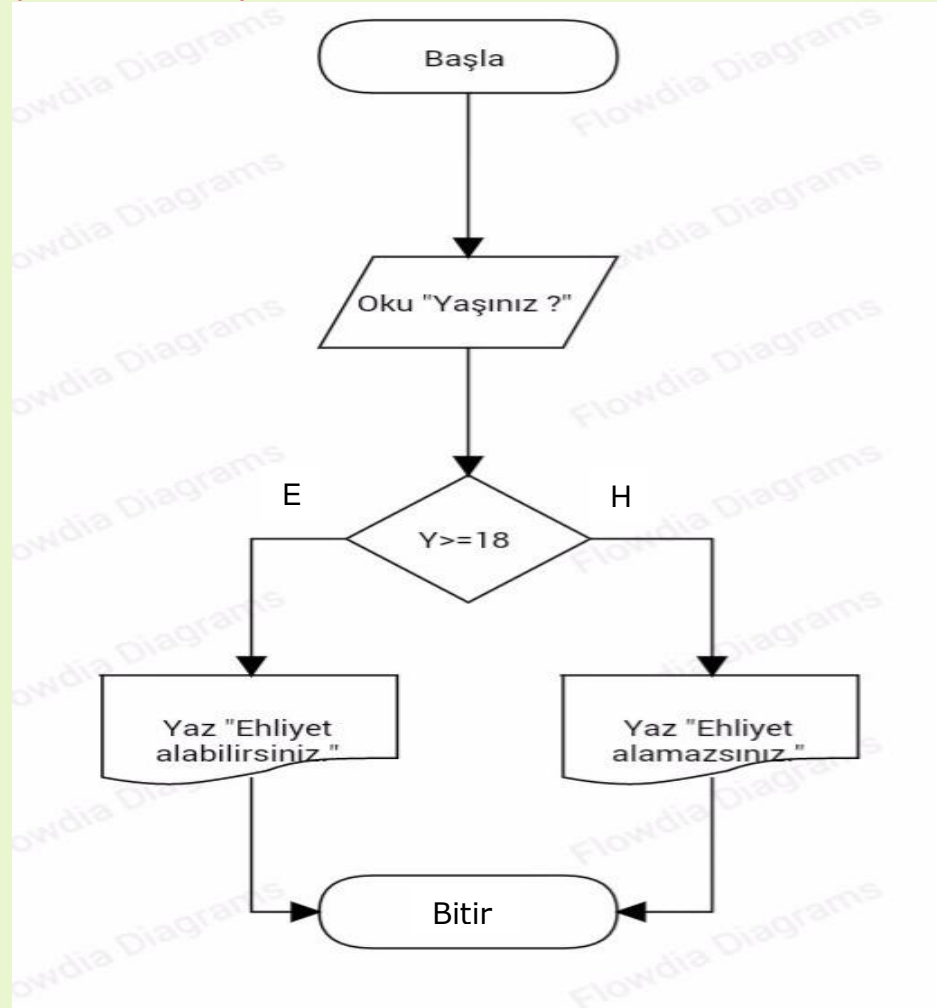
Python Uygulama Kodları 10-Cevaplar

ÖRNEK2) Kullanıcının yaşını isteyerek girdiği değer 18 ve üzeriyse “Ehliyet alabilirsiniz”, değilse “Ehliyet alamazsınız” yazan algoritma, akış şeması ve Python kodunu oluşturunuz.

ALGORİTMA)

- 1) Başla
- 2) Oku "Yaşınız? " (Y)
- 3) Eğer
- 4) ($Y \geq 18$) yaz " Ehliyet alabilirsiniz ."
- 5) (DEĞİLSE) yaz " Ehliyet alamazsınız. "
- 6) Bitir

(AKIŞ ŞEMASI)



PYTHON KODU

```
yas=int(input("Lütfen yaşıınızı giriniz : "))
if yas>=18:
    print("Ehliyet alabilirsiniz")
else:
    print("Ehliyet alamazsınız")
```



Python Uygulama Kodları 10-Cevaplar

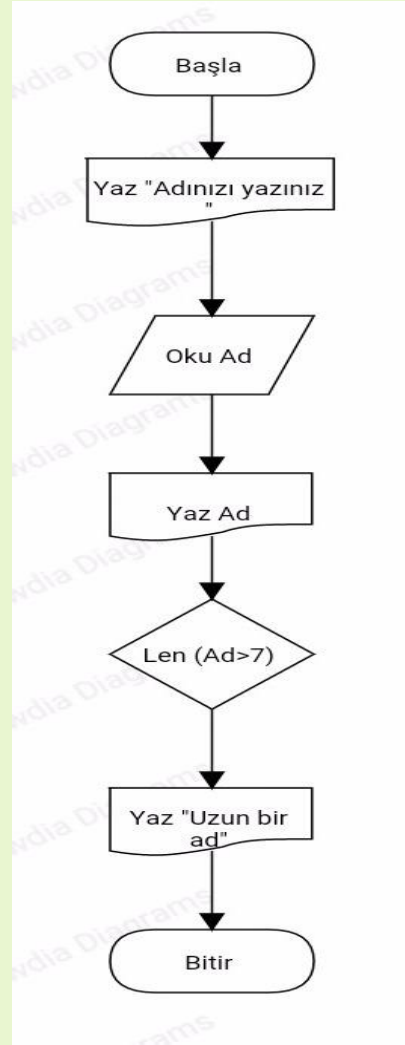
ÖRNEK3)

Kullanıcıdan adını alıp, eğer adı 7 karakterden uzunsa "İsminiz uzun", yazan algoritma, akış şeması ve Python kodunu oluşturunuz.

ALGORİTMA)

- 1) Başla
- 2) Oku Ad
- 3) Yaz Ad
- 4) Eğer Ad uzunluğu 7 karakterden büyükse "Uzun bir ada sahipsin" yaz
- 5) Bitir

(AKIŞ ŞEMASI)



PYTHON KODU

```
isim=input("Lütfen isminizi giriniz ")
if len(isim)>7:
    print("isminiz uzun")
```



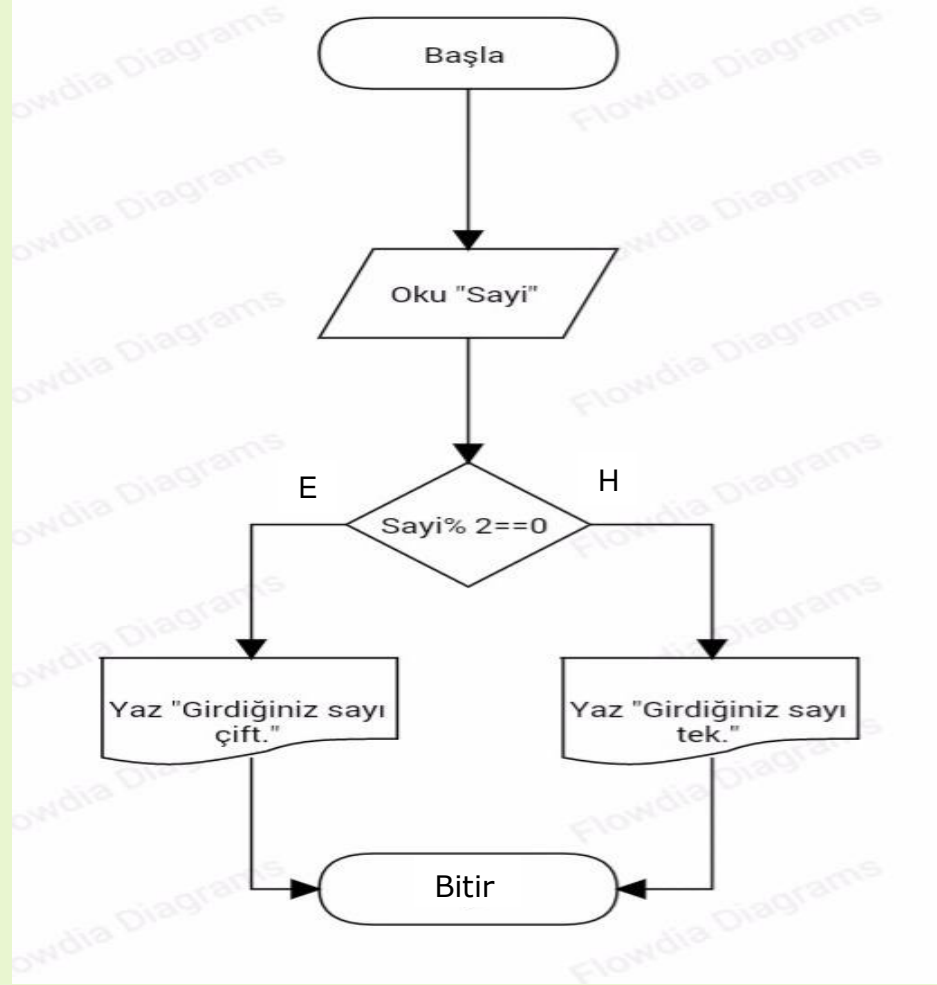
Python Uygulama Kodları 10-Cevaplar

ÖRNEK4) Kullanıcıdan bir sayı girmesini isteyerek girdiği sayı çiftse “Girdiğiniz sayı çift”, tekse “Girdiğiniz sayı tek” yazan algoritma, akış şeması ve Python kodunu oluşturunuz.

ALGORİTMA)

- 1) Başla
- 2) Oku "Sayı? "
- 3) Eğer
- 4) $(\text{Sayi} \% 2 == 0)$ yaz "Sayı çifttir."
- 5) (DEĞİLSE) yaz "Sayı tektir."
- 6) Bitir

(AKIŞ ŞEMASI)



PYTHON KODU

```
sayi = int(input("Bir sayı girin: "))
if sayi % 2 == 0:
    print("Girdiğiniz sayı bir çift sayıdır.")
else:
    print("Girdiğiniz sayı bir tek sayıdır.")
```



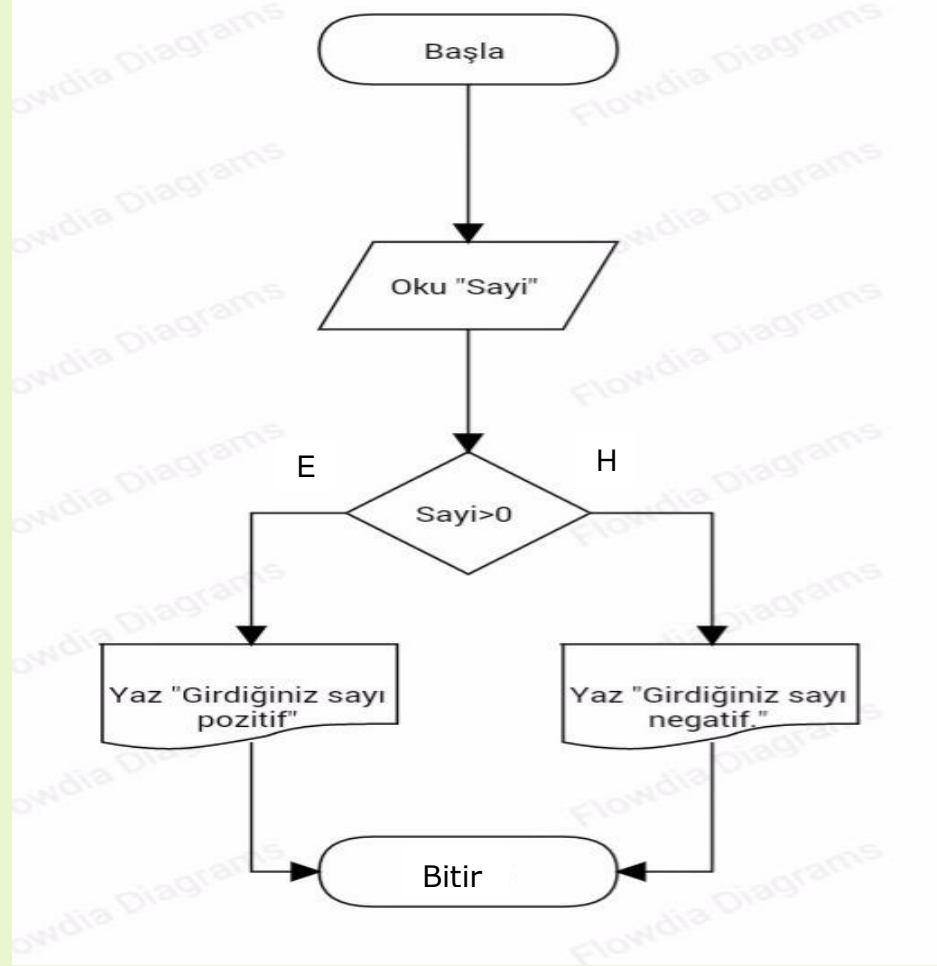

Python Uygulama Kodları 10-Cevaplar

ÖRNEK5) Kullanıcıdan bir sayı girmesini isteyerek girdiği sayı pozitifse “Girdiğiniz sayı pozitif”, negatifse “Girdiğiniz sayı negatif” yazan algoritma, akış şeması ve Python kodunu oluşturunuz.

ALGORİTMA)

- 1) Başla
- 2) Yaz "Bir sayı girin"
- 3) Oku (Sayı)
- 4) Eğer
- 5) (Sayı >0) Yaz ("Girdiğiniz sayı pozitifdir.")
- 6) (Değilse) Yaz ("Girdiğiniz sayı negatiftir.")
- 7) Bitir

(AKIŞ ŞEMASI)



PYTHON KODU

```
sayi=int(input("Lütfen bir sayı giriniz : "))
if sayi>0:
    print("Sayı pozitif")
else:
    print("Sayı negatif")
```



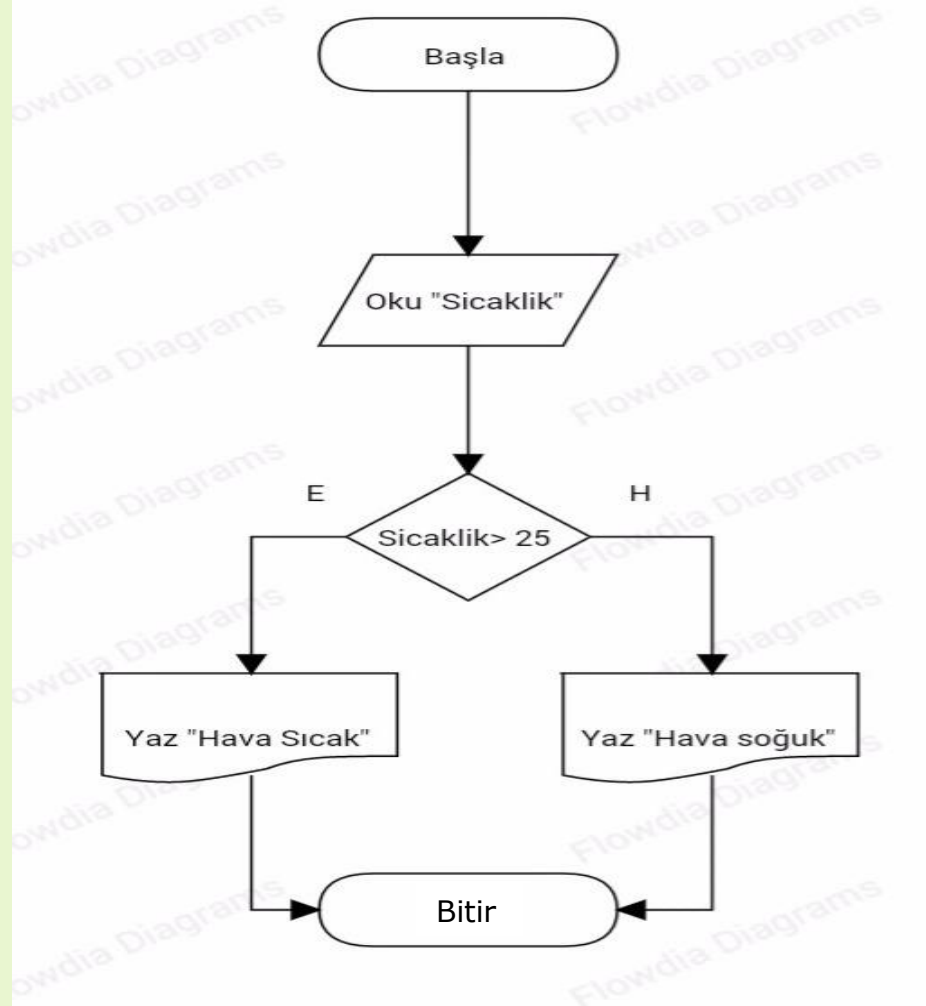
Python Uygulama Kodları 10-Cevaplar

ÖRNEK6) Kullanıcıdan bir sıcaklık değeri girmesini isteyerek girdiği değer 25'ten büyükse "Hava Sıcak", 25'ten küçükse "Hava Soğuk" yazan algoritma, akış şeması ve Python kodunu oluşturunuz.

ALGORİTMA)

- 1) Başla
- 2) Oku "Sicaklik? "
- 3) Eğer
- 4) (Sicaklik >25) yaz "Hava Sıcak"
- 5) (DEĞİLSE) yaz "Hava Soğuk"
- 6) Bitir

(AKIŞ ŞEMASI)



PYTHON KODU

```
sicaklik=int(input("Lütfen bir sıcaklık değeri giriniz : "))
if sicaklik>25:
    print("Hava Sıcak")
else:
    print("Hava Soğuk")
```



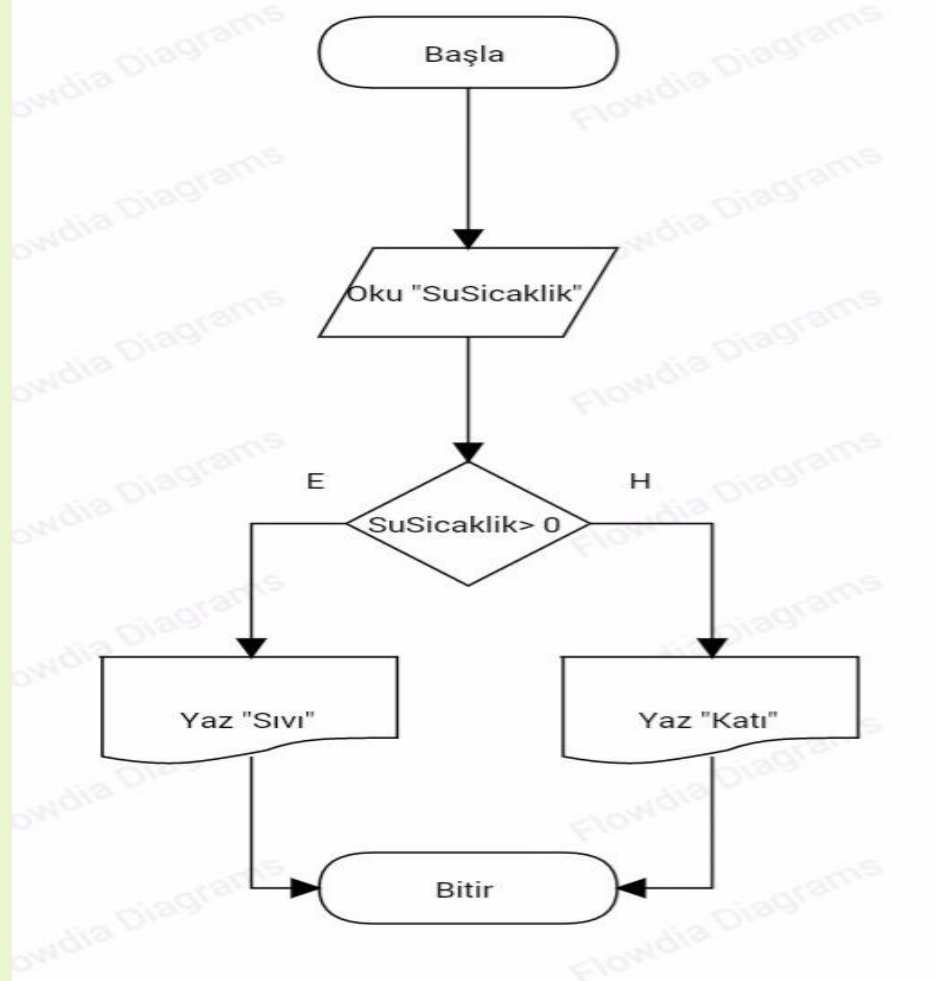
Python Uygulama Kodları 10-Cevaplar

ÖRNEK7) Kullanıcıdan su sıcaklık değeri girmesini isteyerek girdiği değer 0'dan büyükse "Sıvı", 0'dan küçükse "Katı" yazan algoritma, akış şeması ve Python kodunu oluşturunuz.

ALGORİTMA)

- 1) Başla
- 2) Oku "SuSicaklik? "
- 3) Eğer
- 4) (SuSicaklik >0) yaz "Sıvı"
- 5) (DEĞİLSE) yaz "Katı"
- 6) Bitir

(AKIŞ ŞEMASI)



PYTHON KODU

```
susicaklik=int(input("Lütfen bir su sıcaklık değeri giriniz : "))
if susicaklik>0:
    print("Sıvı")
else:
    print("Katı")
```



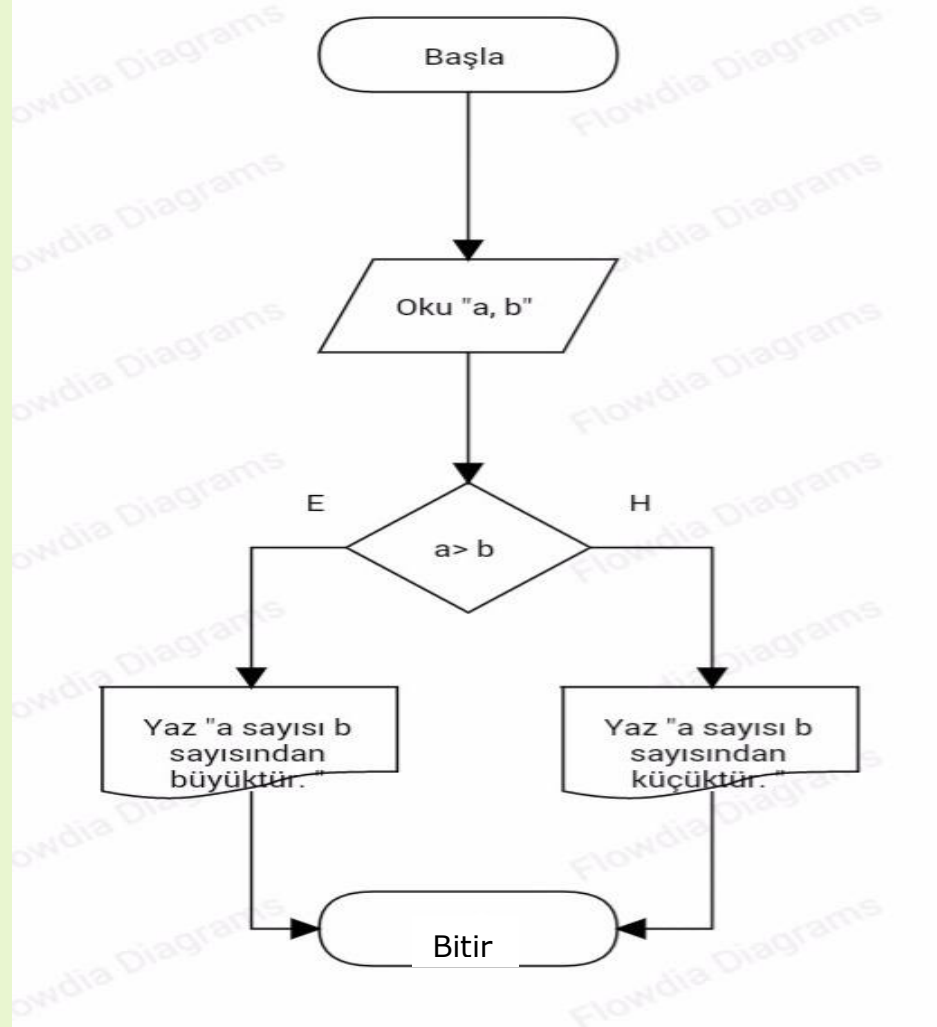
Python Uygulama Kodları 10-Cevaplar

ÖRNEK8) Kullanıcıdan a ve b adında iki sayı alıp bu sayıları karşılaştıran algoritma, akış şeması ve Python kodunu oluşturunuz.

ALGORİTMA)

- 1) Başla
- 2) Oku "a ,b "
- 3) Eğer
- 4) $(a > b)$ yaz "a sayısı b den büyüktür."
- 5) (DEĞİLSE) yaz " a sayısı b den küçüktür."
- 6) Bitir

(AKIŞ ŞEMASI)



PYTHON KODU

```
a=int(input("a sayısı giriniz : "))
b=int(input("b sayısı giriniz : "))
if a<b:
    print("a sayısı b'den küçüktür")
else:
    print("a sayısı b'den büyüktür")
```



Python Uygulama Kodları 11

İfade ve Kavram	Açıklama	Örnek
"if elif" İfadesi	Çok yönlü karar ifadelerinde daha fazla koşulun gerçekleşme durumuna göre işlem yapılır. Koşullardan biri doğru ise ilgili dal yürütülür ve cümlelerin işlevi biter. Eğer birden fazla koşul doğru ise sadece ilk karşılaşılan doğru dal çalışır.	<pre>puan=int(input("Ortalamanızı giriniz : ")) if puan>=85: print("5") elif puan>=70: print("4") elif puan>=55: print("3") elif puan>=45: print("2") elif puan<45: print("1")</pre>

ÖRNEK 1. Kullanıcıya boyunun kaç cm olduğunu soran ve eğer boyu 170 cm'den az ise "Boyunuz kısa", boyu 170 cm ile 180 cm arasında ise "Boyunuz normal", 180 cm'den fazlaysa "Boyunuz uzun" yazan programın algoritmasını yazınız, akış şemasını çizin ve Python kodunu oluşturunuz.

```
boy=int(input("Lütfen boyunuzu giriniz(cm) : "))
if boy>=180:
    print("Boyunuz uzun.")
elif boy>=170:
    print("Boyunuz normal.")
else:
    print("Boyunuz kısa.")
```

ÖRNEK 2. Kullanıcıdan ortalamasını alarak ;ortalama 85 ve üzeriyse "TAKDİR BELGESİ", 70 ve üzeriyse "TEŞEKKÜR BELGESİ", 70'in altındaysa "BELGE YOK" yazan programın algoritmasını yazınız, akış şemasını çizin ve Python kodunu oluşturunuz.

```
puan = int(input("ORTALAMANIZ : "))
if puan>=85 :
    print("TAKDİR BELGESİ")
elif puan>=70 :
    print("TEŞEKKÜR BELGESİ")
elif puan<70 :
    print(" BELGE YOK!")
```

ÖRNEK 3. Kullanıcıdan sıcaklık değerini alarak ;sıcaklık 0'dan küçükse "Su katıdır (BUZ)", 0 ve üzeriyle 100 arasındaysa "Su sıvıdır", bunların dışındaysa "Su gazdır (BUHAR)" yazan programın algoritmasını yazınız, akış şemasını çizin ve Python kodunu oluşturunuz.

```
t = int(input("sıcaklık giriniz: "))
if t<0:
    hal = "Su katıdır (BUZ) "
elif t>=0 and t<100:
    hal = "Su sıvıdır"
else:
    hal = "Su gazdır (BUHAR) "
print(hal)
```

ÖRNEK 4. Kullanıcıdan cinsiyetini ve yaşını alıp eğer kadınsa "Kadınlar için zorunlu askerlik yoktur.", erkekse yaşı 20'den büyükse "Askere gitmelisin.", 20'den küçükse "Henüz askerlik yaşı gelmemiş." yazan programın algoritmasını yazınız, akış şemasını çizin ve Python kodunu oluşturunuz.

```
cinsiyet=input("lütfen cinsiyetinizi giriniz[E/K]:")
yas=int(input("lütfen yaşıınızı giriniz"))
if (cinsiyet == "K"):
    print("KADINLAR İÇİN ZORUNLU ASKERLİK YOKTUR")
elif (cinsiyet == "E" and yas >= 20):
    print("ASKERE GİTMELİSİN")
elif (cinsiyet == "E" and yas < 20):
    print("HENÜZ ASKERLİK YAŞIN GELMEMİŞ")
```



Python Uygulama Kodları 12

İfade ve Kavram	Açıklama	Örnek
for döngüsü	for döngüleri belirli sayıda işlemlerin tekrarlanması için kullanılan döngülerdir. for döngüleri başlangıç ve bitiş değerleri arasında artım miktarına göre istenilen sayıda tekrar yapar.	for n in range(1,6): print(n)
range (başlangıç değeri, son değeri, arttırma/azaltma değeri) :	Başlangıç değeri: Döngü değişkeninin alacağı ilk değerdir. Eğer boş bırakılırsa 0 olarak belirlenir. Son değeri: Döngü değişkeninin bitiş değeridir. Boş bırakılmamalıdır. Arttırma/azaltma değeri: Döngü değişkeninin artırma veya azaltma miktarını belirler. Eğer boş bırakılırsa, 1 olarak belirlenir.	range(10) → 0,1,2,3,4,5,6,7,8,9 range(1, 10) → 1,2,3,4,5,6,7,8,9 range(1, 10, 2) → 1,3,5,7,9 range(10, 0, -1) → 10,9,8,7,6,5,4,3,2,1 range(10, 0, -2) → 10,8,6,4,2 range(2, 11, 2) → 2,4,6,8,10 range(-5, 5) → -5,-4,-3,-2,-1,0,1,2,3,4 range(1, 2) → 1

ÖRNEK 1.

```
for n in range(1,11):  
    print(n)
```

ÖRNEK 2.

```
for n in range(1,11,2):  
    print(n)
```

ÖRNEK 3.

```
for n in range(20,0,-2):  
    print(n)
```

ÖRNEK 4.

```
for n in range(30):  
    print(n)
```

ÖRNEK 5.

```
for a in range(5):  
    print("Merhaba Dünya")
```

ÖRNEK 6.

```
for n in range(21, 0, -3):  
    print(n, end=" ")
```

ÖRNEK 7.

```
toplam= 0  
for i in range(1, 100):  
    toplam+= i  
print(toplam)
```

ÖRNEK 8.

```
kelime= input("Bir kelime yaz: ")  
for harf in kelime:  
    print (harf)
```

ÖRNEK 9.

```
sayi = int(input("Lütfen tablo ölçüsünü giriniz: "))  
for satir in range(1, sayi+1):  
    print("Satır #", satir)
```

ÖRNEK 10.

```
for sayi in range(1,11):  
    kare=sayi**2  
    print("Sayı : ",sayi,"Karesi : ",kare)
```



Python Uygulama Kodları 13

İfade ve Kavram	Açıklama	Örnek
İç içe döngüler	Bir döngü yapısının içine başka bir döngü yapısının yerleştirilmesi ile elde edilen yapıdır. İç içe döngülerde önce içteki çevrimler tamamlanması gerektirir. İç içe döngülerde en içteki çevrim en önce tamamlanır.	for i in range(1,4): print(i) for j in range(1,4): print(j)
break ifadesi	Döngüyü sonlandıran bir komuttur.	
continue ifadesi	continue komutunun görevi kendisinden sonra gelen her şeyin es geçilip döngünün başa dönmesini sağlamaktır.	
pass ifadesi	Herhangi bir işlem yapmadan geçeceğimiz durumlarda kullanılır. Kısaca “Hiçbir şey yapmadan yola devam et!” anlamı katar.	

ÖRNEK 1.

```
for i in range(1,4):  
    print(i)  
    for j in range(1,4):  
        print(j)
```

ÖRNEK 2.

```
for i in range(3):  
    print ("a")  
    for j in range(3):  
        print ("b")
```

ÖRNEK 3.

```
for i in range(3):  
    print ("a")  
    for j in range(3):  
        print ("b")  
        continue
```

ÖRNEK 4.

```
for i in range(3):  
    print ("a")  
    for j in range(3):  
        print ("b")  
        break
```

ÖRNEK 5.

```
for i in range(3):  
    print ("a")  
    break  
    for j in range(3):  
        print ("b")  
        break
```

ÖRNEK 6.

```
for i in range(3):  
    print ("a")  
    continue  
    break  
    for j in range(3):  
        print ("b")  
        break  
        continue
```

ÖRNEK 7.

```
a=0  
for i in range(10):  
    a=a+1  
for j in range(10):  
    a=a+1  
print (a)
```

ÖRNEK 8.

```
a=0  
for i in range(10):  
    a=a+1  
    for j in range(10):  
        a=a+1  
print (a)
```

ÖRNEK 9.

```
for satir in range(1,11):  
    for sutun in range(1,11):  
        print(satir*sutun, '\t', end=" ")  
    print("\n")
```



Python Uygulama Kodları 14

İfade ve Kavram	Açıklama	Örnek
while döngüsü	While ifadesi, ilgili kod satırlarının çalıştırılıp çalıştırılmayacağını belirler. Şart doğru olduğu sürece kod blokunu tekrar tekrar çalıştırır. Şart yanlış olduğunda ise döngü sonlanır. Türkçede '... iken, ... olduğu sürece' gibi anlamlarına gelir.	<pre>n = 1 while n <= 10: print(n) n += 1</pre>
Sonsuz döngü	Sonsuz döngü çeşitli sebeplerle sonsuza kadar döngüye giren ve sonlanamayan bilgisayar programı komutu parçalarına verilen addır. Bu sebepler döngünün; bir sonlandırıcı koşulun bulunmaması, bulunsa da hiçbir zaman sağlanamayacak olması ya da bu koşulun döngünün her seferinde yeniden başlamasına neden olması olabilir.	<pre>a = 1 while a < 10: print("Ali")</pre>
while True:	Aksi belirtilmediği sürece programı çalıştır. Bu haliyle programımızı çalıştırsak sonsuz döngüye girmiş oluruz. Ancak ; break komutunu ilave edersek programa dur komutu vermiş oluruz.	<pre>while True: print("Bilgisayar çıldırdı!") break</pre>
Belirli Döngü	Döngünün kaç defa döneceği belirlidir.	<pre>n = 1 while n <= 10: print(n) n += 1</pre>
Belirsiz Döngü:	Döngünün kaç defa döneceği belirsizdir. Döngünün kaç defa döneceğine şartlar ve kullanıcı karar verir.	<pre>n = 1 karar= int(input("sayılar kaç kadar sıralansın?")) while n <= karar: print(n) n += 1</pre>



Python Uygulama Kodları 14

ÖRNEK 1.

```
a = 1
while a < 10:
    a += 1
    print("Ali", end=" ")
```

ÖRNEK 2.

```
a=1
while a<5:
    print(a, end=" ")
    a+=1
```

ÖRNEK 3. (Sonsuz Döngü örneği Ctrl+C ile çık.)

```
a=1
while a<2:
    print("Merhaba Dünya")
```

ÖRNEK 4.

```
a = 0
while a < 16:
    a += 1
    if a % 2 == 0:
        print(a, end=" ")
```

ÖRNEK 5.

```
while True:
    a=int(input("sayı gir: "))
    b=int(input("sayı gir: "))
    print("Sayıların toplamı:", a+b)
```

ÖRNEK 6.

```
i,x=0,3
while i<3:
    x+=1
    i+=1
print(x)
```

ÖRNEK 7.

```
i=10
while i>0:
    print(i)
    i-=1
    if i==5:
        i=0
print("SON")
```

ÖRNEK 8.

```
n = 1
karar= int(input("sayılar kaç'a kadar sıralansın?"))
while n <= karar:
    print(n)
    n += 1
```

ÖRNEK 9.

```
faktoriyel=1
sayac=1
sayi=int(input("Lütfen bir sayı giriniz.."))
while sayac<=sayi:
    faktoriyel*=sayac
    sayac+=1
print(sayi, " sayısının faktöriyeli:", faktoriyel)
```

ÖRNEK 10.

```
sayi = 0
toplam = 0
print("Bir sayı giriniz, negatif sayı döngüyü sonlandırır:")
while sayi >= 0:
    sayi= int(input())
    toplam += sayi
print("Toplam=", toplam)
```



Python Uygulama Kodları 15

İfade	Görevi	Örnek
Fonksiyon Nedir?	- Fonksiyon, belirli sayıda verileri kullanarak bunları işleyen ve bir sonuç üreten komut grubudur. - Fonksiyonlar programlamada kullanacağımız ve sürekli tekrar eden komutların kullanımını daha da kolaylaştırmak için oluşturacağımız yapılardır. - Bir fonksiyon, tekrar kullanılabilen kod parçacığıdır. - Hazır fonksiyonlar olduğu gibi kendimizin hazırlayabileceği fonksiyonlar da vardır.	<pre>def topla(a,b): sonuc=a+b return sonuc >>> topla(3,8) 11 def selamla(): print("Merhaba Dünya!") selamla()</pre>
Fonksiyon Tanımlama	Kullanıcı tarafında başlayan bir fonksiyon def sözcüğü ile başlar ve aşağıda yazılan formda devam eder. <pre>def fonksiyonismi(parametre listesi): gövde return()</pre> İsmi: Her fonksiyonun, nasıl bir işlem yapılacağını ifade eden bir adı vardır. Parametreler: Bir fonksiyon belli sayıda parametre ile çağrılır ve her birinin doğru türde olması gerekir. Beklenenden daha az ya da çok sayıda parametre göndermek hataya neden olur. Sonuç Türü: Fonksiyon kendini çağıran programa bir değer döndürür.	<pre>def topla(a,b): sonuc=a+b return sonuc</pre>
Fonksiyon Çağırma	Bir fonksiyonu tanımladıktan sonra çağırmak için, ismini yazıp varsa gerekli parametreleri parantez içinde belirtmek gerekir. Parametre almayan bir fonksiyon çağrılırken parametre parantezlerinin içerişi boş bırakılır.	<pre>topla(a,b) selamla()</pre>
Fonksiyonların Değer Döndürmesi	Fonksiyonlar belirli bir işi yerine getirdikten sonra ürettikleri sonucu kendi içlerinde kullanıp çalışmalarını tamamlayabilirler. Örneğin topla fonksiyonunu ele alalım: <pre>def topla(a,b): sonuc=a+b print(sonuc) def topla(a,b): sonuc=a+b return(sonuc)</pre> Fonksiyon kendine parametre olarak gelen iki sayıyı toplar ve sonucu kendi içerisinde print komutuyla ekrana yazdırır. Ekrana yazma işini fonksiyon içerisinde yapmazsak üretilen sonucu görüntülememiz mümkün olmayacaktır. Ancak fonksiyon çağrıldığında geriye değer döndürür. Bu işlemi return sözcüğüyle yapabiliriz.	
Varsayılan parametrelili fonksiyonlar	Bu tür fonksiyonlarda fonksiyon çağrıldığında parametre değeri gönderilmezse, fonksiyon varsayılan değeri döndürür. Bu varsayılan değeri, yandaki örnekte olduğu gibi atama operatörü = ile atayabilirsiniz:	<pre>def topla(a,b=2): return a+b >>> topla(5) 7</pre>



Python Uygulama Kodları 15

Birden çok değeri geri döndüren fonksiyon

Bir fonksiyon birden çok ara sonuç üretebilir ve bunların ayrı ayrı geri döndürülmesi gerekebilir. Kendine parametre olarak gelen iki değişkenin hem toplam hem de çarpım sonuçlarını geri döndüren bir fonksiyon tasarlayalım.

```
def ikiislem(a,b):  
    toplam=a+b  
    carpim=a*b  
    return toplam,carpim
```

ÖRNEK 1.

```
def selamla():  
    print("Merhaba Dünya!")  
selamla()
```

ÖRNEK 2.

```
def selamla(isim):  
    print("Merhaba:",isim)  
selamla("Ali")  
selamla("Ayşe")
```

ÖRNEK 3.

```
def double(a):  
    return a*2  
x = double(5)  
print(x)
```

ÖRNEK 4.

```
def Alan(a):  
    return a*a  
hesapla = Alan(5)  
print(hesapla)
```

ÖRNEK 5.

```
def karesinial(n):  
    return n*n  
a=karesinial(5)  
print(a)
```

ÖRNEK 6.

```
def carp(x, y):  
    return x * y  
print(carp(5, 15))
```

ÖRNEK 7.

```
def say():  
    for i in range(1, 11):  
        print(i, end=" ")  
    print()  
print("10'a kadar sayılıyor. . .")  
say()  
print("Tekrar 10'a kadar sayılıyor. . .")  
say()
```

ÖRNEK 8.

```
def say_n(n):  
    for i in range(1, n + 1):  
        print(i, end=" ")  
    print()  
print("10'a kadar sayılıyor. . .")  
say_n(10)  
print("5'e kadar sayılıyor. . .")  
say_n(5)
```

ÖRNEK 9.

```
def gerisayim(n=5):  
    for sayac in range(n, -1, -1):  
        print(sayac)  
gerisayim()  
gerisayim(8)  
gerisayim(3)
```

ÖRNEK 10.

```
def AralikTopla(n, m=100):  
    toplam = 0  
    for deger in range(n, m + 1):  
        toplam += deger  
    print(toplam)  
AralikTopla(6,10)  
AralikTopla(1,1000)
```



Python Uygulama Kodları 13

İfade	Görevi	Örnek																														
Fonksiyon ve Modüller	Bir Python modülü Python kodları içeren bir dosyadır. Dosyanın adı modülün adına işaret eder. Örneğin math.py isimli bir dosya standart matematik modülünde yer alan fonksiyonları içerir.																															
Modülden fonksiyon ekleme	from modül adı import fonksiyon adı	from math import sqrt, log10, cos																														
Modülün tamamını ekleme	import modül adı	import math y=math.sqrt(x)																														
Rastgele değer üreten fonksiyonlar	random 0 ile 1 arasında rastgele ondalıklı bir değer döndürür. randrange Belirli bir aralıkta rastgele tam sayı bir değer döndürür. randint Belirli bir aralıkta rastgele tam sayı bir değer döndürür. uniform Belirli bir aralıkta ondalıklı bir değer döndürür. choice Değerler arasından rastgele bir değer seçer. sample Değerler arasından istediğimiz miktarda değer seçer																															
Standart matematik fonksiyonları	<table><tr><th>Fonksiyon Adı</th><th>Açıklama</th><th>Örnek Kullanım ve Çıktı</th></tr><tr><td>sqrt()</td><td>İstenilen değerın karekök'ünün bulunmasını sağlar.</td><td>math.sqrt(16) → 4</td></tr><tr><td>exp()</td><td>e (Euler sabiti) sayısının istenilen kuvvetinin alınmasını sağlar.</td><td>math.exp(2) → 7.389056</td></tr><tr><td>log()</td><td>log(x,y) fonksiyonumuz iki parametre alır. İlk parametremiz olan x logaritması, alınacak sayıyı; ikinci parametre olan y taban sayısını temsil etmektedir.</td><td>math.log(2,2) → 1.0</td></tr><tr><td>log10()</td><td>log(x,y) fonksiyonundan tek farkı taban olarak 10 sayısının sabit olmasıdır.</td><td>math.log10(10) → 1.0</td></tr><tr><td>cos()</td><td>cos(x), x derecesinin kosinüs değerini verir.</td><td>math.cos(45) → 0.5253</td></tr><tr><td>pow()</td><td>pow(x,y) fonksiyonu x sayısının y. kuvvetinin alınmasını sağlar.</td><td>math.pow(2,2) → 4</td></tr><tr><td>degress()</td><td>degress(x) fonksiyonu x açısını radyandan dereceye çevirmeye yarar.</td><td>math.degress(45) → 2578.310078088</td></tr><tr><td>radians()</td><td>radians(x) fonksiyonu x açısını dereceden radyana çevirmeye yarar.</td><td>math.radians(45) → 0.7853981633</td></tr><tr><td>fabs()</td><td>fabs(x) fonksiyonu x değerinin mutlak değerinin alınması işlemini gerçekleştirir.</td><td>math.fabs(-5) → 5.0</td></tr></table>	Fonksiyon Adı	Açıklama	Örnek Kullanım ve Çıktı	sqrt()	İstenilen değerın karekök'ünün bulunmasını sağlar.	math.sqrt(16) → 4	exp()	e (Euler sabiti) sayısının istenilen kuvvetinin alınmasını sağlar.	math.exp(2) → 7.389056	log()	log(x,y) fonksiyonumuz iki parametre alır. İlk parametremiz olan x logaritması, alınacak sayıyı; ikinci parametre olan y taban sayısını temsil etmektedir.	math.log(2,2) → 1.0	log10()	log(x,y) fonksiyonundan tek farkı taban olarak 10 sayısının sabit olmasıdır.	math.log10(10) → 1.0	cos()	cos(x), x derecesinin kosinüs değerini verir.	math.cos(45) → 0.5253	pow()	pow(x,y) fonksiyonu x sayısının y. kuvvetinin alınmasını sağlar.	math.pow(2,2) → 4	degress()	degress(x) fonksiyonu x açısını radyandan dereceye çevirmeye yarar.	math.degress(45) → 2578.310078088	radians()	radians(x) fonksiyonu x açısını dereceden radyana çevirmeye yarar.	math.radians(45) → 0.7853981633	fabs()	fabs(x) fonksiyonu x değerinin mutlak değerinin alınması işlemini gerçekleştirir.	math.fabs(-5) → 5.0	
Fonksiyon Adı	Açıklama	Örnek Kullanım ve Çıktı																														
sqrt()	İstenilen değerın karekök'ünün bulunmasını sağlar.	math.sqrt(16) → 4																														
exp()	e (Euler sabiti) sayısının istenilen kuvvetinin alınmasını sağlar.	math.exp(2) → 7.389056																														
log()	log(x,y) fonksiyonumuz iki parametre alır. İlk parametremiz olan x logaritması, alınacak sayıyı; ikinci parametre olan y taban sayısını temsil etmektedir.	math.log(2,2) → 1.0																														
log10()	log(x,y) fonksiyonundan tek farkı taban olarak 10 sayısının sabit olmasıdır.	math.log10(10) → 1.0																														
cos()	cos(x), x derecesinin kosinüs değerini verir.	math.cos(45) → 0.5253																														
pow()	pow(x,y) fonksiyonu x sayısının y. kuvvetinin alınmasını sağlar.	math.pow(2,2) → 4																														
degress()	degress(x) fonksiyonu x açısını radyandan dereceye çevirmeye yarar.	math.degress(45) → 2578.310078088																														
radians()	radians(x) fonksiyonu x açısını dereceden radyana çevirmeye yarar.	math.radians(45) → 0.7853981633																														
fabs()	fabs(x) fonksiyonu x değerinin mutlak değerinin alınması işlemini gerçekleştirir.	math.fabs(-5) → 5.0																														



Python Uygulama Kodları 13

ÖRNEK 1.

```
from math import sqrt, log10, cos
print(sqrt(16))
print(log10(56))
print(cos(45))
```

ÖRNEK 2.

```
from math import sqrt
sayi = float(input("Sayı Giriniz: "))
kok = sqrt(sayi)
print(sqrt(sayi))
print(sayi, " sayısının karekökü" "=", kok)
```

ÖRNEK 3.

```
from random import random
random()
print(random())
```

ÖRNEK 4.

```
from random import random
random()
print(random()*10)
```

ÖRNEK 5.

```
import math
x = math.sqrt(16)
print(x)
y = math.log(8,2)
print(y)
```

ÖRNEK 6.

```
from random import randrange
for i in range(6):
    print(randrange(1, 50), end=" ")
print()
```

ÖRNEK 7.

```
from math import sqrt
y = sqrt(sqrt(256.0))
print(y)
```

ÖRNEK 8.

```
from math import pow
z = pow(pow(2,4),2)
print(z)
```

ÖRNEK 9.

```
import random
a=random.randint(0,100)
print(a)
```

ÖRNEK 10.

```
import random
a=random.randrange(0,100,10)
print(a)
```

ÖRNEK 11.

```
import math
def dikucgenalan():
    a = float(input("Dik üçgenin A kenarını giriniz: "))
    b = float(input("Dik üçgenin B kenarını giriniz: "))
    c = math.hypot(a, b)
    print("Dik üçgenin Hipotenüsü: ", c)
dikucgenalan()
```

ÖRNEK 12.

```
import random
okul="Metin Koloğlu Mesleki ve Teknik Anadolu Lisesi"
a=random.choice(okul)
print(a)
```




Python Uygulama Kodları 17

LİSTELER

Listeler yapıları gereği stringlere oldukça benzerler. Tıpkı stringler gibi ,indekslenirler,parçalanırlar ve üzerinde değişik işlemler yapabildiğimiz metodlar bulunur. Ancak listelerin stringlerden önemli farkları da bulunmaktadır. Stringler konusundan bildiğimiz kadarıyla **stringler değiştirilemez** bir veri tipidir. Ancak, **listelerimiz değiştirilebilir** bir veri tipidir. Listeler bir `[]` köşeli parantez içine veriler eklenerek oluşturulabilir. String türündeki veriler tırnak işareti içinde yazılır.

```
#Liste değişkeninde değişik veri tiplerinden değerleri sakla-yab  
iliyoruz.
```

```
liste = [3,4,5,6,"Elma",3.14,5.324]
```

```
# len fonksiyonu listeler üzerinde de kullanılabilir.
```

```
liste= [3,4,5,6,7,8,9,10]
```

```
len(liste)
```

```
8
```

```
# Boş liste oluşturma
```

```
bos_liste = []
```

```
# 0. eleman
```

```
liste[0]
```

```
3
```

```
# 4. eleman
```

```
liste[4]
```

```
7
```

```
# Sonuncu Eleman
```

```
liste[len(liste)-1]
```

```
10
```

```
# Baştan 4. indekse kadar (dahil değil)
```

```
liste[:4]
```

```
[3,4,5,6]
```

```
# 1.indeksten 5.indekse kadar
```

```
liste[1:5]
```

```
[4,5,6,7]
```

```
liste[5:]
```

```
[8,9,10]
```

```
liste[::2]
```

```
[3, 5, 7, 9]
```

```
liste[::3]
```

```
[3, 6, 9]
```

```
liste[::4]
```

```
[3, 7]
```

```
liste[::-1]
```

```
[10, 9, 8, 7, 6, 5, 4, 3]
```

```
# Bir listeyi başka bir listeye toplamak için.
```

```
liste1 = [1,2,3,4,5]
```

```
liste2 = [6,7,8,9,10]
```

```
liste1 + liste2
```

```
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
```

```
# Bir listeye bir tane eleman eklemek için
```

```
liste = [1,2,3,4]
```

```
liste = liste + ["Murat"]
```

```
[1, 2, 3, 4, 'Murat']
```

```
# Listeler direk olarak değiştirilebilirler.
```

```
liste[0] = 10
```

```
[10, 2, 3, 4, 'Murat']
```

```
# Şöyle bir kullanım da mümkündür
```

```
liste[:2] = [40,50]
```

```
[40, 50, 3, 4, 'Murat']
```

```
# Bir listeyi bir tamsayıyla da çarpabiliriz.
```

```
liste = [1,2,3,4,5]
```

```
liste * 3
```

```
[1, 2, 3, 4, 5, 1, 2, 3, 4, 5, 1, 2, 3, 4, 5]
```

```
print(liste)
```

```
[1, 2, 3, 4, 5]
```

```
(Liste işleme sokulduğunda değişmez!)
```

```
liste = liste * 3
```

```
liste
```

```
[1, 2, 3, 4, 5, 1, 2, 3, 4, 5, 1, 2, 3, 4, 5]
```

```
(Eşitleme yaptığımız için liste değişti.)
```



Python Uygulama Kodları 17

ÖRNEK 1.

```
a=[1,2,3,4,5,6,7,8,9]
print(a[::2])
```

ÖRNEK 2.

```
a=[1,2,3,4,5]
print(a[3:0:-1])
```

ÖRNEK 3.

```
liste=[2,4,6,8]
toplam = 0
for sayi in liste:
    toplam=toplam+sayi
print("Listenin toplamı : ",toplam)
```

ÖRNEK 4.

```
s=[i for i in range(1,50,10)]
print(s)
```

ÖRNEK 5.

```
import random
numaralar=[i for i in range(50)]
secim =random.sample(numaralar,6)
print(secim)
```

ÖRNEK 6.

```
import random
numaralar=[ i for i in range(1,50,7)]
secim = random.choice(numaralar)
print(secim)
```

ÖRNEK 7.

```
listem=[6, 7, 13, 12]
a=listem[3]% listem[1]
print(a)
```

ÖRNEK 8.

```
cumle = ['hayat','python','ile','guzel']
a=cumle[1:]
print(a)
```

ÖRNEK 9.

```
cumle = ['hayat','python','ile','guzel']
a=cumle[:2]
b=cumle[2:]
print(a+b)
```

ÖRNEK 10.

```
cumle = ['hayat','python','ile','güzel']
cumle[3]='çok daha güzel'
print(cumle)
```

ÖRNEK 11.

```
cumle = ['hayat','python','ile','güzel']
print(cumle*2)
print("Cümlemiz :",cumle)
```

ÖRNEK 12.

```
cumle = ['hayat','python','ile','güzel']
cumle=cumle*2
print(cumle)
```



LİSTELER(KOMUTLAR)

append() metodu

append metodu listenin en sonuna eleman eklememizi sağlar.

```
liste = [1,2,3,4,5,6,7]
liste.append(34)
liste
[1, 2, 3, 4, 5, 6, 7, 34]
liste.append("Python")
liste
[1, 2, 3, 4, 5, 6, 7, 34, 'Python']
```

extend() metodu

extend() metodu bir listeye başka bir listenin elemanları eklememizi sağlar.

```
liste = [1,2,3,4,5,6,7]
liste.extend([10,11,12])
liste
[1, 2, 3, 4, 5, 6, 7, 10, 11, 12]
```

insert() metodu

insert() metodu listenin belli bir indeksine bir eleman eklememizi sağlar.

```
liste = [1,2,3,4,5,6,7,8,9]
liste.insert(2,"Python") # 2.indekse "Python" değerini ekliyoruz
liste
[1, 2, 'Python', 3, 4, 5, 6, 7, 8, 9]
```

pop() metodu

pop() metodu içine hiçbir değer vermezsek listenin son elemanını silerek ekrana basar. İçine belli bir indeks değeri verirse o indeksi siler ve ekrana basar.

```
liste = [1,2,3,4,5,6,7]
liste.pop() # Son eleman siliniyor.
7
liste
[1, 2, 3, 4, 5, 6]
liste.pop(2) # 2.indeksteki eleman siliniyor.
3
```

remove() metodu

remove() metodu verdiğimiz değeri listeden çıkarmamızı sağlar.

```
liste = ["Python","Php","Java","C"]
liste.remove("Python") # Python'ı siliyoruz.
liste
['Php', 'Java', 'C']
liste.remove("Javascript") # Listede yok hata verir.
```

index() metodu

index() metodu verilen bir değerın baştan başlayarak hangi indekste olduğunu söyler. Değer listede yoksa hata döner. Eğer ekstra index değeri belirtilirse, *index metodu()* değeri bu indeksten itibaren aramaya çalışır.

```
liste = [1,2,3,4,3,3,5,6,7,8,9]
liste.index(3) # 3 elemanı baştan başlayarak 2.indekste
2
```




Python Uygulama Kodları 18

sort() metodu

sort() metodu bir listenin elemanlarını sayıysa küçükten büyüğe , string ise alfabetik olarak sıralar. Eğer özellikle içine *reverse = True* değeri verilirse elemanları büyükten küçüğe sıralar.

```
liste = [12,-2,3,1,34,100]
liste.sort()
liste
[-2, 1, 3, 12, 34, 100]
liste2 = ["Python","Php","C","Java"]
liste2.sort()
liste2
['C', 'Java', 'Php', 'Python']
liste = [12,-2,3,1,34,100]
liste.sort(reverse = True)
liste
[100, 34, 12, 3, 1, -2]
liste2 = ["Python","Php","C","Java"]
liste2.sort(reverse = True)
liste2
['Python', 'Php', 'Java', 'C']
```

clear() metodu

clear metodu listenin içerisindeki tüm öğeleri siler.

```
liste = [12,-2,3,1,34,100]
liste.clear()
liste
[]
```

del metodu

del metodu listenin belirtilen öğesini sildiği gibi, listenin tamamını silmek için de kullanılır.

```
liste = [12,-2,3,1,34,100]
del liste[2]
liste
[12,-2,1,34,100]
del(liste)
liste
```

```
Traceback (most recent call last):
  File "<pyshell#2>", line 1, in <module>
    liste
NameError: name 'liste' is not defined
```

count() metodu

count() metodu verilen bir değerın listede kaç defa geçtiğini sayar.

```
liste = [1,2,3,4,5,6,1,1,1,1,1,1,1,1,8]
liste.count(1)
9
liste.count(10)
0
```



Python Uygulama Kodları 18

ÖRNEK 1.

```
>>> il=["Van","Muş","Ordu","Bolu","Yozgat"]
>>> il.sort()
>>> il
```

ÖRNEK 2.

```
>>> il=["Van","Muş","Ordu","Bolu","Yozgat"]
>>> il.reverse()
>>> il
```

ÖRNEK 3.

```
meyve=["apple", "banana", "cherry"]
meyve.append("orange")
print(meyve)
```

ÖRNEK 4.

```
meyve=["apple", "banana", "cherry"]
meyve.insert(1,"melon")
print(meyve)
```

ÖRNEK 5.

```
meyve=["apple","melon","banana","cherry"]
meyve.pop()
```

ÖRNEK 6.

```
meyve.clear()
meyve
```

ÖRNEK 7.

```
>>> takimler = ['Galatasaray','Fenerbahçe','Beşiktaş']
>>> sehir=["İstanbul"]
>>> futbol=takimler+sehir
>>> futbol
```

ÖRNEK 8.

```
>>> takimler = ['Galatasaray','Fenerbahçe','Beşiktaş','Trabzonspor']
>>> takimler.remove("Beşiktaş")
>>> takimler
```

ÖRNEK 9.

```
>>> fruits=['orange','apple','pear', 'banana', 'kiwi','apple', 'banana']
>>> fruits.count('apple')
```

ÖRNEK 10.

```
>>> liste=[i for i in range(100) if i**2<100]
>>> liste
```

ÖRNEK 11.

```
>>> a = [-1, 1,66.25, 333, 333, 1234.5]
>>> del a[0]
>>> a
```

ÖRNEK 12.

```
names = ['Ali', 'Ayşe', 'Zeynep', 'Ahmet']
print(names[-1][-1])
```

ÖRNEK 13.

```
ilk=[1,2,3]
son=[6,5,4]
ilk.extend(son)
rakamlar=ilk+son
print(rakamlar)
```

ÖRNEK 14.

```
arac=["BMW","Mercedes","Ford","Honda"]
del(arac)
print(arac)
```

ÖRNEK 15.

```
import random
numaralar=[ i for i in range(1,50,7)]
secim = random.choice(numaralar)
print(secim)
```

STRING METODLARI

```
mesaj="Hello There. My name is Mustafa"
```

```
*ymesaj=mesaj.upper()---upper harfleri büyük yapar.
```

```
*ymesaj=mesaj.lower()---Lower harfleri küçük yapar.
```

```
*ymesaj=mesaj.title()---her kelimenin ilk harfi büyük olur.
```

```
*ymesaj=mesaj.capitalize()---Sadece ilk kelimenin ilk harfi büyük olur.
```

```
*ymesaj=mesaj.split() ---mesaj ifadesindeki her kelimeyi diziye dönüştürdü.
```

```
*ymesaj=mesaj.split(".") ---"nokta " işaretinden sonra ayırmaya başlar.
```

```
Print(ymesaj)
```

```
Çıktı>>>['Hello There', ' My name is Mustafa']---bir nevi ikiye böldük...Burada cümle içinde nokta yerine virgül  
olsaydı bu defa split(",") şeklinde kodlamamız gerekecekti.
```

```
*print(ymesaj[0])---bu durumda Hello There yazısı çıkacaktır.
```

```
*print(ymesaj[1])---bu durumda ekrana My name is Mustafa yazısı çıkacaktır.
```

```
*index=mesaj.find('Mustafa')--- mesaj cümlesinde Mustafa kelimesinin index numarasını verir.  
print(index) ---ekranda 24 yazar.
```

```
*acaba= mesaj.startswith("H")---mesaj ifadesi H harfiyle mi başlamış diye sorar. endwith de cümle sonundaki  
karakteri kontrol eder.  
print(acaba) --- ekrana True yazar.
```

```
*ymesaj=mesaj.replace('Mustafa','Çınar')---Mustafa yerine Çınar yazar.
```

```
*ymesaj=mesaj.replace(' ','*')---boşluk yerine * işareti gelir.
```

- a=("kitap")
sonuc=sorted(a)
print(sonuc)
['a', 'i', 'k', 'p', 't']--- sorted metodu string ifadeleri alfabetik sıraya dizer.

len() fonksiyonu: Stringlerin uzunluğunu ölçer.

len("Türkiye") **Çıktı: 7**

len("Bilgisayar Bilimi") **Çıktı: 17**

len("Haydi, Kodla!") **Çıktı: 13** Noktalama işaretleri de sayılır.

type() fonksiyonu: Verilerin türünü sorgular.

string indeksleme ve parçalama: stringlerde her bir karakterin kendine has bir konumu vardır. Bu konumlara indeks adı verilir. Python'da ve çoğu programlama dilinde indeksleme "0" dan başlar. Aşağıdaki örnekleri inceleyelim.

```
>>> a="Kodlama"
>>> a[0]
'K'
>>> a[2]
'd'
>>> a[-1] #sondan 1.karakter
'a'
>>> a[-3]
'a'
>>> a[2:6]
'dlam'
>>> a[:6] #baştan 6.karaktere kadar tüm karakterler
'Kodlam'
>>> a[2:] #2.karakterden sona kadar tüm karakterler
'dlama'
>>> a[:] #tüm karakterler
'Kodlama'
>>> a[2:-2]
'dla'
>>> a[:-2]
'Kodla'
>>> a="Kodlama Eğitimi"
```

```
>>> a[::2] #Baştan sona kadar 2 sıra atlayarak
'KdaaEiii'
>>> a[::-1] #Sondan başa kadar 1 sıra atlayarak
'imitiğE amaldoK'
```

ÖRNEK UYGULAMALAR GENEL

```
#klavyeden girilen sayının çarpanlarını veren program...|
a=int(input("bir sayı giriniz:"))
if a!=0:
    for i in range(1,a+1):
        if (a%i==0):
            print(i)
```

```
#en az 18 yaş ve lise mezunlarına ehliyet veren program
isim=input("isminizi giriniz:")
yas=int(input("yaşınızı giriniz: "))
egitim=input("egitim durumunuz nedir ?: ")

if (yas>=18) and (egitim=="üniversite" or egitim=="lise"):
    print("ehliyet alabilirsiniz..")
else :
    print("ehliyet alamazsınız..")
```

```
#klavyeden girilen iki sayı arasındaki sayıların toplamını bulan program..
```

```
a=int(input("birinci sayıyı gir:"))
b=int(input("ikinci sayıyı gir:"))
toplam=0
for i in range(a,b):
    toplam=toplam+i
print(toplam)
```

```
#emekli olabilmek için erkekler en az 7800 gün prim yatırmalı ve 58 yaşını doldurmuş olmalıdır.  
#Kadınlar ise 7200 ve 55 yaş.buna göre kullanıcıya cinsiyet ve yaş bilgilerini sorup  
#emekli olabilecekse yaş*1000 şeklinde ikramiyeyi hesaplayacak python programını oluşturun...
```

```
cinsiyet=str(input("cinsiyetiniz nedir? : \n'E' veya 'K' \n"))  
yas=int(input("yaşınızı giriniz:\n"))  
prim=int(input("prim gün sayınızı giriniz:\n"))  
sonuc=0  
if cinsiyet=="E" and prim>=7800 and yas>=58 :  
    sonuc=yas*1000  
    print("Emekli olabilirsiniz...\n ikramiyeniz:",sonuc)  
elif cinsiyet=="K" and prim>=7200 and yas>=55 :  
    print("emekli olabilirsiniz..\n ikramiyeniz:",sonuc)  
  
else :  
    print("emekli olamazsınız...")
```

```
#klavyeden girilen sayının faktöriyelini hesaplayan program  
sayac=1  
faktöriyel=1  
sayi=int(input("bir sayı giriniz :"))  
while sayac<=sayi :  
    faktöriyel=faktöriyel*sayac  
    sayac+=1  
print(sayi,"sayısının faktöriyeli :", faktöriyel)
```



```
#klavyeden girilen bir sayının 1 den başlayarak arttığını gösteren program. #klavyeden girilen bir sayının 0'a kadar azalışını sağlayan program..
```

```
n=1
```

```
karar=int(input("kaç kadar arttıralım :"))
```

```
while n<= karar :
```

```
    print(n)
```

```
    n+=1
```

```
a=int(input("bir sayı giriniz :"))
```

```
for i in range(a,-1,-1):
```

```
    print(i)
```

```
#kelimedeki sesli harf sayısını bulan program|...
```

```
s = input("bir kelime giriniz: \n")
```

```
sesli_harf = 'aeioöü'
```

```
sayac = 0
```

```
for i in s:
```

```
    if i in sesli_harf:
```

```
        sayac += 1
```

```
print(s," metnindeki sesli harf sayısı:",sayac)
```

```
#farklı tutar bilgilerine göre indirim yapan program...
```

```
tutar=int(input("Kaç TL'lik alışveriş yaptınız ? :"))
```

```
result=0
```

```
if (tutar>0 and tutar<=200) :
```

```
    result=(tutar)-(tutar*10/100)
```

```
    print("toplam ödeyeceğini tutar :",result,"TL")
```

```
elif (tutar>200)and (tutar<=400) :
```

```
    result=(tutar)-(tutar*15/100)
```

```
    print("toplam ödeyeceğiniz tutar :",result,"TL")
```

```
elif (tutar>400):
```

```
    result=(tutar)-(tutar*20/100)
```

```
    print("toplam ödeyeceğiniz tutar :",result,"TL")
```



```
#kullanıcıdan 2 vize (%60) ve final (%40) notunu alıp ortalama hesaplayınız.  
#   eğer ortalama 50 ve izeri ise geçti değilse kalsın  
#   a) ortalama 50 olsa bile final notu en az 50 olmalıdır.
```

```
vize1=float(input("1.vize notunu giriniz:"))  
vize2=float(input("2.vize notunu giriniz:"))  
final=float(input("final notunu giriniz:"))  
ort= (vize1*0.3)+(vize2*0.3)+(final*0.4)  
# a şıkkı:  
result=(ort>=50)and(final>=50)  
print(f"öğrencinin ortalaması:{ort} ve geçme durumu:{result}")
```

```
#bir öğrencinin 2 yazılı bir sözlü notunu alıp hesaplanan ortalamaya göre,  
#not aralığına karşılık gelen not bilgisini yazdırınız...
```

```
#0-24=0  
#25-44=1  
#45-54=2  
#55-69=3  
#70-84=4  
#85-100=5
```

```
yazilil=float(input("1.yaziliyi giriniz: "))  
yazili2=float(input("2.yaziliyi giriniz: "))  
sozlu=float(input("sözlü notunuzu giriniz: "))  
ortalama=(yazilil+yazili2+sozlu)/3  
  
if (ortalama>=0) and (ortalama<25) :  
    print(f"ortalamanız: {ortalama} ve notunuz:0")  
elif (ortalama>=25) and (ortalama<45) :  
    print(f"ortalamanız: {ortalama} ve notunuz:1")  
  
elif (ortalama >= 45) and (ortalama <55):  
    print(f"ortalamanız: {ortalama} ve notunuz:2")  
  
elif (ortalama >=55) and (ortalama< 70):
```

#Bir öğrencinin iki yazılı ve bir sözü notunu alıp ortalamasını hesaplayan program

```
yazili1=float(input("1.yazılı notunu gir:"))
yazili2=float(input("2.yazılı notunu gir:"))
sozlu=float(input("sözlü notunu gir:"))
ortalama=(yazili1+yazili2+sozlu)/3
if(ortalama>=0 and ortalama<25):
|   print("notunuz:0 ortalama:",ortalama)
elif(ortalama>=25 and ortalama<45):
|   print("notunuz:1 ortalama:",ortalama)
elif(ortalama>=45 and ortalama<55):
|   print("notunuz:2 ortalama:",ortalama)
elif(ortalama>=55 and ortalama<70):
|   print("notunuz:3 ortalama:",ortalama)
elif(ortalama>=70 and ortalama<85):
|   print("notunuz:4 ortalama:",ortalama)
else:
|   print("notunuz:5 ortalama:",ortalama)
```

```
#yıldız çizen program
import turtle
yildiz=turtle.Turtle()
for i in range(5):
    yildiz.forward(150)
    yildiz.right(144)
turtle.done()
```

#Üçgen çizen program

```
a=int(input("birinci kenarı giriniz:"))
b=int(input("birinci kenarı giriniz:"))
c=int(input("birinci kenarı giriniz:"))
if (a+b) and (a+c) and (b+c)> a and b and c :
    print("üçgen oluşturulabilir")
else :
    print("üçgen oluşturamazsınız..")
```