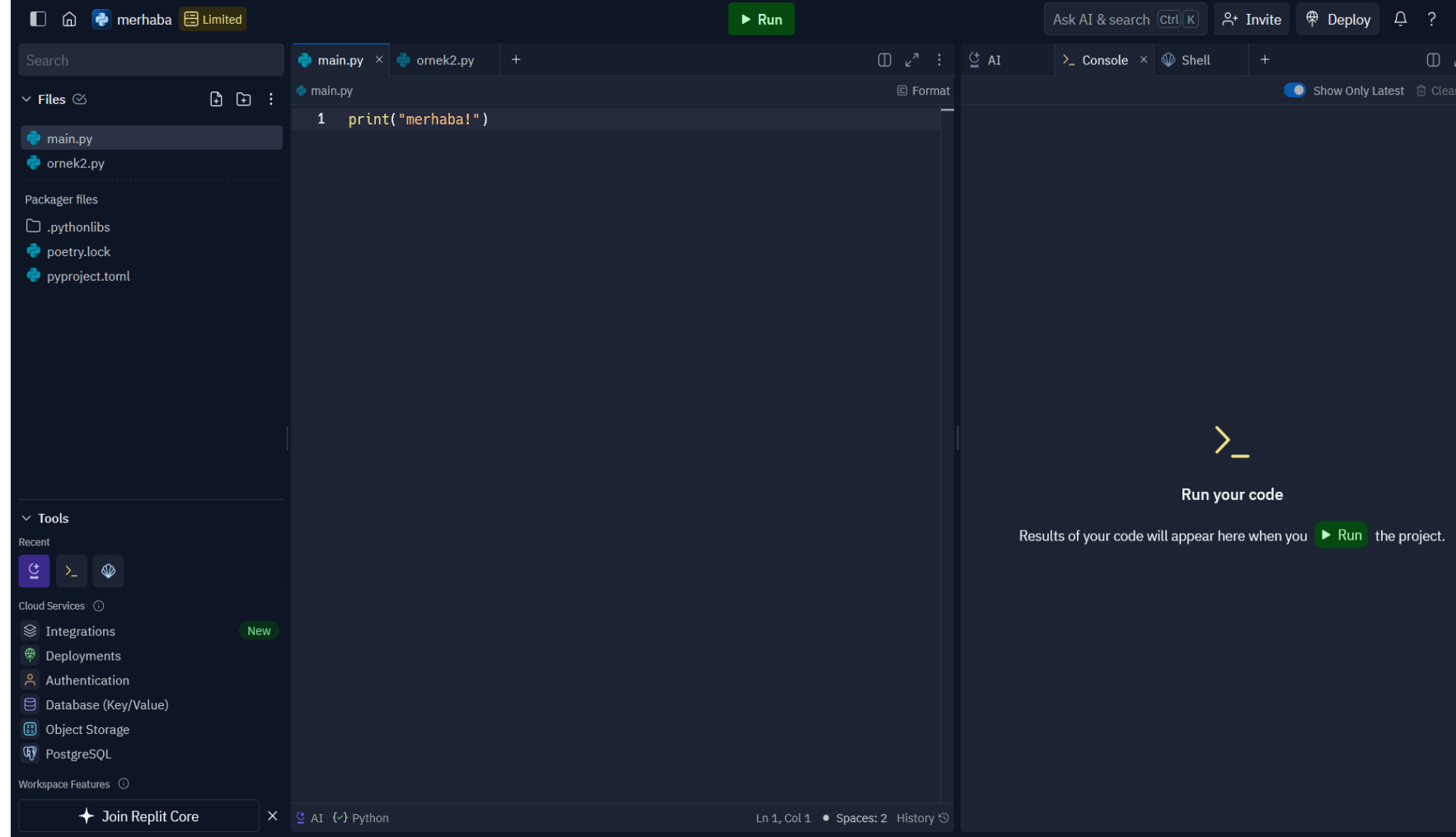


Abstract geometric lines in the top-left corner of the page, consisting of several thin, dark lines that intersect to form various polygons and shapes.

PYTHON DERS NOTLARI

Replit İle Kurulum Gerektirmeden Python Geliştirme

replit ile bir çok alanda yazılım geliřtirmek için gereken alt yapı hazırlanmıř. Tek yapmanız gereken New Repl ile yeni bir proje oluřturmak.





PYTHON İLE VERİ TİPLERİ

Python Sayı Veri Tipleri: Integer(int) Ve Float

Pythonda sayı veri türlerini (numbers) 2 grup altında ele alabiliriz.

Type fonksiyonu: Verinin yada değişkenin veri tipini bildirir.

int tam sayılar

Örneğin 5 tamsayıdır.

- `type(5)` int bilgisini verir

float ondalıklı sayılar

Örneğin 2.5 ondalıklı sayıdır.

- `type(2.5)` float bilgisini verir

Python Metin Veri Tipi: String(str)

Pythonda metin veri türüne str denir. **str metinsel ifadeler**. Değişkenlere sözel bir atama (string) işlemi yaparken tek tırnak ya da çift tırnak kullanabiliriz.

Örneğin Ali metinsel ifadedir.

- `type('Ali')` str bilgisini verir
- `type("Ay")` str bilgisini verir

Python Matematiksel Operatörler

+	Toplama	10 + 15	25
-	Çıkarma	15 - 10	5
*	Çarpma	2 * 5	10
/	Bölme	8 / 2	4.0
%	Mod Alma	15 % 2	1
**	Üs alma	2 ** 3	8
//	Kalansız Bölme	15 // 2	7

Değişken Nedir?

Python programlarımızda geçici olarak veri saklamak için oluşturduğumuz alanlara değişken denir.

Programcılıkta istediğimiz verileri, bilgileri saklamamıza sağlayan yapılara değişken (variable) denilmektedir. Değişkenlere verdiğimiz isimler ile erişebilir, birbirinden ayırabilir, işlemlerimizi yapabilir. Kurallar çerçevesinde değişkenlere verdiğimiz isimlerin bilgisayar için bir önemi yoktur, sadece kodun okunabilirliği ve anlaşılabilir olması için yazılımcılar için önemlidir.

Python Değişken Tanımlama

Tanımlamış olduğumuz x, y, z, a, name, surname değişkenleri bellekte tanımlanan geçici alanlardır.

str

```
name = "Ali"
```

```
surname = 'AY'
```

int, float

```
x = 10
```

```
y = 20
```

```
z = x + y
```

```
a=2.3
```

```
k
```

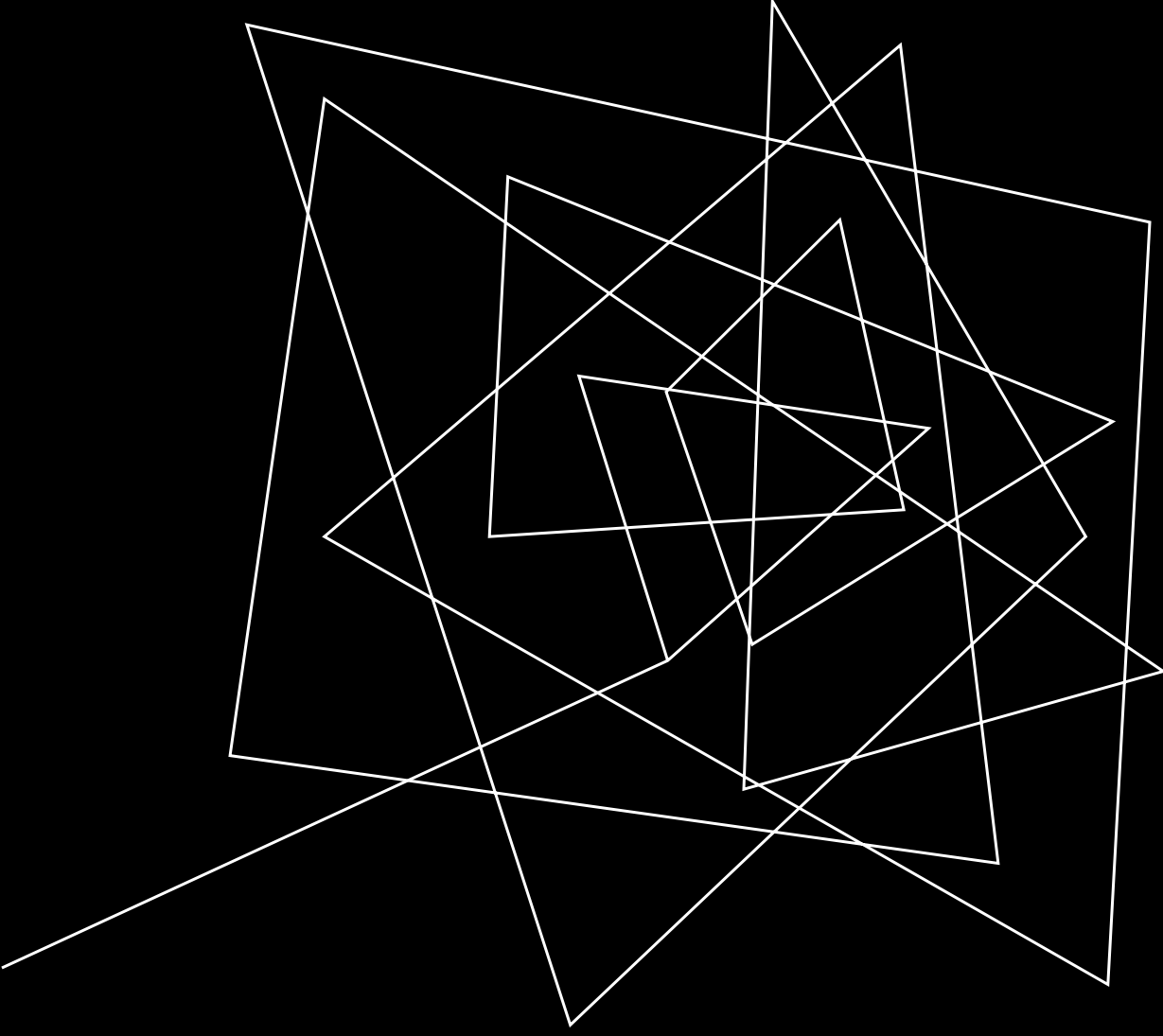
```
print(x) # 10
```

```
print(y) # 20
```

```
print(z) # 30
```

```
print(a) # 2.3
```

```
print(k) # NameError: name 'k' is not defined
```



Değişken İsimlerini
Belirlerken Uyulması
Gerekten Kurallar



Değişken İsimlerini Belirlerken Uyulması Gereken Kurallar

1

Değişken isimleri rakam ile başlayamaz.

1yas => hatalı

yas1 => geçerli



Değişken İsimlerini Belirlerken Uyulması Gereken Kurallar

2

Komut isimleriyle değişken tanımlama yapılamaz.

Örneğin if ya da for kelimesi değişken ismi olamaz.



Değişken İsimlerini Belirlerken Uyulması Gereken Kurallar

3

Değişken isimlerinde türkçe karakter ve özel karakter (.,/*-+ vb.) kullanılmaz.

(Python açısından türkçe karakter kullanımında sıkıntı yok ancak bu alışkanlığı edinmemenizi tavsiye ederim.)



Değişken İsimlerini Belirlerken Uyulması Gereken Kurallar

4

Büyük küçük harf duyarlılığı vardır.

`firstName = 'Ali'`

`FirstName = "Ay"`



Değişken İsimlerini Belirlerken Uyulması Gereken Kurallar

5

Pythonda ayrı satırlarda yapılan değişken tanımlaması aynı satırda da yapılabilir;

```
x, y, name, isStudent = 1, 2.3, 'Ali', True
```

Değişken Örnekleri

```
maasAli = 3000
```

```
maasAhmet = 5000
```

```
vergi = 0.25
```

```
print(maasAli - (maasAli * vergi ))    # ali maaşı
```

```
print(maasAhmet - (maasAhmet * vergi )) # ahmet maaşı
```

Python Veri Tipi Dönüşümleri

`int()` fonksiyonu ile değerleri tam sayıya,
`float()` fonksiyonunu kullanarak değerleri float veri tipine,
`str()` fonksiyonu ile metin veri tipine çevirebiliriz.

Uygulama:

Yarı çapı verilen bir dairenin
alan ve çevresini
hesaplayınız.

```
# Daire Alanı : pi*r**2  
# Daire Çevresi : 2*pi*r
```

```
pi = 3.14  
r = float(input("yarı çap: "))
```

```
alan = pi * (r ** 2)  
print(type(alan))
```

```
cevre = 2 * pi * r  
print(type(cevre))
```

```
print("alan: " + str(alan) + " çevre: " + str(cevre))
```

STRING FORMATLAMA

+' operatörünü kullanarak string birleştirme işlemi bazen zor olabiliyor. Dolayısıyla kullanabileceğimiz **format()** metodu ile **f-string** isminde iki farklı alternatifimiz mevcut.

format()

.format() metoduna vereceğimiz her parametre sırasıyla { }' lerin yerine kopyalanır.

```
print('My name is {} {}'.format(name, surname)) # My name is Çınır Turan
```

--{ }' ler için indeks numarası da verebiliriz. Yani format metodu içindeki name, 0 ve surname, 1 değerlerini alır.

```
print('My name is {0} {1}'.format(name, surname)) # My name is Turan Çınar
```

STRING FORMATLAMA

f-string

f-string ile kolaylıkla string birleştirme işlemi yapabiliriz. Çünkü string ifadenin içinde süslü parantezler içinde değişkenleri yazabiliriz.

```
print(f"My name is {name} {surname} and I'm {age} years old.")
```

STRING PARÇALAMA (STRING SLICING) - 1

String veri tipleri bir karakter dizisi olarak anılır yani bir karakter listesi olarak tanımlanırlar. Dizilerin her bir karakterine soldan 0' dan başlayarak bir indeks numarası atanır. Atanan indeks numaraları dizi içerisindeki her bir elemana ulaşırken kullanılır.

```
Greeting="Merhaba"  
print(greeting[0])  # M  
print(greeting[1]) # e  
print(greeting[-1]) # a
```

Karakter dizilerinin daha doğrusu dizilerin eleman sayısını **len()** metodu ile bulabiliriz.

```
print(greeting[len(greeting)-1]) # a
```

STRING PARÇALAMA (STRING SLICING) - 2

Bir karakter dizisini başlangıç ve bitiş aralığında istediğimiz şekilde bölebiliriz.

```
greeting="Merhaba"  
print(greeting[1:3]) # er  
print(greeting[:3]) # Mer  
print(greeting[3:]) # haba
```

Karakter dizilerinin daha doğrusu dizilerin eleman sayısını **len()** metodu ile bulabiliriz.

```
print(greeting[len(greeting)-1]) # a
```

Adımlama ile her bir karakterden sonra kaç karakter atlanması gerektiğini de belirtebiliriz.

```
print(greeting[0:7:2]) # Mraa  
print(greeting[0:len(greeting):2]) # Mraa
```

UYGULAMA

```
name, surname, age, job = 'Bora','Yılmaz', 32, 'mühendis'
```

Yukarıda verilen değişkenler ile ekrana aşağıdaki ifadeyi yazdırın.

'Benim adım Bora Yılmaz, Yaşım 32 ve mesleğim mühendis.'

```
result = "Benim adım "+ name+ " " + surname+ ", Yaşım "+ str(age) + " ve mesleğim "+ job
result = "Benim adım {0} {1}, Yaşım {2} ve mesleğim {3}.".format(name,surname,age,job)
result = f'Benim adım {name} {surname}, Yaşım {age} ve "mesleğim" {job}.'
```

'abc' ifadesini yan yana 3 defa yazdırın.

```
result = 'abc ' * 3
```

PYTHON STRING METOTLARI - 1

- **split Metodu:** Split metodu, karakter dizisinde belirtilen bir karaktere göre parçalama işlemi yapar.
- **upper Metodu:** Upper metodu, karakterleri büyük harfe çevirir.
- **lower Metodu:** Karakterleri küçük harfe çevirir.
- **title() metodu:** karakter dizisindeki her kelimenin baş harfini büyük harfe çevirir.
- **capitalize():** karakter dizisindeki sadece ilk kelimenin baş harfini büyük harfe çevirir.

PYTHON STRING METOTLARI - 2

```
message = 'Hello, There.'  
message = message.split()    # ['Hello,', 'There.']  
message = message.upper()    # HELLO THERE.  
message = message.lower()    # hello there.
```

PYTHON STRING METOTLARI - 3

- **replace Metodu:** Replace metodu karakter güncellemesi için kullanılır.
- **find Metodu:** Find metodu verilen string ifade içinde arama yapar ve bulduğu ilk indeks numarasını döndürür. Eğer bulamazsa exception döndürür.
- **index Metodu:** index metodu verilen string ifade içinde arama yapar ve bulduğu ilk indeks numarasını döndürür. Eğer bulamazsa find metodundan farklı olarak geriye -1 değerini döndürür.

PYTHON STRING METOTLARI - 4

```
message = 'My name is Sadık Turan'  
message = message.replace('Sadık','Çınar') # My name is Çınar Turan
```

```
txt = "My name is Sadık Turan."  
x = txt.find("name")  
print(x) # x = 3
```

```
txt = "My name is Sadık Turan."  
x = txt.index("name")  
print(x) # x = 3
```



python

PYTHONDA KOŞUL İFADELER

```
9 from watson.com
10 from watson.com
11
12
13 ACCEPTABLE_RETURN
14
15
16 class Base(Containe
17
18 """The base class
19
20 Attributes:
21     __action__ (st
22
23 """
24
25 def execute(self, **
26     method = self.ge
27     self.__action__ =
28     return method(**kw
29
30 def __str__(self):
31     return self.__class__.__name__
32
33 def __repr__(self):
34     return self.__str__()
```

KOŞULLU İFADELER

Pythonda koşul durumlar ile bir koşuldan elde ettiğimiz True ya da False değere göre programın farklı yönlerden ilerlemesini sağlayabiliriz.

Örneğin; kullanıcıdan aldığımız bir parolanın ya da email adresinin doğru olup olmadığını kontrol ederek 'uygulamaya giriş yapıldı' ya da 'parolanız yanlış' şeklinde mesajlar verebiliriz.

Python Karşılaştırma Operatörler

Operatör Adı	Sembolü	Örnek
Eşittir	==	ad=='özge'
Eşit Değildir	!=	ad!='özge'
Büyüktür	>	a>45
Küçüktür	<	a<45
Büyük Eşittir	>=	5>=a
Küçük Eşittir	<=	a<=5

Python Karşılaştırma Operatörler

Operatör Adı	Sembolü	Örnek
Eşittir	==	ad=='özge'
Eşit Değildir	!=	ad!='özge'
Büyüktür	>	a>45
Küçüktür	<	a<45
Büyük Eşittir	>=	5>=a
Küçük Eşittir	<=	a<=5

Python Mantıksal Operatörler

Operatör Adı	Sembolü	Örnek
Ve (and)	and	a<4 and a>8
veya (or)	or	a<4 or a<3
değil (not)	not	not(a==b)

PYTHONDA IF BLOĞU

Bir koşulun True ya da False olan sonucuna göre farklı kod blokları oluşturmak için If komutunu kullanırız.

```
a = 10
b = 20
if b > a:
    print("b, a'dan büyüktür")
```

PYTHONDA ELIF BLOĞU

Bazen birden fazla koşul yazmak isteriz bu durumda eğer ilk if bloğundaki koşul False değer üretirse Elif bloğunda tanımladığımız koşula bakılır.

```
a = 10
b = 10
if b > a:
    print("b, a'dan büyüktür")
elif a == b:
    print("a ile b eşittir")
```

PYTHONDA ELSE BLOĞU

Else ve Elif bloklarındaki koşullardan her hangi biri True değilse bir Else bloğu oluşturup Else bloğundaki kodların çalıştırılmasını sağlayabiliriz.

```
a = 15
b = 10
if b > a:
    print("b, a'dan büyüktür")
elif a == b:
    print("a ile b eşittir")
else:
    print("a, b'den büyüktür")
```

PYTHONDA İÇ İÇE İF BLOKLARI

Bir if bloğu içerisinde başka bir if bloğunu başlatabiliriz.

```
username = 'sadikturan'
password = '1234'

if (username == 'sadikturan'):
    if (password == '12345'):
        print('Hoş geldiniz')
    else:
        print('parola yanlış')
else:
    print('username yanlış')
```

KOŞUL İFADELERİNDE AND VE OR OPERATÖRLERİ

Koşul ifadelerinde and ve or operatörleri yardımıyla birden fazla durumu aynı anda kontrol edebiliriz.

```
if (username == 'sadikturan') and (password == '12345'):
    print('Hoş geldiniz')
else:
    print('username ya da parola yanlış')
```

PYTHON KOŞUL İFADELERİ ÖRNEKLERİ

Kullanıcıdan isim, yaş ve eğitim bilgilerini isteyip ehliyet alabilme durumunu kontrol eden python uygulamasını yapınız.

**** Ehliyet alma koşulu en az 18 ve eğitim durumu lise ya da üniversite olmalıdır.**

```
isim = input('isminiz: ')
yas = int(input('yaşınız: '))
egitim = input('eğitim: ')

if (yas>=18):
    if (egitim=='lise' or egitim=='üniversite'):
        print(f'{isim} ehliyet alabilirsin.')
    else:
        print(f'{isim} ehliyet alamazsın eğitim durumun
yetersiz.')
else:
    print(f'{isim} ehliyet alamazsın yaşın tutmuyor.')
```

PYTHON KOŞUL İFADELERİ ÖRNEKLERİ

Kullanıcıdan isim, yaş ve eğitim bilgilerini isteyip ehliyet alabilme durumunu kontrol eden python uygulamasını yapınız.

**** Ehliyet alma koşulu en az 18 ve eğitim durumu lise ya da üniversite olmalıdır.**

```
isim = input('isminiz: ')
yas = int(input('yaşınız: '))
egitim = input('eğitim: ')

if (yas>=18):
    if (egitim=='lise' or egitim=='üniversite'):
        print(f'{isim} ehliyet alabilirsin.')
    else:
        print(f'{isim} ehliyet alamazsın eğitim durumun yetersiz.')
else:
    print(f'{isim} ehliyet alamazsın yaşın tutmuyor.')
```

PYTHON KOŞUL İFADELERİ ÖRNEKLERİ

Bir öğrencinin 2 yazılı bir sözlü notunu alıp hesaplanan ortalamaya göre not aralığına karşılık gelen not bilgisini yazdıran python uygulamasını yapınız.

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

```
yazili1 = float(input('1.yazılı: '))
yazili2 = float(input('2.yazılı: '))
sozlu = float(input('sözlü: '))

ortalama = (yazili1 + yazili2 + sozlu)/3

if (ortalama>=0) and (ortalama<25):
    print(f'ortalamanız: {ortalama} notunuz: 0')
elif (ortalama >= 25 ) and (ortalama<45):
    print(f'ortalamanız: {ortalama} notunuz: 1')
elif (ortalama >= 45 ) and (ortalama<55):
    print(f'ortalamanız: {ortalama} notunuz: 2')
elif (ortalama >= 55 ) and (ortalama<70):
    print(f'ortalamanız: {ortalama} notunuz: 3')
elif (ortalama >= 70 ) and (ortalama<85):
    print(f'ortalamanız: {ortalama} notunuz: 4')
elif (ortalama >= 85 ) and (ortalama<=100):
    print(f'ortalamanız: {ortalama} notunuz: 5')
else:
    print('yanlış bilgi girdiniz.')
```

PYTHON KOŞUL İFADELERİ ÖRNEKLERİ

Girilen 3 sayıyı büyüklük olarak karşılaştıran python uygulamasını yapınız.

```
a = int(input('a: '))
b = int(input('b: '))
c = int(input('c: '))

if (a > b) and (a > c):
    print(f'a en büyük sayıdır.')
elif (b > a) and (b > c):
    print(f'b en büyük sayıdır.')
elif (c > a) and (c > b):
    print(f'c en büyük sayıdır.')
```

PYTHON KOŞUL İFADELERİ ÖRNEKLERİ

Kullanıcıdan 2 vize (%60)

Eğer ortalama 50 ve üstündeyse geçti değilse kaldı yazdırın.

a-) Ortalama 50 olsa bile final notu en az 50 olmalıdır.

b-) Finalden 70 alındığında ortalamanın önemi olmasın.

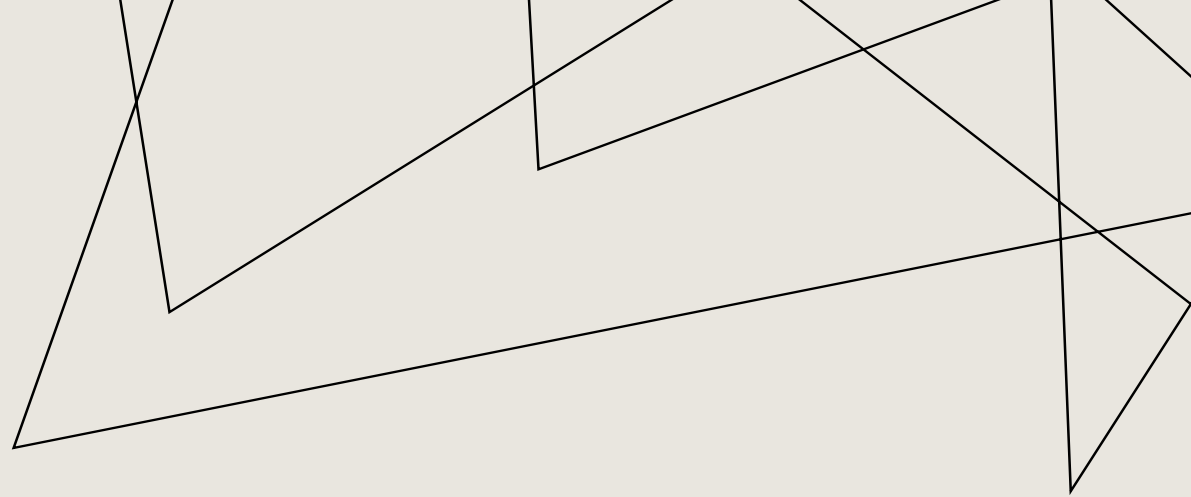
ve final (%40) notunu alıp ortalama hesaplayan python uygulamasını yapınız.

PYTHON KOŞUL İFADELERİ ÖRNEKLERİ

```
vize1 = float(input('vize 1: '))  
vize2 = float(input('vize 2: '))  
final = float(input('final : '))
```

```
ortalama = ((vize1+vize2)/2)*0.6 + (final * 0.4)
```

```
result = (ortalama>=50) and (final>=50)  
result = (ortalama >=50) or (final>=70)
```



PYTHON KOŞUL İFADELERİ ÖRNEKLERİ

```
** durum-1
```

```
if (ortalama>=50):  
    if (final>=50):  
        print(f'öğrencinin ortalaması: {ortalama} ve geçme durumu: başarılı')  
    else:  
        print(f'öğrencinin ortalaması: {ortalama} ve geçme durumu: başarısız.  
Finalden en az 50 almalısınız.')  
else:  
    print(f'öğrencinin ortalaması: {ortalama} ve geçme durumu: başarısız')
```

PYTHON KOŞUL İFADELERİ ÖRNEKLERİ

```
** durum-2
```

```
if (ortalama >=50):  
    print(f'öğrencinin ortalaması: {ortalama} ve geçme durumu: başarılı')  
else:  
    if (final>=70):  
        print(f'öğrencinin ortalaması: {ortalama} ve geçme durumu: başarılı.  
Finalden en az 70 aldığınız için geçtiniz.')  
    else:  
        print(f'öğrencinin ortalaması: {ortalama} ve geçme durumu: başarısız')
```

PYTHON KOŞUL İFADELERİ ÖRNEKLERİ

Kişinin ad, kilo ve boy bilgilerini alıp kilo indekslerini hesaplayan python uygulamasını yapınız.

Formül: $(\text{Kilo} / \text{boy uzunluğunun karesi})$

Aşağıdaki tabloya göre kişi hangi gruba girmektedir.

0-18.4 => Zayıf

18.5-24.9 => Normal

25.0-29.9 => Fazla Kilolu

30.0-34.9 => Şişman (Obez)

PYTHON KOŞUL İFADELERİ ÖRNEKLERİ

```
name = input('adınız: ')
kg = float(input('kilonuz: '))
hg = float(input('boyunuz: '))
```

```
index = (kg) / (hg ** 2)
```

```
if (index >= 0) and (index<=18.4):
    print(f'{name} kilo indeksin: {index} ve kilo değerlendirmen zayıf.')
elif (index>18.4) and (index<=24.9):
    print(f'{name} kilo indeksin: {index} ve kilo değerlendirmen normal.')
elif (index>24.9) and (index<=29.9):
    print(f'{name} kilo indeksin: {index} ve kilo değerlendirmen kilolu.')
elif (index>=29.9) and (index<=45.9):
    print(f'{name} kilo indeksin: {index} ve kilo değerlendirmen obez.')
else:
    print('bilgileriniz yanlış.')
```