DİZE(STRINGS)

Tek tırnak ve ya çift tırnak içerisinde yazdığımız tüm metin veri türlerine dize diyporuz.Python metin veri tanımlama kısmında da ele aldığımız bu konuyu detayları ve örneklerle size aktarmak isterim.

Hatırlarsanız len() komutuyla '' içerindeki ifadenin kaç harfden oluştuğunu yani uzunluğunu bize çıktı olarak veriyordu dilerseniz örneğimizde bir metin verisini tek tırnakla yazalım veuzunluğunun ne kadar olduğunu görelim.

sayılar= 's' # tek bir karakter ile dize olabilir.

print(sayılar)  # s

print(len(sayılar))# 1

mesaj='Python Dünyasına Hoş Geldiniz!'# tek tırnakla dize oluşturdum

print(mesaj) #Python Dünyasına Hoş Geldiniz!

print(len(mesaj))#30

#uzun bir cümleyi tek tırnak dize şeklinde yazalım.

cümle='Python programını öğrendiğim için çok mutluyum.'

print(cümle)#Python programını öğrendiğim için çok mutluyum.

Uzun bir cümle yazmak istiyorum bunu iki şekilde de yapabilirim. Birincisi tek tırnakla cümle oluşturmak aşağıdaki örneği inceleyelim.

uzun\_cümle=''' Ben bir istatistikçiyim aslında

bu tam bana göre bir meslek çocukluğumdan beri

sürekli gözlem yaparak etrafımda olup bitenleri

tahmin etmeye çalışırdım.'''

print(uzun\_cümle)

Ben bir istatistikçiyim aslında

bu tam bana göre bir meslek çocukluğumdan beri

sürekli gözlem yaparak etrafımda olup bitenleri

tahmin etmeye çalışırdım.

Aynı işlemi çift tırnakla da yapabiliriz örneğimize bakalım.

uzun\_cümle=""" Ben bir istatistikçiyim aslında

bu tam bana göre bir meslek çocukluğumdan beri

sürekli gözlem yaparak etrafımda olup bitenleri

tahmin etmeye çalışırdım."""

print(uzun\_cümle)

## DİZE BİRLEŞTİRMEK

Dizeleri birbirine bağlayabiliriz. Dizeleri basit bir şekilde birleştirme yapabiliriz aşağıdaki örneği inceleyim.

ad='büşra'

meslek='istatistikçi'

boş= " "

AD\_MESLEK= ad + boş + meslek

print(AD\_MESLEK)

print(len(AD\_MESLEK))#18

print(len(meslek))#12

print(len(meslek)>len(ad))#True

## DİZELERDEKİ KAÇIŞ DİZİLERİ

Python ve diğer programlama dillerinde \ ardından bir karakter gelen bir boşluk dizisidir. Bunun ile ilgili en sık kullanılan kalıplara bakalım.

* \n: Alt satıra geçer
* \t: Sekme anlamına gelir (8 boşluk bırakır)
* \\: Ters eğik çizgi
* \’: Tek alıntı (‘)
* \”: Çift alıntı (“)

Bu kalıplarla ilgili aşağıdaki örneklerimizi inceleyelim.

* \n: Alt satıra geçer

print ( 'Umarım herkes rahatlıkla kod yazabiliyor \n Öyle mi?' )

Umarım herkes rahatlıkla kod yazabiliyor

 Öyle mi?

* \t: Sekme anlamına gelir (8 boşluk bırakır)

ÖRNEK:

print('il\tölüm sayısı\tVakaSayısı')

print('Adana   \t8\t587')

print('Diyarbakır\t14\t547')

print('Edirne  \t8\t500')

print('Giresun  \t8\t647')

print('İstanbul \t13\t1050')

print('Kayseri \t18\t897')

print('Mersin    \t11\t201')

ÇIKTI:

il   ölüm sayısı VakaSayısı

Adana       8   587

Diyarbakır  14  547

Edirne      8   500

Giresun     8   647

İstanbul    13   1050

Kayseri     18 897

Mersin      11   201

* \”: Çift alıntı (“)

print('Program açıldığında il yazdığım \"Merhaba, Dünya!\" oldu.')

Program açıldığında il yazdığım "Merhaba, Dünya!" oldu.

## DİZE BİÇİMLENDİRME

### (%) operatorü

Python'da dizeleri biçimlendirmenin birçok yolu vardır. Bu bölümde sizlere % yardımıyla dize şekillendirmesinden bahsedeceğiz.

* %s: Dize (veya sayılar gibi dize temsili olan herhangi bir nesne)
* %d:Tamsayılar
* %f: Ondalık sayılar
* "%.number of digitsf":Sabit hassasiyetli kayan noktalı sayılar

Bu kalıplarla ilgili aşağıdaki örnekleri inceleyelim. Birinci %s ad ,ikinci %s soy\_ad ,üçüncü %s program değişkenini yazdırıyor.

ad= 'Büşra'

soy\_ad = 'Akkaya'

program = 'Python'

cümle\_oluşturma = 'Ben %s %s. Ben %s öğreniyorum.' %(ad, soy\_ad, program)

print(cümle\_oluşturma)

ÇIKTI:

Ben Büşra Akkaya. Ben Python öğreniyorum.

Örneğin elimde bir liste var ve bunu cümleme eklemek istiyorum %s' % ifadesi ile eklenir. Aşağıdaki örnekle inceleyelim:

matematik\_konu = ['Köklü Sayılar', 'Üslü Sayılar', 'Problemler', 'Temel Kavramlar','Rasyonel Sayılar']

cümle\_ekle = 'Matematikte bir çok konu vardır.Bunlardan en temelleri:%s' % (matematik\_konu)

print(cümle\_ekle)

ÇIKTI:

Matematikte bir çok konu vardır.Bunlardan en

temelleri:['Köklü Sayılar', 'Üslü Sayılar',

'Problemler', 'Temel Kavramlar', 'Rasyonel Sayılar']

### (str.format())

Bu format Python'ın yeni sürümünde görmekteyiz. Örnekleri aşağıdaki şekilde görmekteyiz.

{} ifadesinden sonra herhangi bir kelime gelmez.

ad  =  'Büşra'

soy\_ad=  'Akkaya'

dil  =  'Python anlatacağım'

yeni\_cümle=  'Ben {} {}. Sizlere {}'  . format(ad , soy\_ad , dil)

print( yeni\_cümle )

ÇIKTI:

Ben Büşra Akkaya. Sizlere Python anlatacağım

{} ifadesi ile biraz dört işlem yapalım aşağıdaki örneği inceleyelim. Burada birinci satıra bakalım Birinci {} ifadesi x, değerini ikinci {} ifadesi y değerini, üçüncü {} ifadesi x+y değerini anlatmaktadır.

x=10

y=5

print('{} + {} = {}'.format(x, y, x + y))

print('{} - {} = {}'.format(x, y, x - y))

print('{} \* {} = {}'.format(x, y, x \* y))

print('{} / {} = {:.2f}'.format(x, y, x / y))

print('{} % {} = {}'.format(x, y, x % y))

print('{} // {} = {}'.format(x, y, x // y))

print('{} \*\* {} = {}'.format(x, y, x \*\* y))

ÇIKTI:

10 + 5 = 15

10 - 5 = 5

10 \* 5 = 50

10 / 5 = 2.00

10 % 5 = 0

10 // 5 = 2

10 \*\* 5 = 100000

### F-DİZELERİ(f')

Başka bir yeni dize biçimlendirmesi de f-dizeleridir. Arkadaşlar burada verileri karşı gelen komutlara yerleştirebiliriz. Nasıl olduğunu örneğimizde hep beraber görelim.

x=10

y=5

print ( f' { x } + { y } = { x  + y } ' )

print( f' { x } - { y } = { x  -  y } ' )

print( f' { x } \* { y } = { x \*  y } ')

print( f' {x } / { y } = { x /  y } ' )

print ( f' { x } % { y } = { x  %  y } ' )

print ( f' { x } / / { y } = { x  //  y } ' )

print (f' { x } \*\* { y } = { x \*\*  y} ' )

Burada f'{x} f komutuna x i yerleştirdiğimiz anlmına gelir ve diğer adımlar buna göre yapılır.

 10 + 5 = 15

 10 - 5 = 5

 10 \* 5 = 50

 10 / 5 = 2.0

 10 % 5 = 0

 10 / / 5 = 2

 10 \*\* 5 = 100000

## Karakter Dizileri Olarak Python Dizeleri

Herhangi bir diziden bireysel bir elemanın çıkmasını istiyorum elimdeki tüm karekterleri açmalıyım bunun için karakter açma diye tanımladığımız örneğe göz atalım.

### Karakter Açma

ÖRNEK:

 PROGRAM\_DİL = 'Python'

a,b,c,d,e,f = PROGRAM\_DİL

print(a) # P

print(b) # y

print(c) # t

print(d) # h

print(e) # o

print(f) # n

### **Dizilerdeki Karakterlere Dizine Göre Erişim**

Yazdığımız her harfin rakamlarla bir karşılığı var elbette programlamada 0 dan başlar ve dizinin uzunluğu son harfin rakamaı -1 olarak adlandırma gerçekleşir.Örneğimizi inceleyim.

dil  =  'Python'

ilk\_harf =  dil [ 0 ]

print ( ilk\_harf) # P

ikinci\_harf =  dil [ 1 ]

print ( ikinci\_harf) # y

son\_hal =  len ( dil ) -  1

last\_letter  =  dil [ son\_hal ]

print ( son\_hal ) # n

### **Python Dizelerini Dilimleme**

Bir diziyi alt dizilere aşağıda gördüğünüz gibi bölebiliriz.

dil  =  'Python'

ilk\_üç =  dil [ 0 : 3 ] # 3 dahil değil

print ( ilk\_üç ) #Pyt

son\_üç =  dil [ 3 : 6 ]

print ( son\_üç )

# Başka bir yol

son\_üç  =  dil [ - 3 :]

print( son\_üç )

son\_üç  =  dil [3 :]

print( son\_üç )

hon

hon

hon

### **Bir Dizeyi Ters Çevirmek**

Python'da kolayca ters çevirme işlemini gerçekleştiririz. (:: -1) komutu ters hale gelmekte olduğunu örneğimizde görebilriz.

selamlama  =  'Merhaba Dünya!'

print ( selamlama [:: - 1 ]) # !aynüD abahreM

# Dize Yöntemleri

Dizeleri biçimlendirmemize izin veren birçok dize yöntemi vardır. Aşağıdaki örnekte bazı dize yöntemlerine bakalım:

* capitalize(): Dizinin ilk karakterini büyük harfe dönüştürür.

değiş= 'iklim değişiyor'

print(değiş.capitalize()) # Iklim değişiyor

* count():Dizedeki alt dize oluşumlarını döndürür, count(substring, start=.., end=..). Başlangıç, saymak için bir başlangıç ​​indekslemesidir ve bitiş, sayılacak son indekstir.

değiş= 'iklim değişiyor'

print(değiş.count('i')) # 4

print(değiş.count('y', 7, 14)) # 1,

print(değiş.count('or')) # 1

* endswith():Bir dizenin belirtilen bir sonla bitip bitmediğini kontrol eder

değiş= 'iklim değişiyor'

print(değiş.endswith('yor'))   #True

print(değiş.endswith('değ')) # False

* expandtabs(): Sekme karakterini boşluklarla değiştirir, varsayılan sekme boyutu 8'dir. Sekme boyutu bağımsız değişkenini alır.

değiş= 'iklim\değişiyor'

print(değiş.expandtabs ())   #iklim   değişiyor

print(değiş.expandtabs (10))   #iklim       değişiyor

* find(): Bir alt dizenin ilk oluşumunun dizinini döndürür, bulunamazsa -1 döndürür

değiş= 'iklim değişiyor'

print(değiş.find('y'))  # 12

print(değiş.find('kl'))#1

* rfind(): Bir alt dizenin son oluşumunun dizinini döndürür, bulunamazsa -1 döndürür

değiş= 'iklim değişiyor'

print(değiş.rfind('y'))  # 12

print(değiş.rfind('kl'))#1

* format(): dizeyi daha güzel bir çıktıya biçimlendirir

ad = 'Büşra'

soy\_ad = 'Akkaya'

yaş= 25

meslek = 'veri analisti'

ülke = 'Türkiye'

cümle= 'Ben  {} {}. Benim yaşım {}. Ben {}yim. Ben  {} yaşıyorum.'.format(ad, soy\_ad, yaş, meslek, ülke)

print(cümle)

Ben  Büşra Akkaya. Benim yaşım 25.

 Ben veri analistiyim. Ben  Türkiye yaşıyorum.

* index(): Bir alt dizinin en düşük dizinini döndürür, ek bağımsız değişkenler başlangıç ​​ve bitiş dizinini belirtir.

“şi” başlangısına bakalım

değiş= 'iklim değişiyor'

ek='de'

print(değiş.index('ek')) #6

print(değiş.index('ek',9))

* isalnum(): Sayısal karakteri kontrol eder

meydan   =  'OtuzGünProgram'

print( meydan . isalnum ()) # Doğru

meydan =  '30GünProgram'

print( meydan . isalnum ()) # Doğru

meydan   =  'otuz gün program'

print ( meydan okuma . isalnum ()) # Yanlış, boşluk bir sayısal karakter algılamaz

meydan  =  'otuz gün python 2019'

print( meydan okuma . isalnum ()) # Yanlış

* isalpha(): Tüm dize öğelerinin alfabe karakterleri olup olmadığını kontrol eder.

meydan   =  'OtuzGünProgram'

print( meydan . isalnum ()) # True

meydan =  '30GünProgram'

print( meydan . isalnum ()) # True

meydan   =  'otuz gün program'

print ( meydan . isalnum ()) # False, boşluk bir sayısal karakter algılamaz

meydan  =  'otuz gün python 2019'

print( meydan . isalnum ()) # False

* isalpha(): Tüm dize öğelerinin alfabe karakterleri olup olmadığını kontrol eder

meydan   =  'OtuzGünProgram'

print( meydan . isalpha ()) # True

meydan =  '30GünProgram'

print( meydan . isalpha()) # False,30 RAKAMSAL BİR İFADE

meydan   = 'otuz gün program' #False , AYRI YAZILMIŞ

print ( meydan  . isalpha ())

meydan  = 'otuz gün python 2019'

print( meydan . isalpha ()) # FalsE,2019 VE AYRIK YAZILDIĞI İÇİN

* isdecimal(): Bir dizedeki tüm karakterlerin rakam olup olmadığını kontrol eder.

Sdc

meydan   =  'OtuzGünProgram'

print( meydan . isdecimal ()) # False

meydan =  '123'

print( meydan . isdecimal()) # True

meydan   = '30.2'

print ( meydan  . isdecimal ()) #False,2 basamaklı bir sayıdır.

meydan  = 'otuz gün python 2019'

print( meydan . isdecimal()) # False

* isdigit(): Bir dizedeki tüm karakterlerin sayı olup olmadığını kontrol eder.

metin='otuz'

print(metin.isdigit())# False

metin='30'

print(metin.isdigit())# True

metin='pandas34'

print(metin.isdigit())#False

* isnumeric(): Bir dizgedeki tüm karakterlerin sayılarla mı yoksa harflerle mı ilgili olduğunu kontrol eder. Eğer hem sayı hem harf içeriyorsa ½ işareti olur.

numara='10'

print(numara.isnumeric())#True

numara='b021'

print(numara.isnumeric())#True,½

numara='17.5'

print(numara.isnumeric())#False

* isidentifier(): Geçerli bir tanımlayıcı olup olmadığını kontrol eder - bir dizenin geçerli bir değişken adı olup olmadığını kontrol eder

tanım='üç\_gün\_sonra'

print(tanım.isidentifier())#True

tanım='3GünSonrası'

print(tanım.isidentifier())#False

* islower(): Dizedeki tüm alfabe karakterlerinin küçük harf olup olmadığını kontrol eder.

ower='siz gerçek çok zekisiniz'

print(ower.islower())# True

ower='Takipte kalın'

print(ower.islower())# False

* isupper(): Dizedeki tüm alfabe karakterlerinin büyük harf olup olmadığını kontrol eder

ower='siz gerçek çok zekisiniz'

print(ower.isupper())# False

ower='TAKİPTE KALIN'

print(ower.isupper())# True

* join(): Birleştirilmiş bir dize döndürür

dil=['fransızca','almanca', 'rusça']

sonuç=' ' .join(dil)

print(sonuç)

* strip(): Dizenin başından ve sonundan başlayarak verilen tüm karakterleri kaldırır

mesaj='BBugün harika bilgiler öğrendim'

print(mesaj.strip(' '))

* replace(): Alt dizeyi belirli bir dizeyle değiştirir.

mesaj='Ben karpuz isterim'

print(mesaj.replace('karpuz', 'şeftali'))# Ben şeftali isterim

* split(): Ayırıcı olarak verilen dizgiyi veya boşluğu kullanarak dizgiyi böler

mesaj='Ben karpuz isterim'

print(mesaj.split())# ['Ben', 'karpuz', 'isterim']

* title(): Başlık büyük / küçük harfli bir dize döndürür

mesaj='ben karpuz isterim'

print(mesaj.title())#Ben Karpuz Isterim

* swapcase(): Tüm büyük harfleri küçük harfe ve tüm küçük harfleri büyük harfe dönüştürür.

mesaj='ben karpuz isterim'

print(mesaj.swapcase())#BEN KARPUZ ISTERIM

* startwith(): Dizenin Belirtilen Dize ile Başlayıp Başlamadığını kontrol eder.

meydan   =  'OtuzGünProgram'

print( meydan . startswith ('Otuz')) # Otuz ile başladığı için True

meydan =  '30GünProgram'

print( meydan . startswith ('30')) # True

meydan   =  'otuz gün program'

print ( meydan . startswith ('otuz')) # True

meydan  =  'otuz gün python 2019'

print( meydan .startswith ('otuz')) # True