

Temel Programlama 1

Öğr. Gör. Furkan DURMUŞ

Samsun Üniversitesi
Teknik Bilimler
Meslek Yüksekokulu



Bu Haftanın Ders Kazanımları

Bu haftayı bitirdiğinizde,

{ 1 } Değişken ve Sabit Kavramı

{ 2 } Atama ve Aritmetik Operatörler

{ 3 } Karşılaştırma Operatörleri

{ 4 } Mantıksal Operatörler

{ 5 } Kimlik Operatörleri

öğrenmiş olacaksınız.

Değişken ve Sabit

```
0 response = requests.get(url) # load from the website
1
2 # checking response.status_code (if you get 502, try rerunning the code)
3 if response.status_code != 200:
4     print(response.status_code) - Try rerunning the code\n")
5
6 # to parse the response object
7 soup = BeautifulSoup(response.content, "html.parser")
8
9 # finding Post images in the soup
10 soup.find_all("img", attrs={"alt": "Post image"})
```

{01} Değişken ve Sabit Kavramları

Programda bir veriyi (sayı, metin, liste, vs.) geçici olarak tutmak için kullanılan isimlendirilmiş alanlardır.

Programlama dillerinde ihtiyaç olduğu an ulaşılacak veri tutuculara **değişken** adı verilir.



{01} Değişken Tanımlama



Sabit kavramı ise uygulama çalıştığı sürece değeri değişmeyen veriler olarak ifade edilebilir. Örneğin bir inç 2,54 cm'dir.

Bu gibi değeri değişmemesi gereken veriler **sabit** olarak tanımlanabilir.

Değeri değiştirilmeyen (veya değiştirilmemesi gereken) değişkenlerdir. **Python'da sabitleri özel olarak tanımlayan bir sözdizimi YOKTUR.** Yani **const** gibi bir anahtar kelime yoktur.

{01} Değişken Tanımlama Kuralları



Değişken tanımlamak için her programlama dilinde önceden belirlenmiş bazı kurallar bulunmaktadır.

Python programlama dilinde değişken tanımlarken önce değişken adı yazılır. Değişken adı yazıldıktan sonra **= (eşittir)** işareti konulur ve değişkenin değeri yazılır.

Ancak burada değişken isimlendirme kurallarına dikkat edilmelidir.

{01} Değişken Tanımlama Kuralları



{01} Değişken isimleri case sensitive yani büyük küçük harf duyarlıdır.

Örneğin; değişken isminin `adres` yada `Adres` olması bu değişkenlerin farklı iki değişken olduğunu gösterir.

{01} Değişken Tanımlama Kuralları



{01} Değişken isimleri case sensitive
yani büyük küçük harf duyarlıdır.

```
name = "Ahmet"  
Name = "Mehmet"  
NAME = "Ayşe"
```



{01} Değişken Tanımlama Kuralları



{01} Değişken isimleri case sensitive
yani büyük küçük harf duyarlıdır.

```
print(name)  
print(Name)  
print(NAME)
```



{01} Değişken Tanımlama Kuralları



{02} Değişken isimlerinin anlaşılır ve karşıladığı değerle ilişkili olmalıdır.

Örneğin; kullanıcıdan elektronik posta bilgisi alınacağı zaman bunu e_posta gibi anlaşılır bir değişken ismi ile ifade edebilirsiniz.

{01} Değişken Tanımlama Kuralları



{02} Değişken isimlerinin anlaşılır ve karşıladığı değerle ilişkili olmalıdır.

```
x = "ahmet"  
y = 70  
z = True
```



{01} Değişken Tanımlama Kuralları



{02} Değişken isimlerinin anlaşılır ve karşıladığı değerle ilişkili olmalıdır.

```
ogrenci_ismi = "ahmet"  
vize_notu = 70  
odev_teslimi = True
```



{01} Değişken Tanımlama Kuralları



{03} Değişken isimlendirilirken farklı standartlar kullanılmaktadır. Python'da genel kabul gören standart Snake Case standardıdır.

Snake case, yazılımda değişken, fonksiyon veya dosya isimlerini yazarken kullanılan bir isimlendirme biçimidir.

Bu biçimde:

- Tüm harfler küçük yazılır.
- Kelimeler alt çizgi (_) ile ayrılır.

Derslerimizde Lower Snake Case kullanılmıştır.

{01} Değişken Tanımlama Kuralları



{03} Değişken isimlendirilirken farklı standartlar kullanılmaktadır. Python'da genel kabul gören standart Snake Case standardıdır.

```
ogrenci_ismi = "ahmet"  
vize_notu = 70  
odev_teslimi = True  
ders = "Temel Prog. 1"
```



{01} Değişken Tanımlama Kuralları



{04}

Değişken isimlendirilirken hem harfler hem de sayılar kullanılabilir. Ancak sayılar başa gelmez.

Örneğin **sayi1** doğru bir isimlendirmeyken **1sayi** doğru bir isimlendirme değildir.

{01} Değişken Tanımlama Kuralları



{04}

Değişken isimlendirilirken hem harfler hem de sayılar kullanılabilir. Ancak sayılar başa gelmez.

```
1user = "Mehmet"
user1 = "Mehmet"
user_1 = "Mehmet"
```



{01} Değişken Tanımlama Kuralları



{05} Değişken isimlendirilirken alt tire (_) kullanılabilir. Ancak boşluk ve diğer özel karakterler (?, %, !, ., -, + vb.) kullanılmaz.

Örneğin **ev adresi** ya da **kimlik%no** gibi değişken isimleri kurallara aykırı olduğundan hataya neden olacaktır.

{01} Değişken Tanımlama Kuralları



{05} Değişken isimlendirilirken alt tire () kullanılabilir. Ancak boşluk ve diğer özel karakterler (?, %, !, ., -, + vb.) kullanılmaz.

```
user_name = "Ali" ✓  
price_2023 = 150.75 ✓  
  
user-name = "Ayşe" ✕  
total price = 99.99 ✕
```



{01} Değişken Tanımlama Kuralları



{06}

Değişken isimlendirilirken özel kullanım için ayrılmış olan if, for, True vb. ifadeler kullanılmamalıdır.

```
if  
else  
elif  
for  
while  
return  
def  
True  
False  
class
```

{01} Değişken Tanımlama Kuralları



{06}

Değişken isimlendirilirken özel kullanım için ayrılmış olan if, for, True vb. ifadeler kullanılmamalıdır.

```
False = "Ali" ✕  
if = 150.75 ✕
```



{01} Değişken Tanımlama Kuralları



{07}

Bazı programlama dillerinde Türkçe karakterlerin (ç,ğ,ı,ö,ş,ü) kullanımı kabul edilirken bazılarında kabul edilmez.

Python'da 3.0 sürümü sonrası Türkçe karakterler kullanılması hataya neden olmaz. Ancak farklı programlama dillerinde problem yaşanmaması için değişken tanımlarken Türkçe karakter kullanılmaması önerilmektedir.

Biz dersimizde Türkçe karakterli değişken kullanmayacağız.

{01} Değişken Tanımlama Kuralları



{07} Python'da Türkçe karakterler kullanılması hataya neden olmaz.

```
ad = "Ali"  
yaş = 25  
şehir = "İstanbul"  
öğrenci_mi = True  
  
print(ad, yaş, şehir, öğrenci_mi)
```



{01} Değişken Tanımlama Kuralları



{08}

Python'da sabit değerler, genellikle **upper snake case** (büyük harflerle ve kelimeler arasına alt çizgi konarak) yazılır.

Bu isimlendirme tarzı, sabitlerin program boyunca değişmeyecek değerler olduğunu belirtmek için kullanılır.

Örneğin, `MAX_SPEED`, `DEFAULT_TIMEOUT` veya `API_KEY` gibi isimler sabitleri ifade eder.

{01} Değişken Tanımlama Kuralları



{08} Python'da sabit değerler, genellikle **upper snake case** (büyük harflerle ve kelimeler arasına alt çizgi konarak) yazılır.

```
PI = 3.14159  
MAX_CONNECTIONS = 100  
DEFAULT_TIMEOUT = 30  
API_KEY = "abcdef12345"
```



{01} Değişken Tanımlama Kuralları



{09}

Python'da tek alt çizgi (_) ile başlayan değişkenler, “bu değişken dahili, dışarıdan kullanılmamalı” anlamına gelen **gizlilik** işaretidir.

Ancak bu sadece bir anlaşmadır (konvansiyon).

Python erişimi engellemez, sadece geliştiricilere “özel” olduğunu belirtir. Böylece kod okunabilirliği artar ve yanlış kullanım önlenir.

{01} Değişken Tanımlama Kuralları

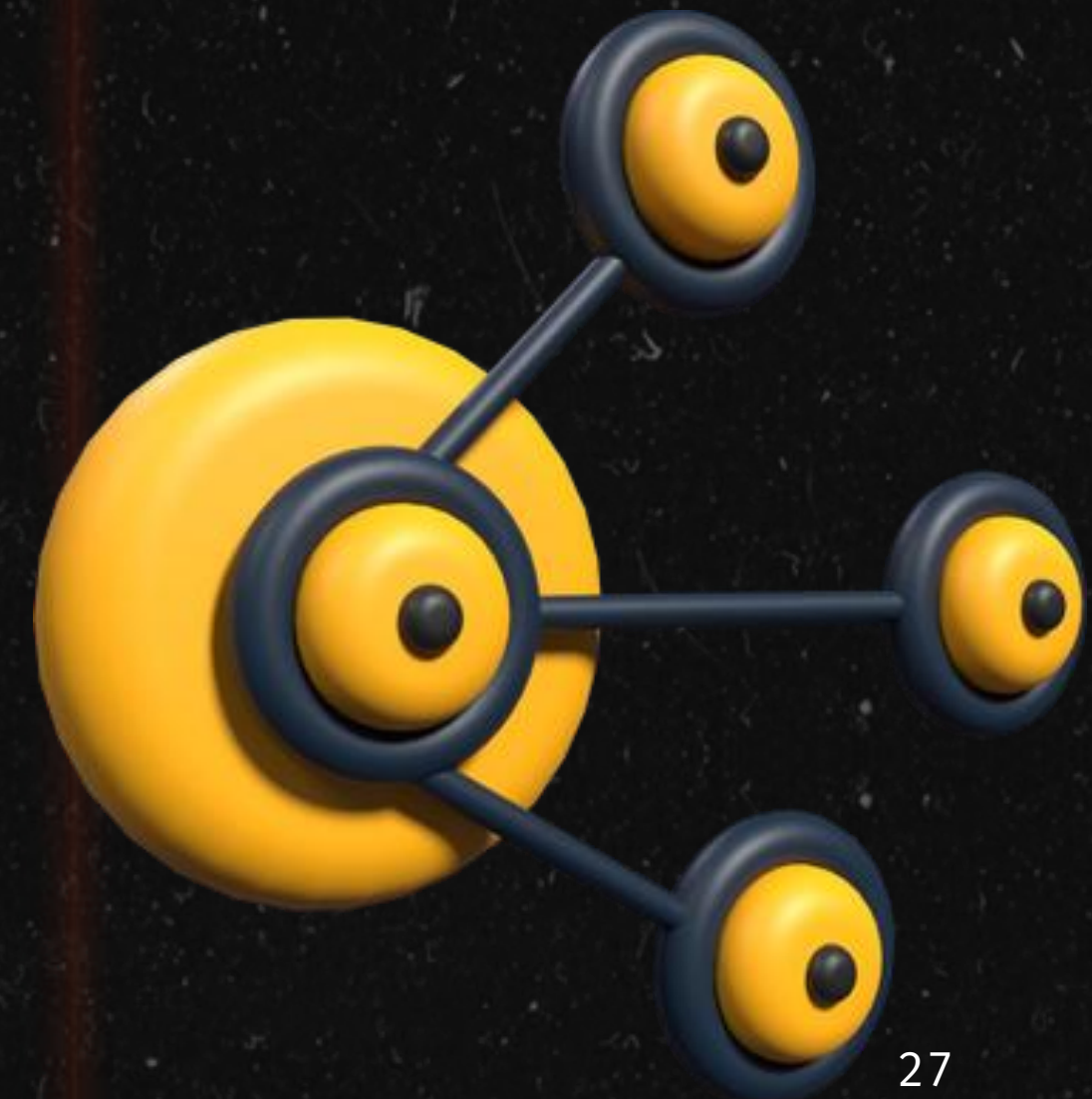


{09} Python'da tek alt çizgi (_) ile başlayan değişkenler, “bu değişken dahili, dışarıdan kullanılmamalı” anlamına gelen **gizlilik** işaretidir.



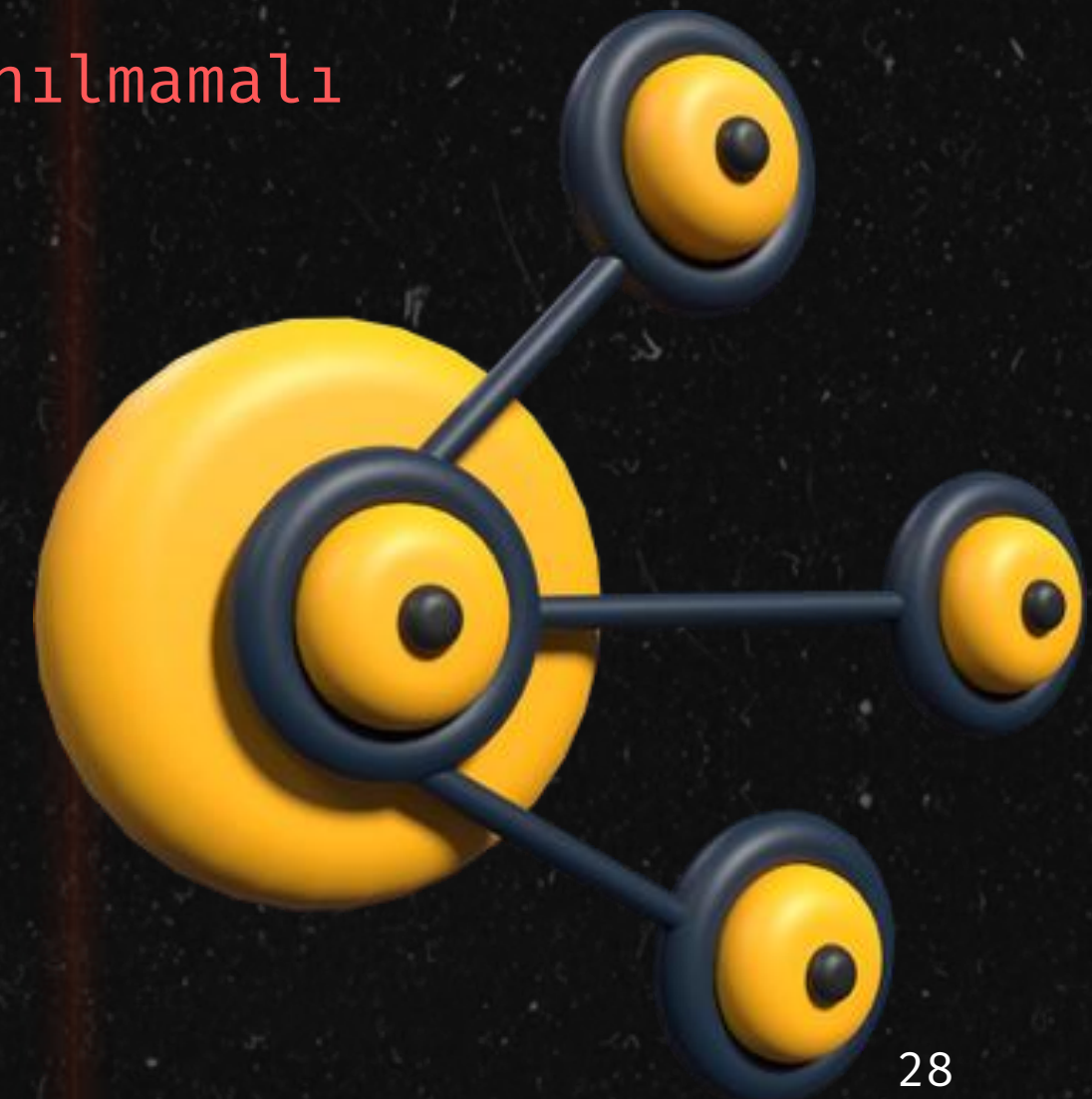
{01} Değişkenler

```
#      Değişken İsmi
1      kullanıcı_adi
2      ürün_fiyatı
3      MAX_SPEED
4      1_sayı
5      e-posta
6      _gecici_veri
7      else
8      kullanıcı adres
9      sayı_2
10     DEFAULT_VALUE
```



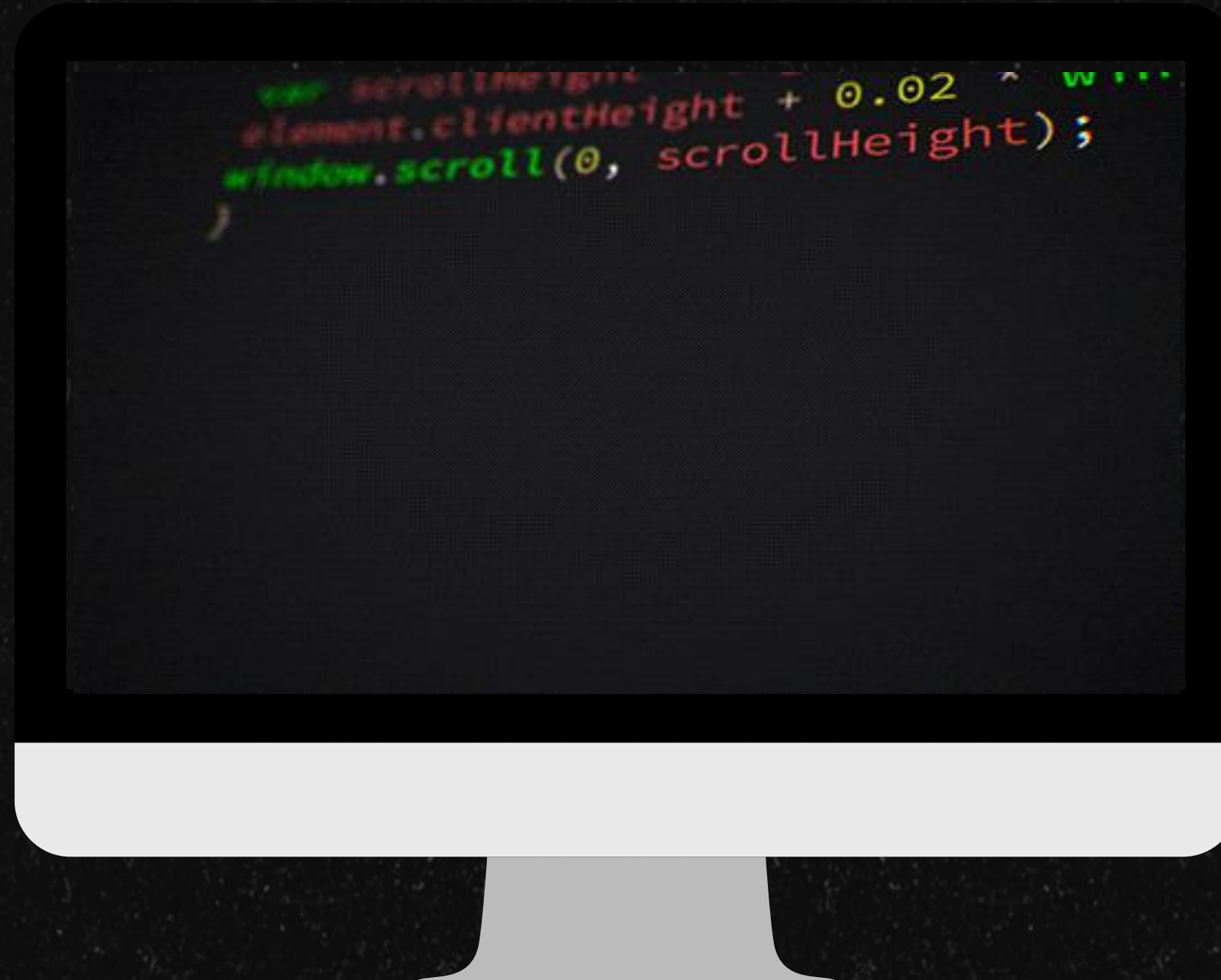
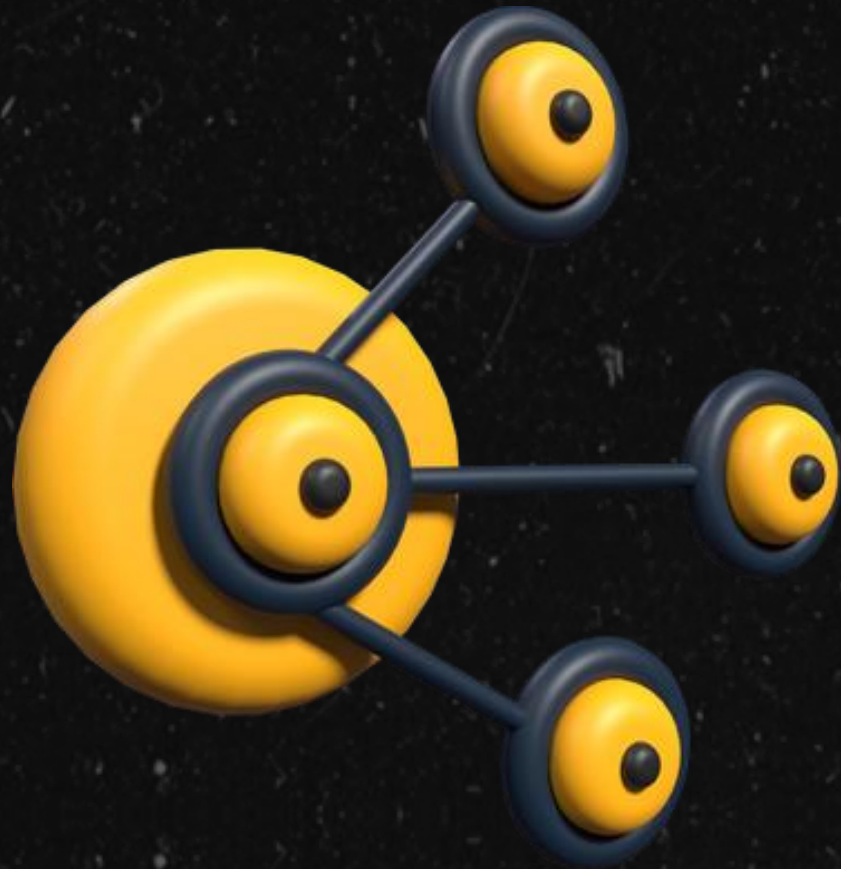
{01} Değişkenler

#	Değişken İsmi	Doğru mu?
1	kullanici_adi	Doğru
2	ürün_fiyatı	Hata vermez ama kullanılmamalı
3	MAX_SPEED	Doğru
4	1_sayi	Yanlış
5	e-posta	Yanlış
6	_gecici_veri	Doğru
7	else	Yanlış
8	kullanici adres	Yanlış
9	sayi_2	Doğru
10	DEFAULT_VALUE	Doğru



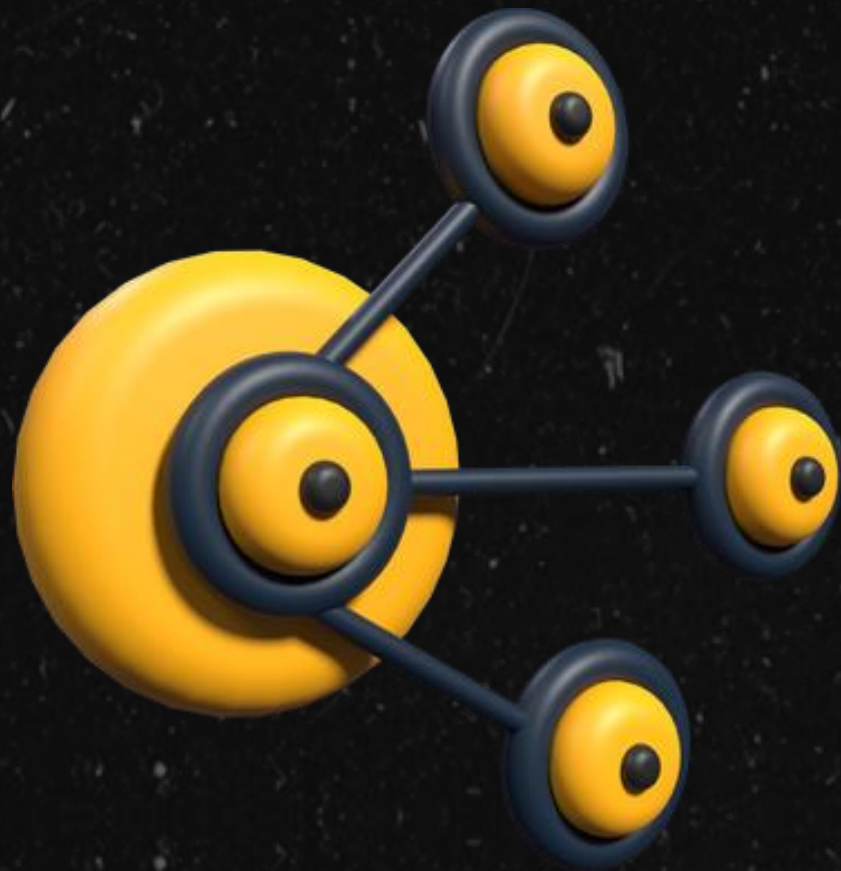
{01} Değişkenler

Kısa kenarı 3 cm, uzun kenarı 5 cm olan dikdörtgenin alanını hesaplayınız.



{01} Değişkenler

Kısa kenarı 3 cm, uzun kenarı 5 cm olan dikdörtgenin alanını hesaplayınız.



```
kisa_kenar=3  
uzun_kenar=5  
alan=kisa_kenar * uzun_kenar  
print(alan)
```

Aritmetik Operatörler

{02} Operatör Nedir ?

Veriler üzerinde işlem yaparak yeni değerler üretilmesini sağlayan programlama dili sembollerine operatör adı verilir.

Python programlama diline yeni başlayanlar için aritmetiksel, atama, karşılaştırma, mantıksal ve kimlik operatörleri öğrenmek son derece önemlidir.



{02} Aritmetiksel Operatörler

Operatör	Tanımı	Örnek
+	Toplama	$a + b$
-	Çıkarma	$a - b$
*	Çarpma	$a * b$
/	Bölme	a / b
%	Mod alma (Bir sayının diğer sayıya bölümünden kalan)	$a \% b$
**	** Kuvvet alma (a üzeri b)	$a ** b$
//	Tam sayı bölme (Bölme işleminde sadece tam kısım alınır.)	$a // b$

{02} Aritmetiksel Operatörler

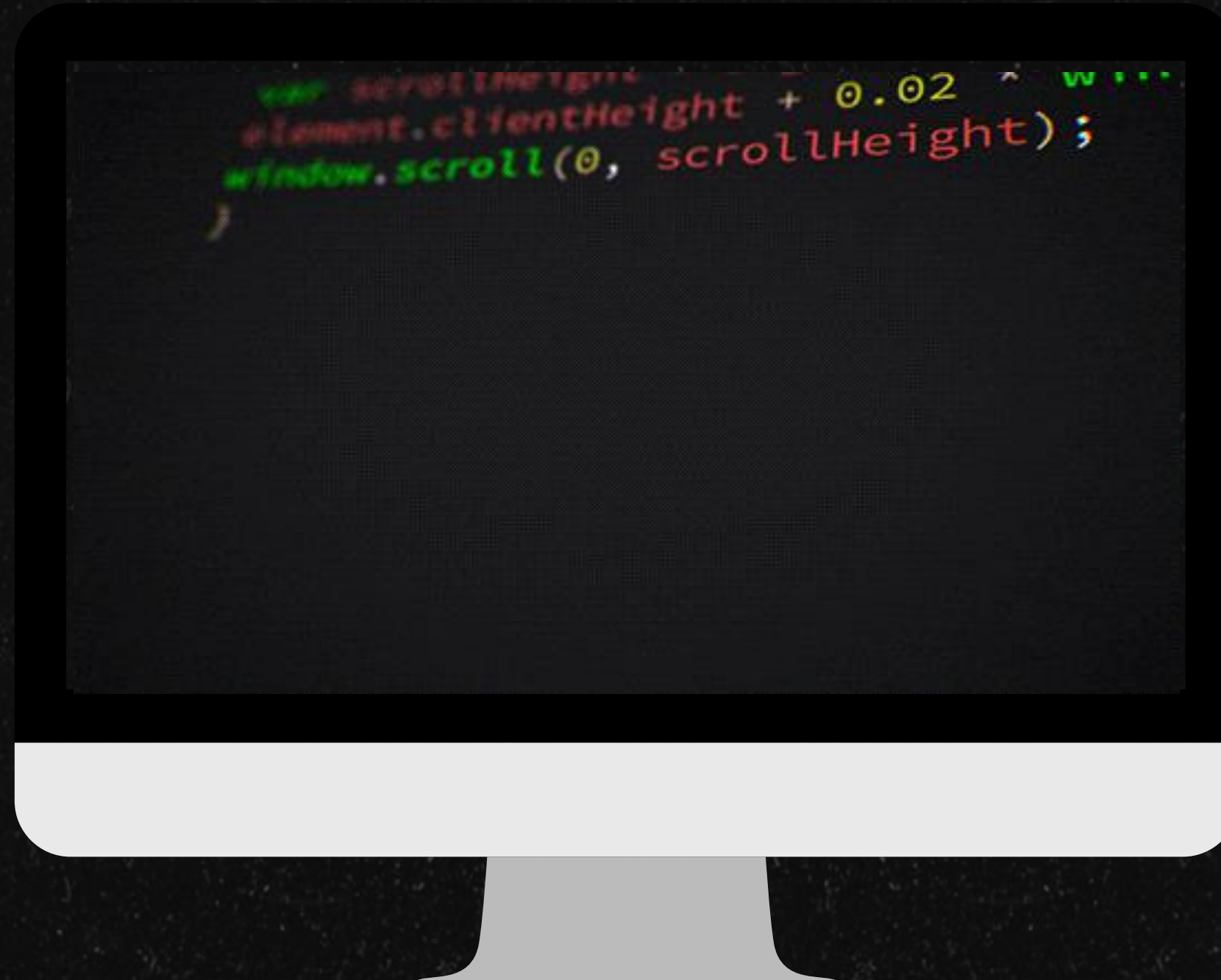
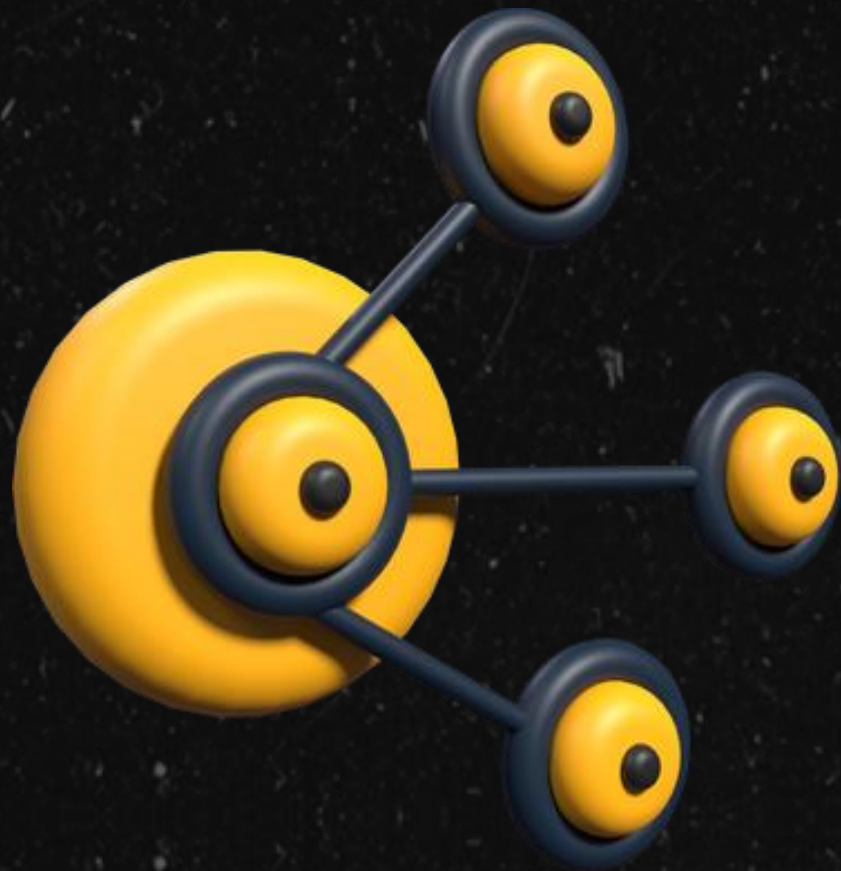
İşlem	Sonuç
$2**5$	
$5//2$	
$4+3$	
$70\%50$	
$5/2$	
$(4-1)**2$	
$(7//3)/2$	

{02} Aritmetiksel Operatörler

İşlem	Sonuç
$2^{**}5$	32
$5//2$	2
$4+3$	7
$70\%50$	20
$5/2$	2.5
$(4-1)^{**}2$	9
$(7//3)/2$	1

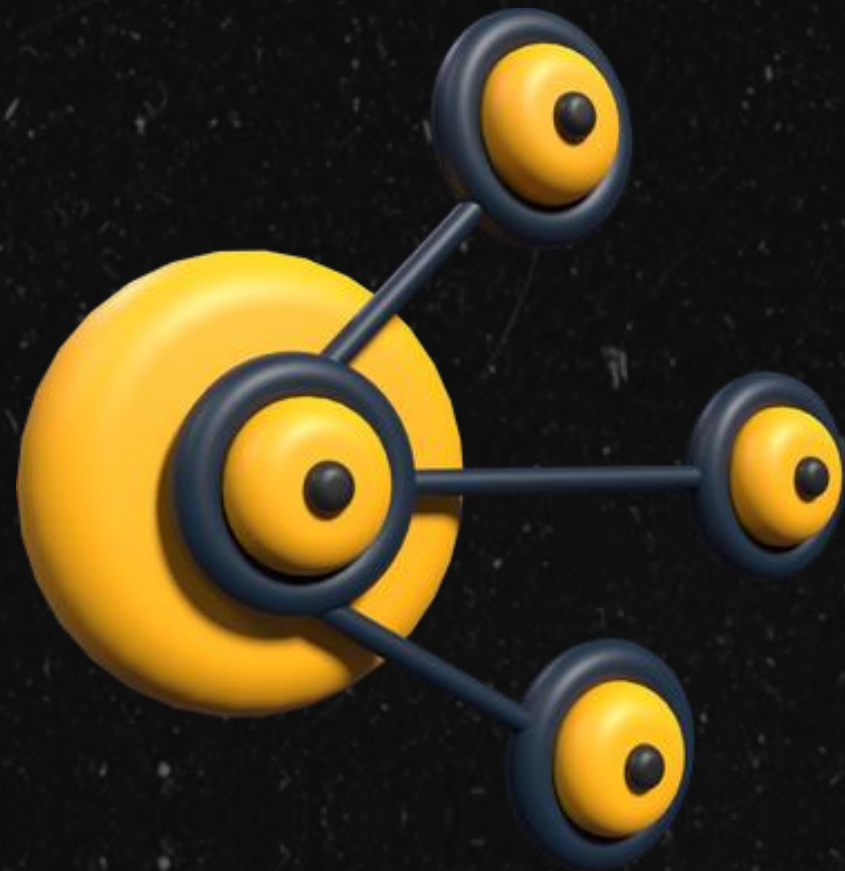
{02} Aritmetiksel Operatörler

Tabanı 8 cm, yüksekliği 6 cm olan üçgenin alanını hesaplayın



{02} Aritmetiksel Operatörler

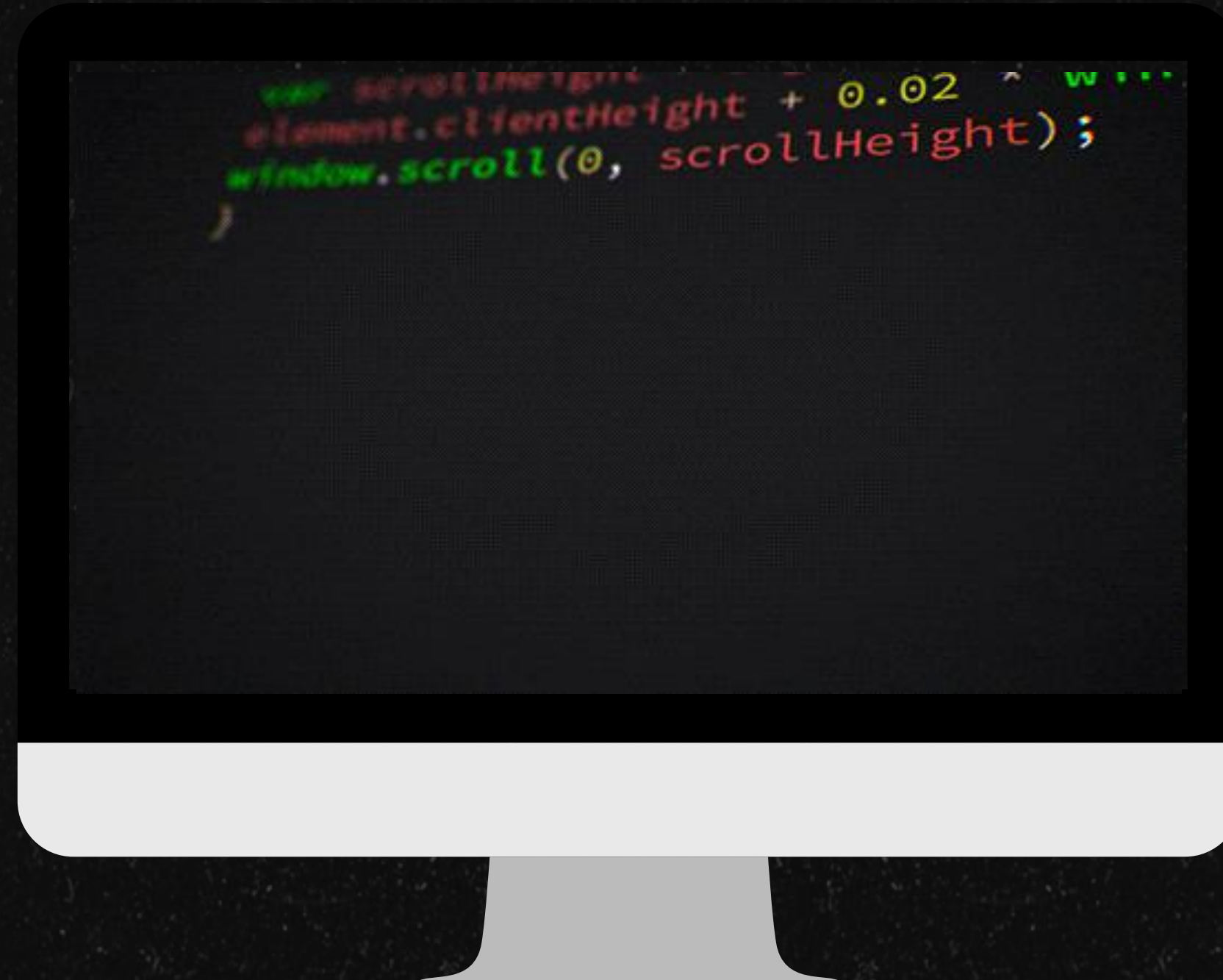
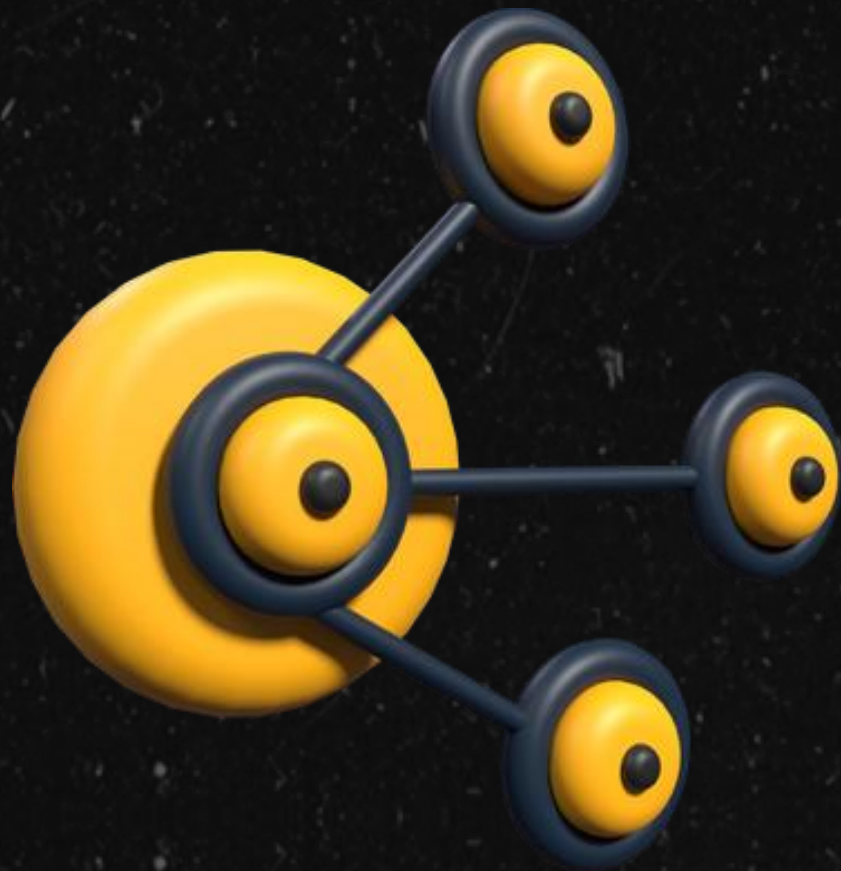
Tabanı 8 cm, yüksekliği 6 cm olan üçgenin alanını hesaplayın



```
taban=8  
yukseklık=6  
alan=(taban * yukseklık) / 2  
print(alan)
```

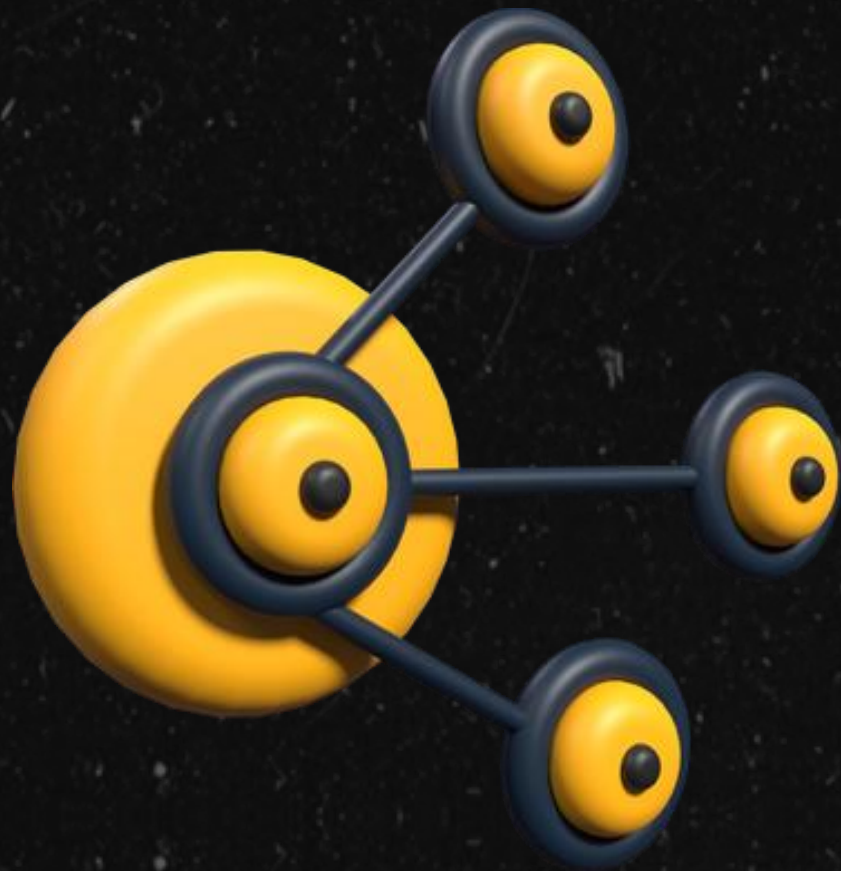
{02} Aritmetiksel Operatörler

2489 sayısı çift ise ekrana 0 tek ise 1 yazdıran kodu sadece aritmetik operatör kullanarak yazınız.



{02} Aritmetiksel Operatörler

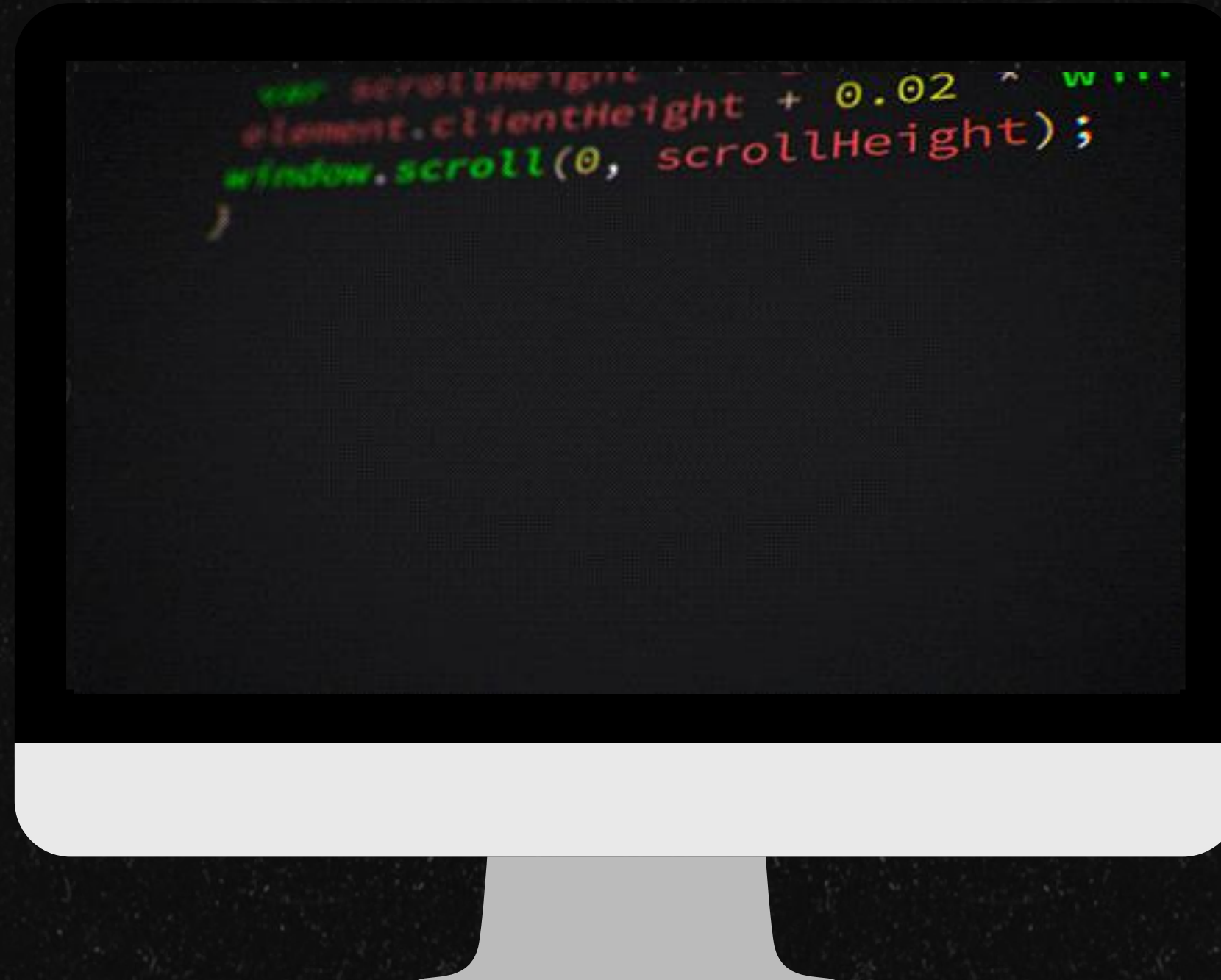
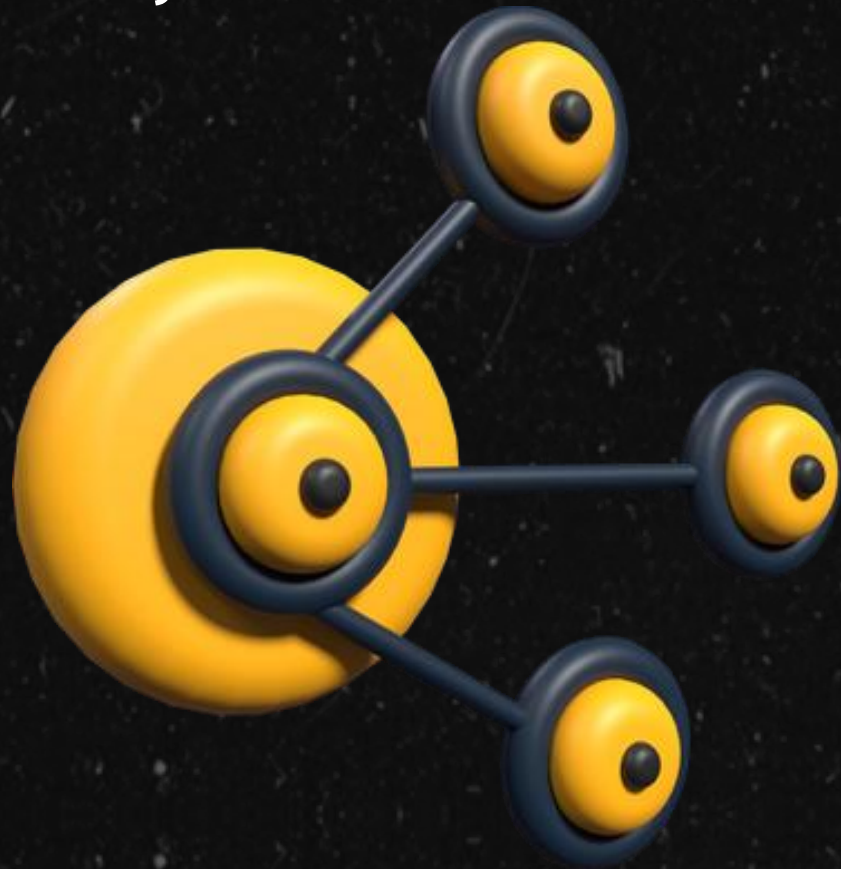
2489 sayısı çift ise ekrana 0 tek ise 1 yazdıran kodu sadece aritmetik operatör kullanarak yazınız.



```
sayi = 2489  
print(sayi % 2)
```

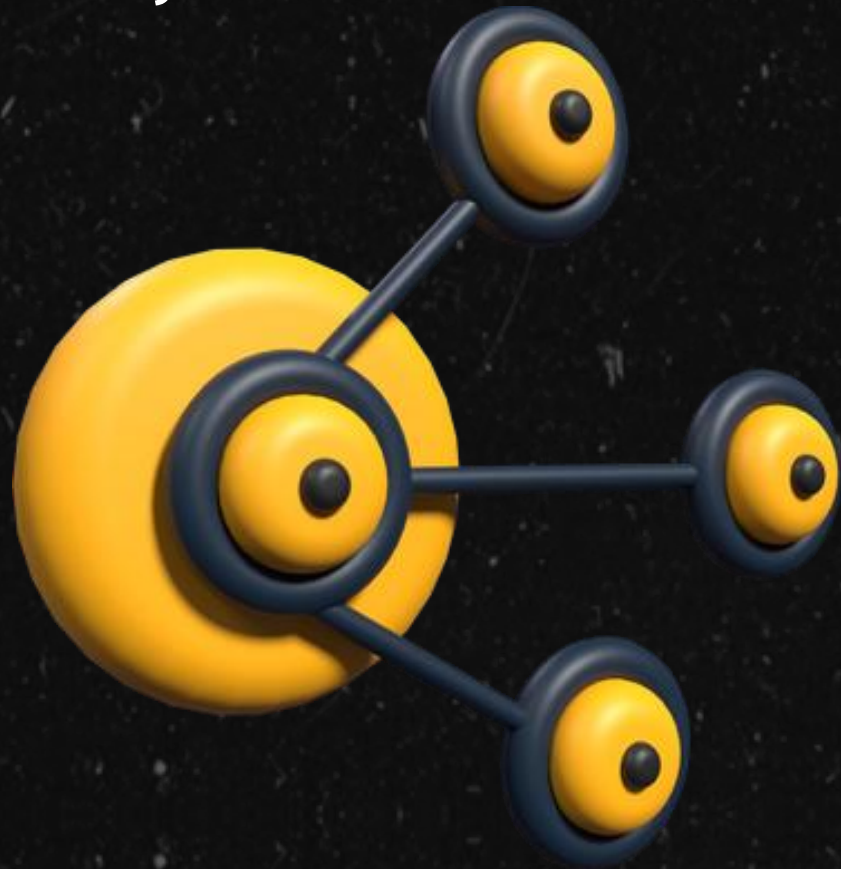
{02} Aritmetiksel Operatörler

458 TL çekilmek istenen bir bankamatikte sadece 5 TL'lik banknot kalmıştır. Kaç adet 5 TL lik banknot kullanılacağını hesaplayan programı yazınız.



{02} Aritmetiksel Operatörler

458 TL çekilmek istenen bir bankamatikte sadece 5 TL'lik banknot kalmıştır. Kaç adet 5 TL lik banknot kullanılacağını hesaplayan programı yazınız.



```
tutar = 458  
banknot = 5  
  
adet = tutar // banknot  
print(adet)
```

{02} Atama Operatörleri

Operatör	Örnek	Açıklama
=	a=2	a değişkenine 2 değeri atanmıştır.
+=	a+=2	a değişkenine 2 değerini ekleyerek yine a değişkenine atanmıştır. Başka bir ifadeyle a=a+2 anlamına gelmektedir.
-=	a-=2	a değişkeninden 2 değeri çıkarılarak yine a değişkenine atanmıştır. Başka bir ifadeyle a=a-2 anlamına gelmektedir.
=	a=2	a değişkeni 2 ile çarpılarak yine a değişkenine atanmıştır. Başka bir ifadeyle a=a*2 anlamına gelmektedir.
/=	a/=2	a değişkeni 2 değerine bölünerek yine a değişkenine atanmıştır. Başka bir ifadeyle a=a/2 anlamına gelmektedir.
%=	a%=2	a değişkenin 2 değeri ile modu alınarak yine a değişkenine atanmıştır. Başka bir ifadeyle a=a%2 anlamına gelmektedir.
=	a=2	a değişkeninin ikinci kuvveti alınarak yine a değişkenine atanmıştır. Başka bir ifadeyle a=a**2 anlamına gelmektedir.
//=	a//=2	a değişkeni 2 değerine tam bölünmüş (kalan dikkate alınmadan) ve çıkan değer yine a değişkenine atanmıştır. Başka bir ifadeyle a=a//2 anlamına gelmektedir.

{02} Atama Operatörleri

İşlem
a=3 a+=2 print(a)
a=6 a*=3 print(a)
a=5 a**=3 print(a)
a=11 a%=3 a**=3 print(a)

{02} Atama Operatörleri

İşlem	Sonuç
a=3 a+=2 print(a)	5
a=6 a*=3 print(a)	18
a=5 a**=3 print(a)	125
a=11 a%=3 a**=3 print(a)	8

algoritma_ornegi.py

17 öğrenci ve 4 tam bilet alınacak.

Öğrenci bileti 60 TL, tam bilet 100 TL.

Toplam bilet fiyatının üzerine %10 hizmet bedeli alınacak.

Her 160 tl lik bilet alımına 1 çekiliş bileti verilecek.

Fatura üzerinde yazacak olan Toplam Bilet Tutarı'nı, Hizmet Bedeli'ni, Ödenecek Toplam Tutar'ı ve Kazanılan Çekiliş Bileti Sayısı'nı veren algoritmayı yazınız. (çekiliş bilet sayısını kdv hariç fiyattan hesaplayınız)

KOD BAŞLANGICI:

```
ogrenci_bilet_fiyati = 60
```

```
tam_bilet_fiyati = 100
```

```
ogrenci_adi = 17
```

```
tam_adi = 4
```

```
.
```

```
.
```

```
.
```

```
print("Toplam Bilet Tutarı:", toplam_bilet_tutari, "TL")
```

```
print("Hizmet Bedeli:", hizmet_bedeli, "TL")
```

```
print("Ödenecek Toplam Tutar:", toplam_tutar, "TL")
```

```
print("Kazanılan Çekiliş Bileti Sayısı:", cekilis_bileti_sayisi, "adet")
```

`algoritma_ornegi.py`

17 öğrenci ve 4 tam bilet alınacak.

Öğrenci bileti 60 TL, tam bilet 100 TL.

Toplam bilet fiyatının üzerine %10 hizmet bedeli alınacak.

Her 160 tl lik bilet alımına 1 çekiliş bileti verilecek.

Fatura üzerinde yazacak olan Toplam Bilet Tutarı'nı, Hizmet Bedeli'ni, Ödenecek Toplam Tutar'ı ve Kazanılan Çekiliş Bileti Sayısı'nı veren algoritmayı yazınız. (çekiliş bilet sayısını kdv hariç fiyattan hesaplayınız)

ÇIKTI :

Toplam Bilet Tutarı: 1420 TL

Hizmet Bedeli : 142.0 TL

Ödenecek Toplam Tutar: 1562.0 TL

Kazanılan Çekiliş Bileti Sayısı: 8 adet

Son

Karşılaştırma Operatörleri

{03} Karşılaştırma Operatörleri

Operatör	Tanımı	Örnek
==	Eşittir	a==b
!=	Eşit değildir	a!=b
<	Küçüktür	a	Büyüktür	a>b
<=	Küçük eşittir	a<=b
>=	Büyük eşittir	a>=b

{03} Karşılaştırma Operatörleri

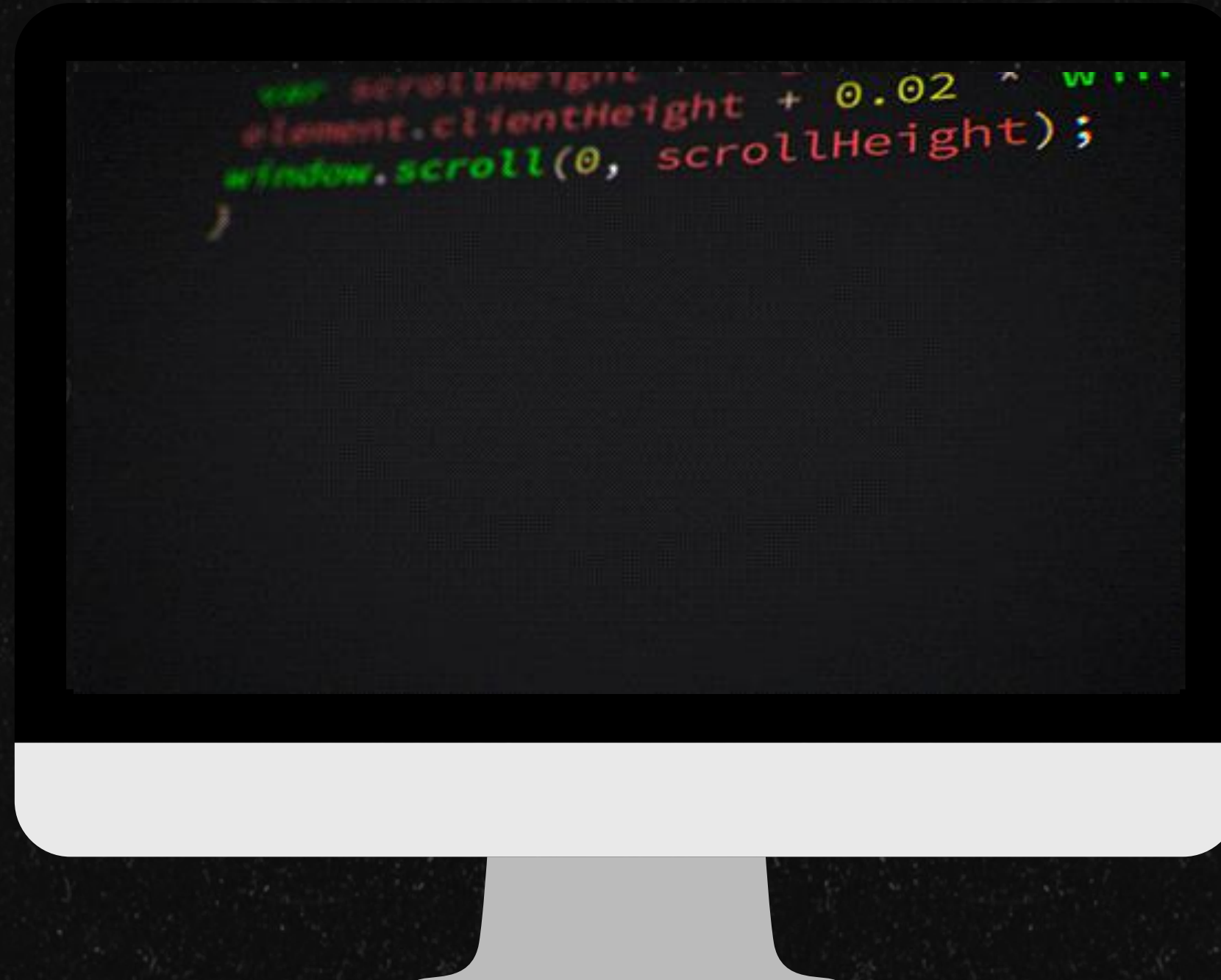
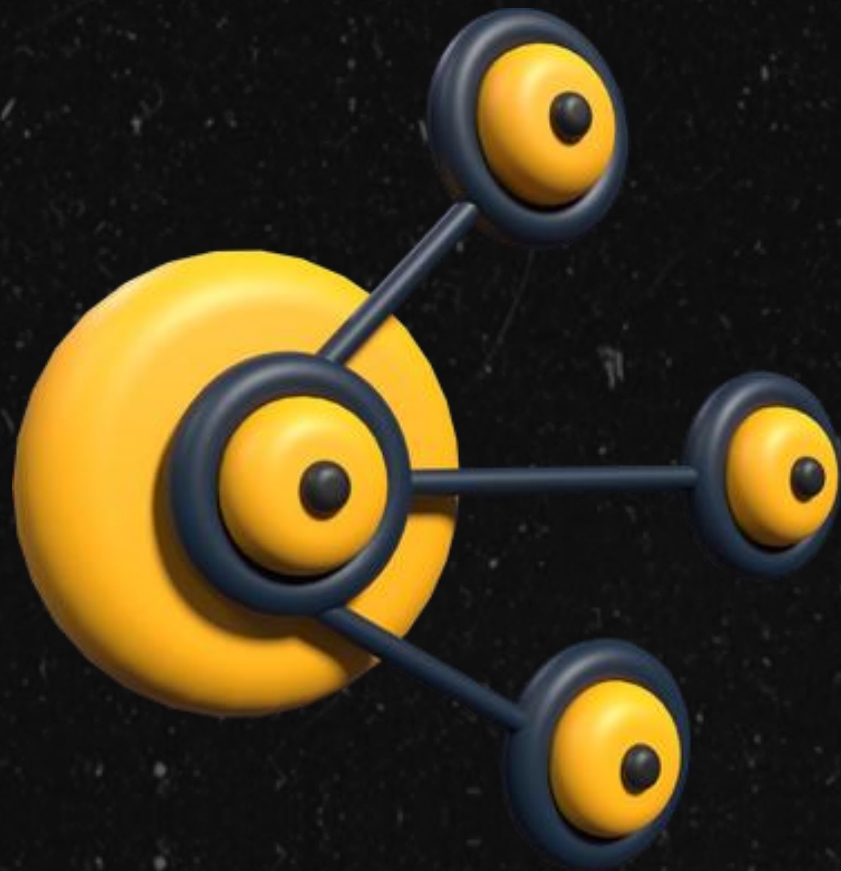
İşlem
a=3 b=4 print (a==b)
a=6 b=6 print (a==b)
a=2 b=1 print (a!=b)
a=6 b=11 print (a<b)
a=9 b=7 print (a>b)

{03} Karşılaştırma Operatörleri

İşlem	Sonuç
a=3 b=4 print (a==b)	False
a=6 b=6 print (a==b)	True
a=2 b=1 print (a!=b)	True
a=6 b=11 print (a<b)	True
a=9 b=9 print (a>=b)	True

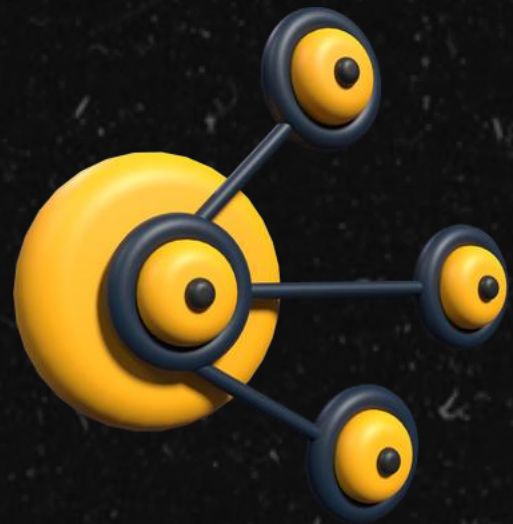
{03} Karşılaştırma Operatörleri

Bakiyesi 20 TL olan bir müşteri, fiyatı 40 TL olan bir ürünü alabilir mi?
Ekрана True veya False yazdıran algoritmayı yazınız.



{03} Karşılaştırma Operatörleri

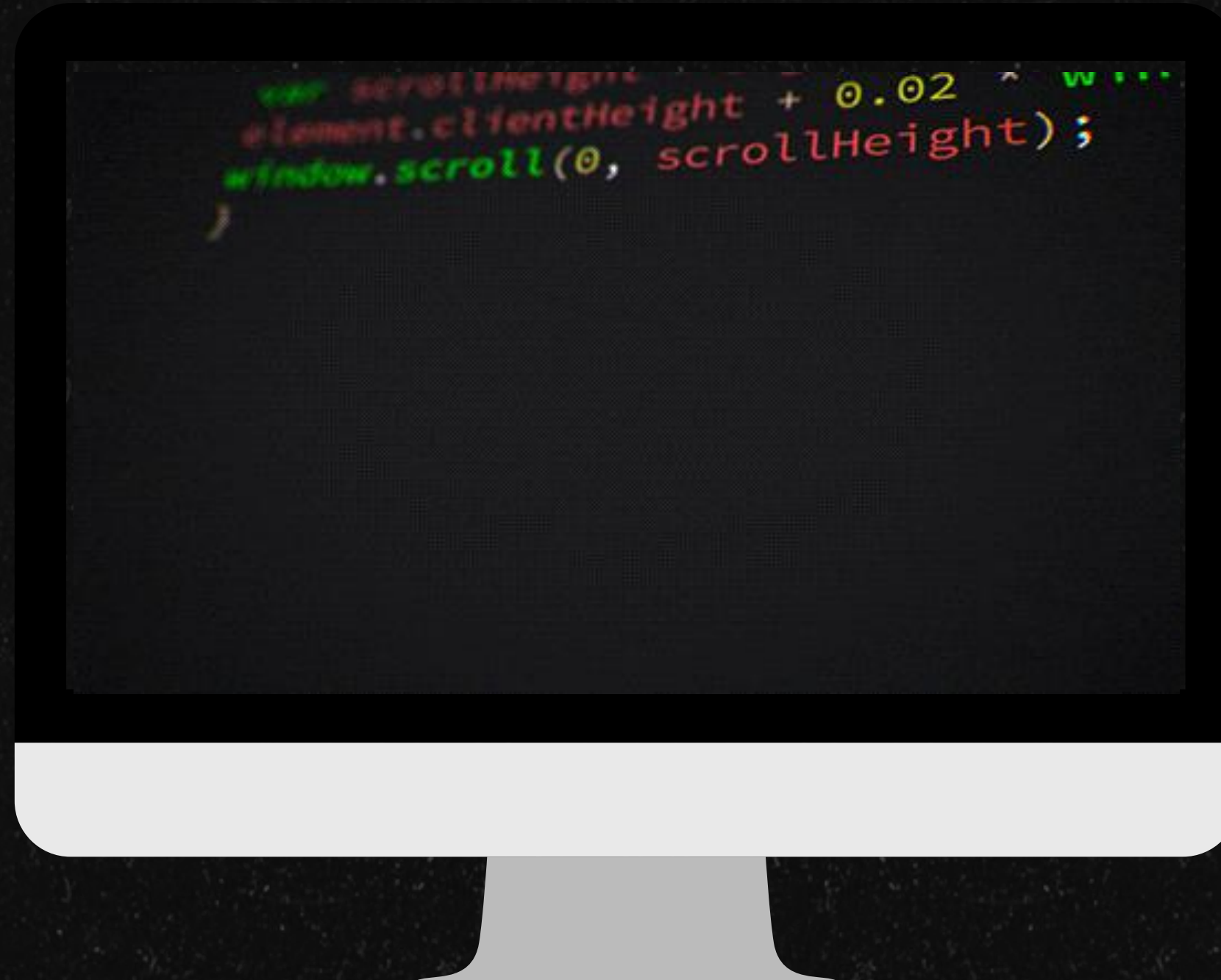
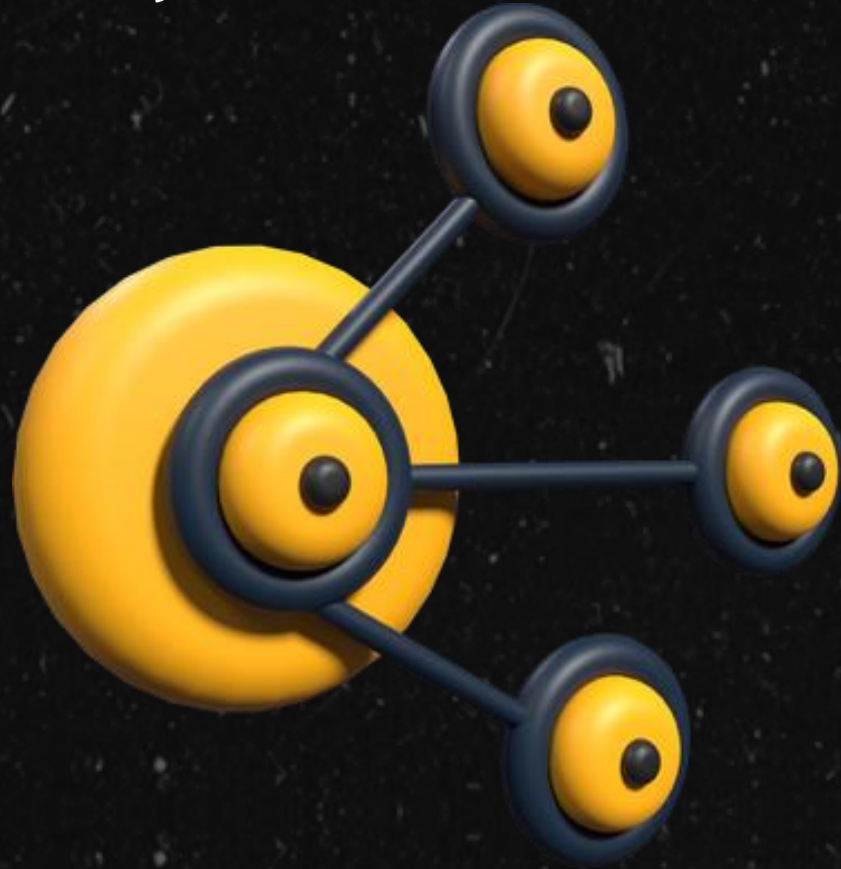
Bakiyesi 20 TL olan bir müşteri, fiyatı 40 TL olan bir ürünü alabilir mi?
Ekрана True veya False yazdıran algoritmayı yazınız.



```
bakiye = 20  
urun_fiyati = 40  
print('Müşteri ürünü alabilir:', bakiye >= urun_fiyati)
```

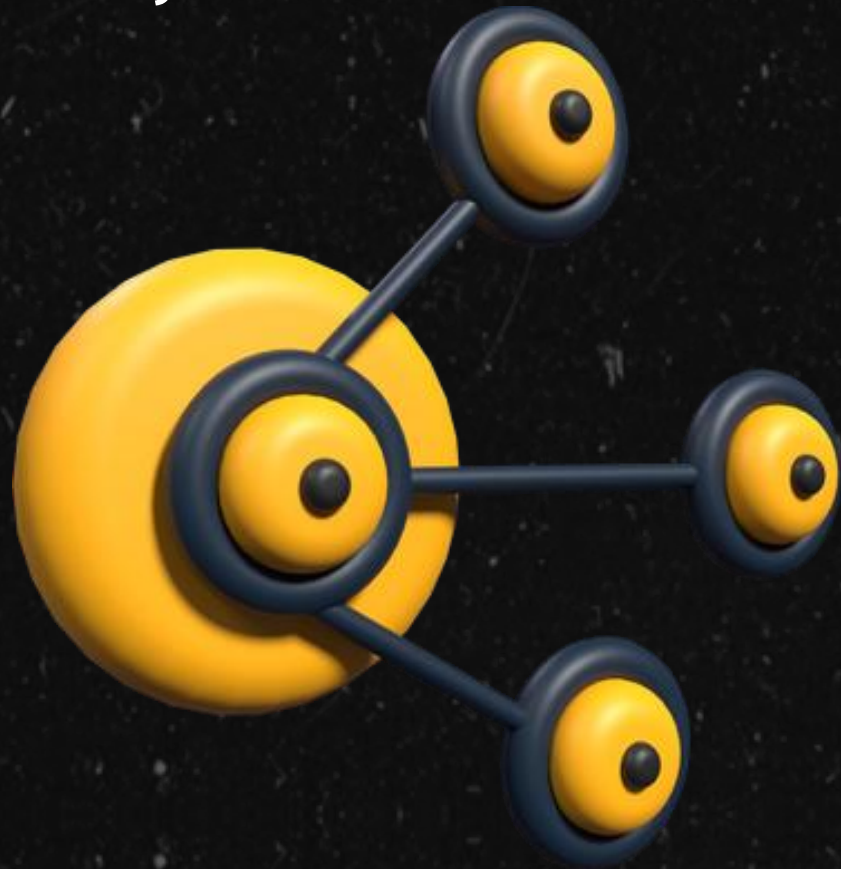
{03} Karşılaştırma Operatörleri

Bir öğrencinin yaşı 17. Bu öğrencinin reşit sayılmak için 18 yaşında olmak gerekiyor. Bu öğrencinin reşit olup olmadığını ekrana True / False olarak yazdırınız.



{03} Karşılaştırma Operatörleri

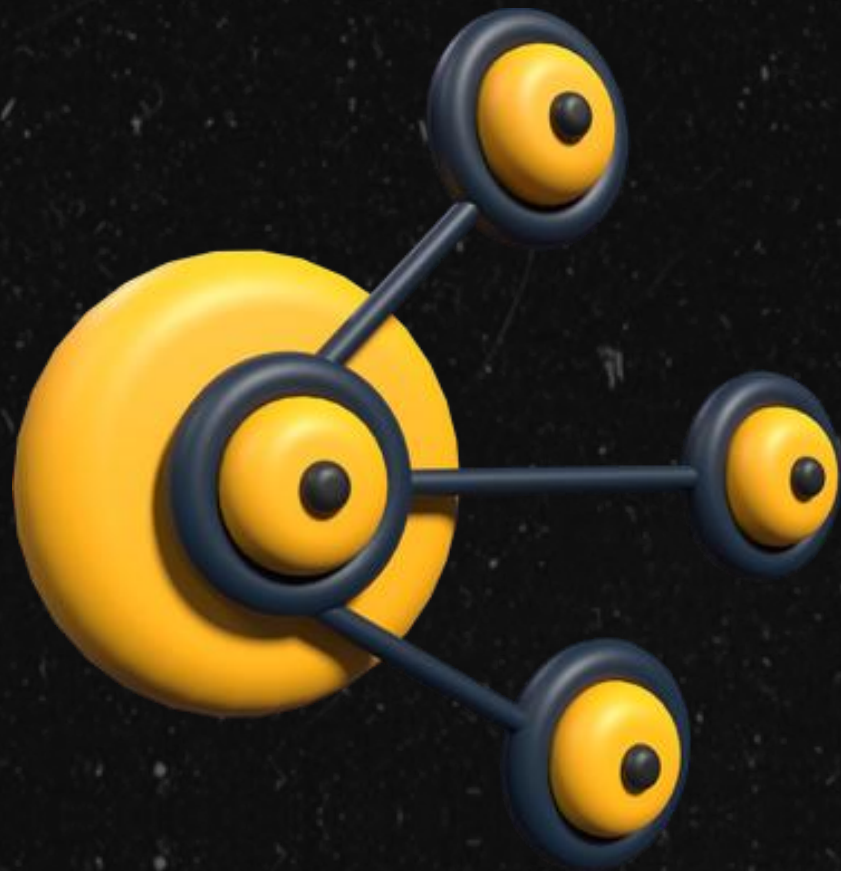
Bir öğrencinin yaşı 17. Bu öğrencinin reşit sayılmak için 18 yaşında olmak gerekiyor. Bu öğrencinin reşit olup olmadığını ekrana True / False olarak yazdırınız.



```
yas = 17  
print('Öğrenci reşit mi:', yas >= 18)
```

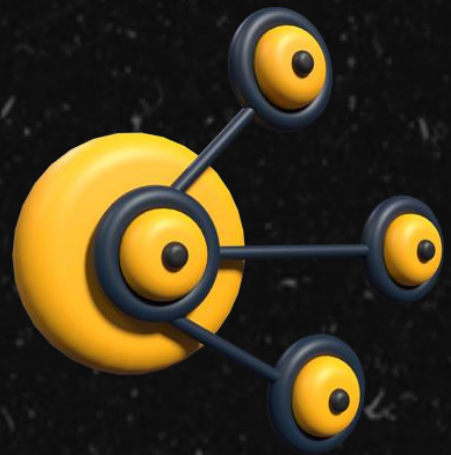
{03} Karşılaştırma Operatörleri

Doğum tarihi 2009 olan birinin reşit olup olmadığını True / False olarak ekrana yazdıran algoritmayı yazınız.



{03} Karşılaştırma Operatörleri

Doğum tarihi 2009 olan birinin reşit olup olmadığını True / False olarak ekrana yazdıran algoritmayı yazınız.

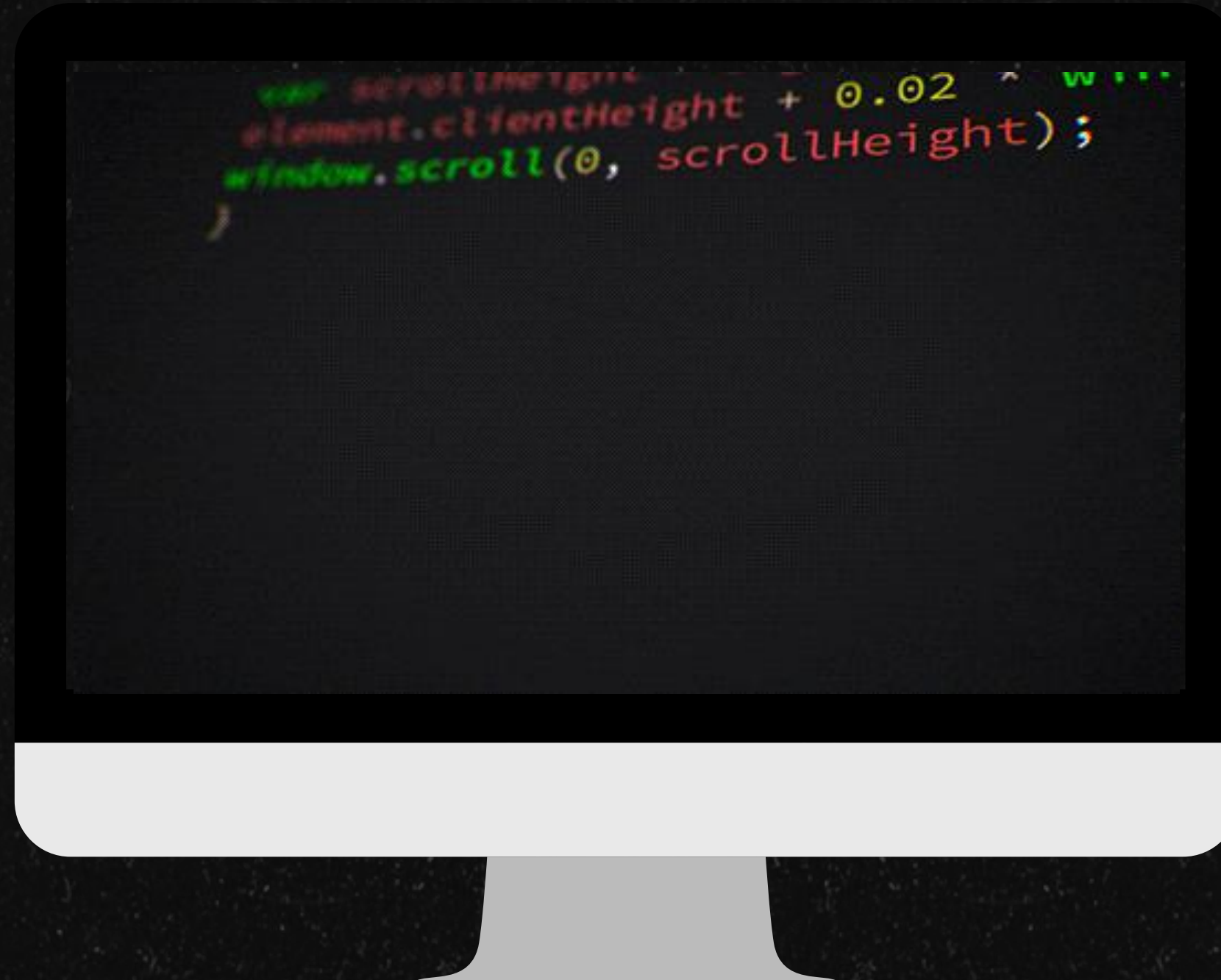
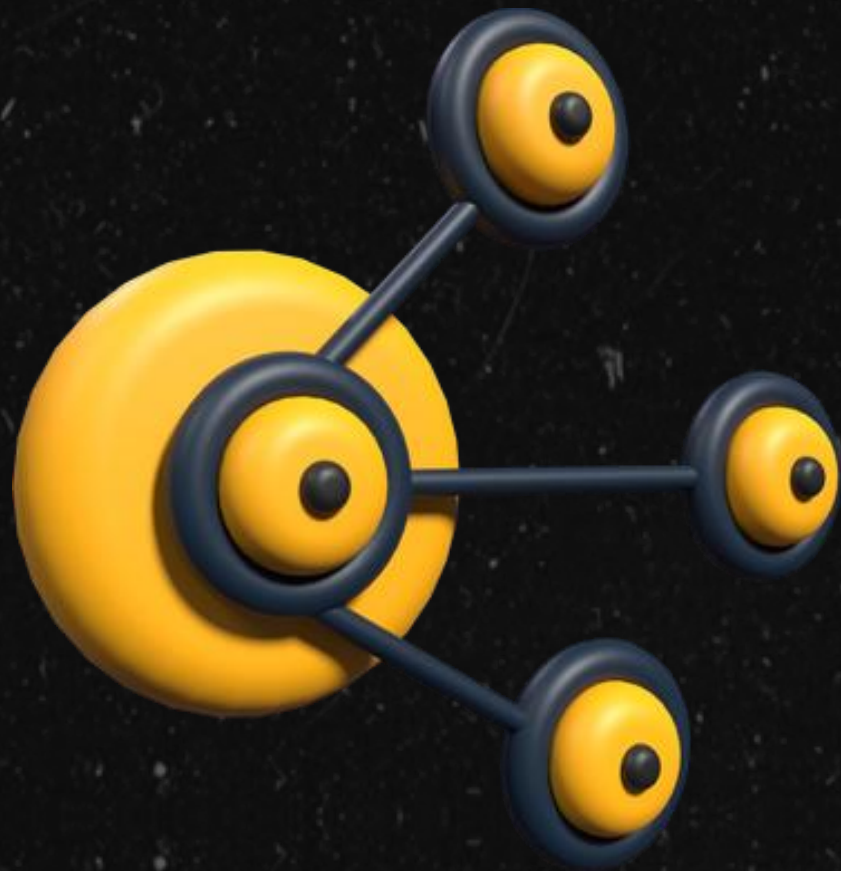


```
dogum_yili = 2009
mevcut_yil = 2025
yas = mevcut_yil - dogum_yili

print('Kişi reşit mi:', yas >= 18)
```

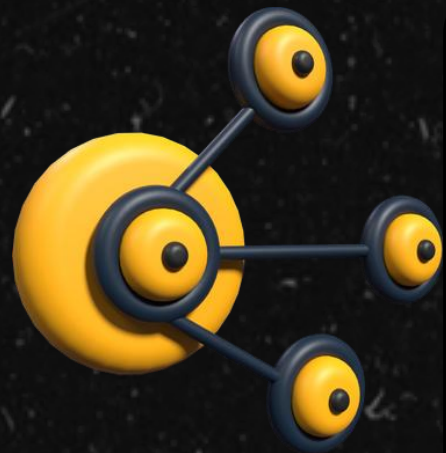
{03} Karşılaştırma Operatörleri

Kayıtlı şifre "abc123" ve girilen şifre "abc123".Girilen şifre doğru mu True / False olarak yazdırınız.



{03} Karşılaştırma Operatörleri

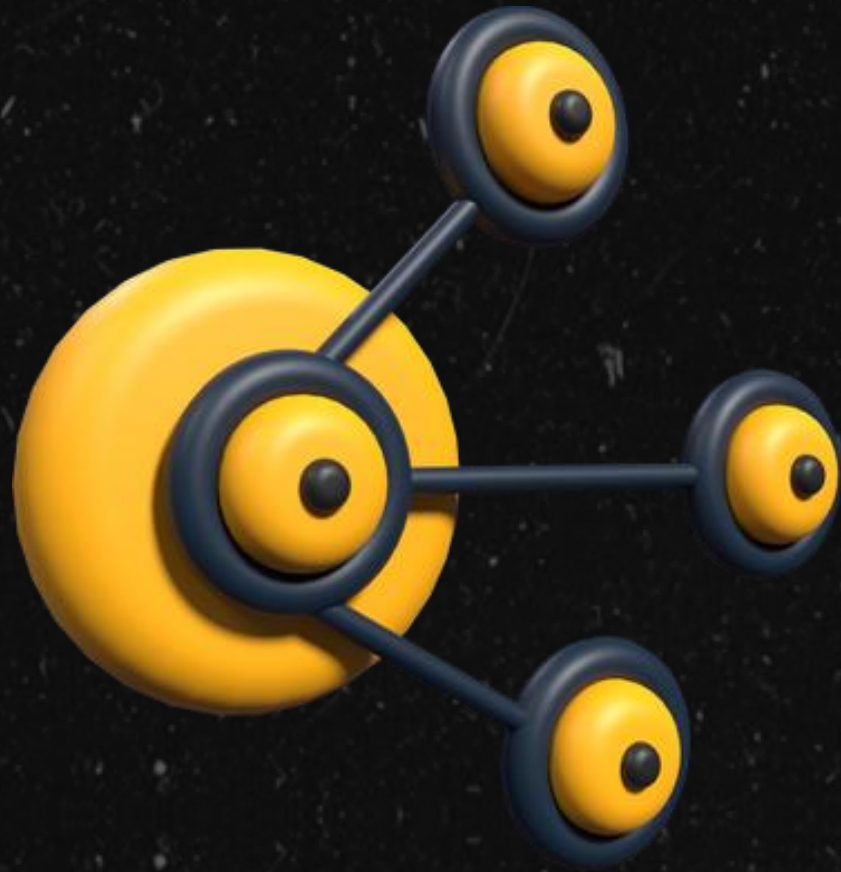
Kayıtlı şifre "abc123" ve girilen şifre "abc123".Girilen şifre doğru mu True / False olarak yazdırınız.



```
kayitli_sifre = "abc123"  
girilen_sifre = "abc123"  
print('Şifre doğru mu:', kayitli_sifre == girilen_sifre)
```

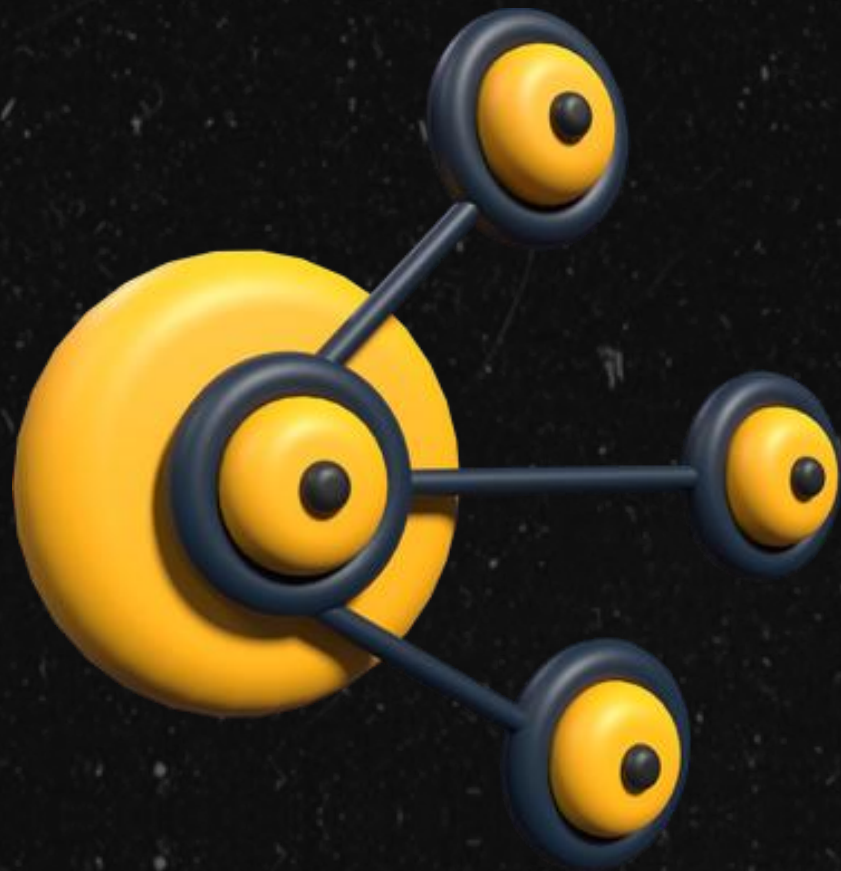
{03} Karşılaştırma Operatörleri

1071 sayısının çift ise ekrana True tek ise False yazdıran algoritmayı yazınız.



{03} Karşılaştırma Operatörleri

1071 sayısının çift ise ekrana True tek ise False yazdıran algoritmayı yazınız.

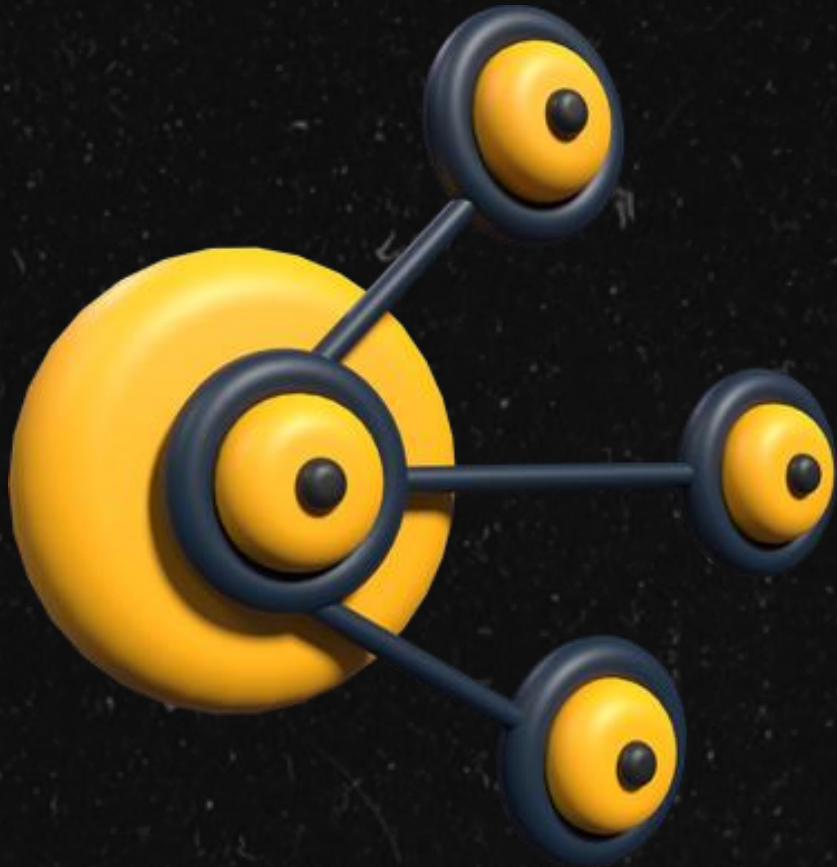


```
sayi = 1071  
print('Sayı çift mi:', sayi % 2 == 0)
```

{03} Karşılaştırma Operatörleri

Bir öğrencinin ödev notları sırasıyla: 40, 58, 99, 65, 100, 85 olarak girilmiştir.

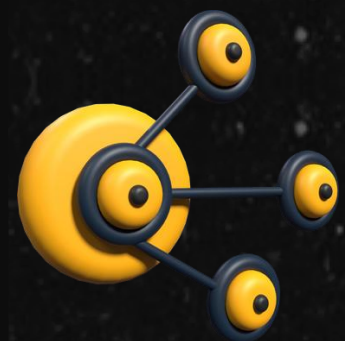
Eğer ödev ortalaması 70 üzeri ise geçtiğini True, kaldı ise False olarak ekrana yazdırınız. Ek olarak ortalamasını da ekrana yazdırınız.



{03} Karşılaştırma Operatörleri

Bir öğrencinin ödev notları sırasıyla: 40, 58, 99, 65, 100, 85 olarak girilmiştir. Eğer ödev ortalaması 70 üzeri ise geçtiğini True, kaldı ise False olarak ekrana yazdırınız. Ek olarak ortalamasını da ekrana yazdırınız.

```
not1 = 40
not2 = 58
not3 = 99
not4 = 65
not5 = 100
not6 = 85
ortalama = (not1 + not2 + not3 + not4 + not5 + not6) / 6
print('Ödev ortalaması:', ortalama)
print('Öğrenci geçti mi:', ortalama >= 70)
```



Mantıksal Operatörler

{04} Mantıksal Operatörler

Operatör	Tanımı	Örnek
and	İki veya daha fazla şartın tamamının doğru olması durumunda True değerini döndürür. Buradaki örnekte a değişkeni 3'ten küçük ve b değişkeni 5'e eşit ya da 5'ten büyük olursa True değeri döndürülür.	a<3 and b>=5
or	İki veya daha fazla şartın en az birinin doğru olması durumunda True değerini döndürür. Buradaki örnekte a değişkeninin 3'ten küçük olması ya da b değişkeninin 4'ten büyük olması True değeri döndürmek için yeterlidir.	a<3 or b>4
not	Durumu tersine çevirmek (True ise False; False ise True) için kullanılır. Buradaki örnekte parantez içindeki mantıksal sınamanın sonucu tersine çevrilir. İfadenin not komutu olmadan yazıldığında true döndüreceği varsayıldığında bu haliyle false döndürecektir.	not(a<3)

{04} Mantıksal Operatörler

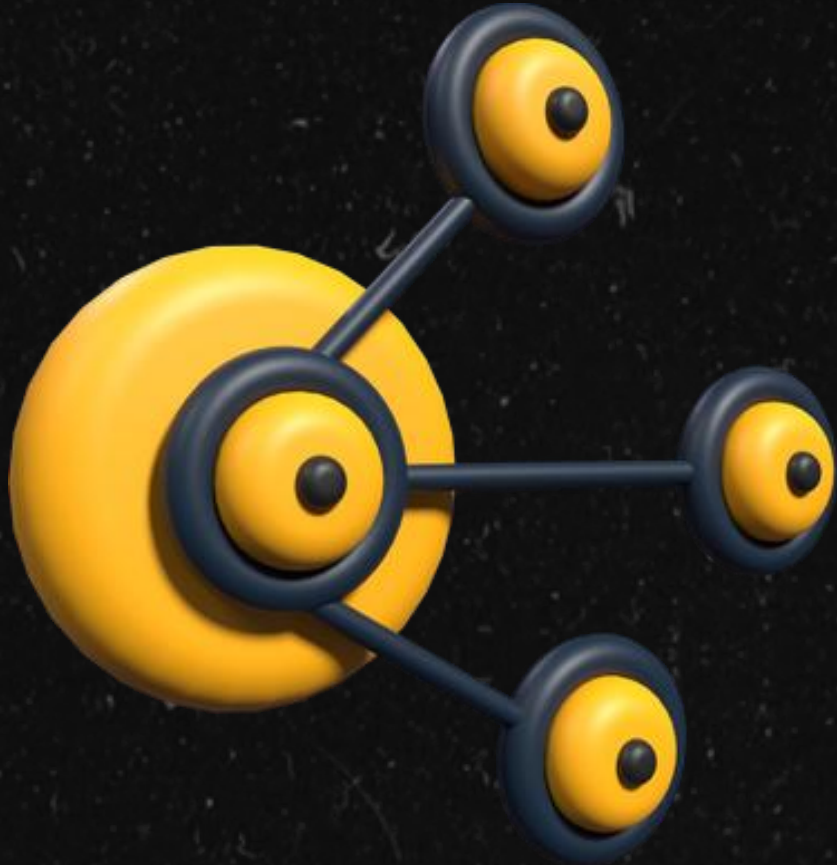
İşlem
a=5 b=3 print (a>b and b>2)
a=6 b=6 print (a==b and a<10)
a=2 b=4 print (a==b or a>b)
a=2 b=4 print (a!=b or a>8)
a=5 b=5 print (not(a>=b and a<1))

{04} Mantıksal Operatörler

İşlem	Sonuç
a=5 b=3 print (a>b and b>2)	True
a=6 b=6 print (a==b and a<10)	True
a=2 b=4 print (a==b or a>b)	False
a=2 b=4 print (a!=b or a>8)	True
a=5 b=5 print (not(a>=b and a<1))	True

{04} Mantıksal Operatörler

Bir öğrencinin sınavdan geçmesi için hem notunun 50'den büyük veya eşit hem de devamsızlık oranının %20'den az olması gerekiyor. Notu 70 ve devamsızlık oranı 30 olan öğrencinin dersten geçip geçmediğini ekrana True / False olarak yazdıran kodu yazınız.

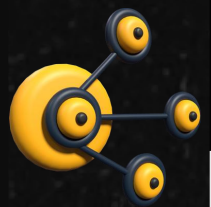


{04} Mantıksal Operatörler

Bir öğrencinin sınavdan geçmesi için hem notunun 50'den büyük veya eşit hem de devamsızlık oranının %20'den az olması gerekiyor. Notu 70 ve devamsızlık oranı 30 olan öğrencinin dersten geçip geçmediğini ekrana True / False olarak yazdıran kodu yazınız.

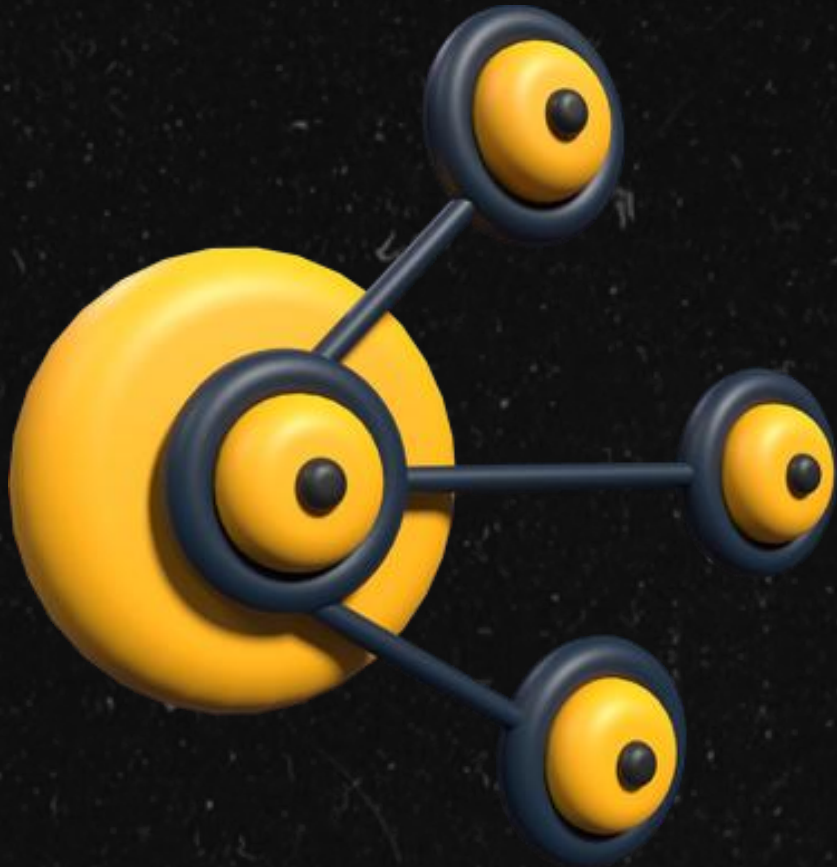
```
notu = 70
devamsizlik_orani = 30

print('Öğrenci geçti mi :', notu >= 50 and devamsizlik_orani < 20)
```



{04} Mantıksal Operatörler

Üniversitede düzenlenen gençlik şenliğinde sadece 18 ile 35 yaş arası etkinliğe alınacaktır. 33 yaşındaki bir katılımcının etkinliğe katılıp katılamayacağı True / False olarak ekrana yazdıran algoritmayı yazınız.

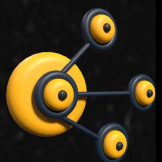


{04} Mantıksal Operatörler

Üniversitede düzenlenen gençlik şenliğinde sadece 18 ile 35 yaş arası(iki yaş dahil) etkinliğe alınacaktır. 33 yaşındaki bir katılımcının etkinliğe katılıp katılamayacağı True / False olarak ekrana yazdıran algoritmayı yazınız.

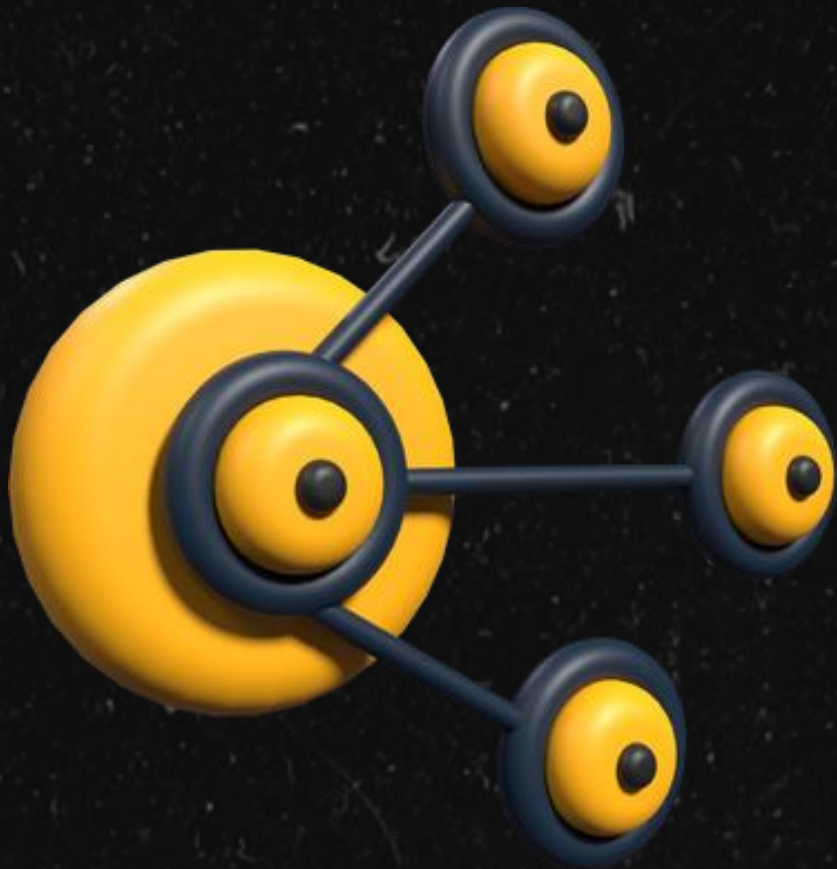
```
yas = 33
```

```
print('Etkinliğe katılabilir mi :', yas >= 18 and yas <= 35)
```



{04} Mantıksal Operatörler

Kullanıcı adı "admin" veya "root" ise ve şifre "1234" ise girişin başarılı olacağı kodu yazınız.



{04} Mantıksal Operatörler

Kullanıcı adı "admin" veya "root" ise ve şifre "1234" ise girişin başarılı olacağı kodu yazınız.

```
kullanici_adi = "admin"  
sifre = "1234"  
  
print('Giriş başarılı mı?', (kullanici_adi == "admin" or kullanici_adi == "root") and sifre == "1234")
```

Kimlik Operatörleri

{05} Kimlik Operatörleri



Python programlama dilinde karşımıza çıkan operatörlerden biri de kimlik operatörleridir.

Kimlik operatörleri değişken değerlerini karşılaştırmak yerine, değişken ya da nesne adreslerini karşılaştırarak sonuç üretir.

Python'da her nesnenin bellek adresini ifade eden kimlik numarası (identity) vardır. Değişken ya da nesnelerin bellek adresleri yani kimlik numaraları `id()` fonksiyonu ile öğrenilebilir.

Kimlik operatörlerini kullanmak için `is` ve `is not` ifadeleri kullanılır.

{05} Kimlik Operatörleri



```
a=100  
b=101  
print(id(a))  
print(id(b))  
a+=1  
print(id(a))  
print (id(b))
```

```
Ekran Çıktısı:  
140729385594352  
140729385594384  
140729385594384  
140729385594384
```

{05} Kimlik Operatörleri



```
a=10
b=11
c=12
print (a is c)
print (a is not b)
a+=1
print (a is b)
a+=1
print(a is c)
```

{05} Kimlik Operatörleri



```
a=10
b=11
c=12
print (a is c)
print (a is not b)
a+=1
print (a is b)
a+=1
print(a is c)
```

ÇIKTI :
False
True
True

SON

Son