NOVA
it academy

HOŞGELDİNİZ! :)

Tipini Sevdiğimin Verileri Serisi - 1 : String

NOVA it academy

BİR HATIRLATMA

- Sınıfa **ağzı açık kaplarda** içecekle LÜTFEN gelmeyiniz (Çay, kahve…)
  - Tetikli Termos, Kapaklı Su şişesi…
- Ayrılırken;
  + Masa üstlerinde LÜTFEN ÇÖP BIRAKMAYINIZ,
  + Bilgisayarlarınızı LÜTFEN KAPATINIZ,
  + Sandalyelerinizi LÜTFEN DÜZENLİ bırakınız,
- Ders başlangıcı : Max. 5 Dk. Tolerans…

  19:05 Ders başlar!

NOVA it academy

Aşırı Hızlı Tekrar

- Hayırdır len()
- range() in kaç olm,
- Siz iterable'laştıramadıklarımızdan mısınız?
- for döngüsü kimin için dönüyor
- Not değil "not" :)
- Sonsuz döngü her zaman kötü mü?
  - Nasıl kırarız bu döngüyü?
- Sorusu, ilavesi olan?

NOVA it academy

İlk Programlarımız

Problem :

Kullanıcının adı ve soyadı girmesini istedikten sonra, isim ve soyisminin içinde yer alan “a” harfinin sayısını ekrana yazdıran programı for döngüsü kullanarak yazınız.

```
longText = input(“Ad ve Soyad Giriniz : “)
counter = 0

for letter in longText:
    if letter == “a” or letter == “A”:
        counter+=1

print(counter)
```

NOVA
it academy

Yazmadan

Range(2,15,3) fonksiyonunun yaptığı işlemi açıklayınız.

2,5,8,11,13

NOVA
it academy

Yazmadan

```
for i in “Python”:
    print(i, end=“”)
```

Programının çıktısının ne olacağını söyleyiniz.

Python| → imleç aynı satırda kalır

NOVA
it academy

Yazmadan

```
data = [5, 6, 12, 104, 204, 'Python', 'Dersim']

sayac = 0

for i in data:
    if data[i] % 2 == 0
        sayac +=1

print(sayac)
```

Test ekibi tarafından yapılan inceleme sonucunda, programının iki noktada hata vermesi beklenmektedir. Hata vermesi beklenen noktaları belirtiniz ve gerekli düzeltmeleri yapınız.

1) if bloğu “:” ile bitirilmemiş,

2) data verisi, integer ve string türlerinde veri içermektedir. String verisi, mod “%” operatörü ile kullanılamaz.

NOVA it academy

Yazmadan

```
sayac = 0

while sayac > -10:
    print(sayac)

sayac-=1
```

Programının çıktısının ne olacağını söyleyiniz.

```
0
0
0
0
0
0
0
0
0
0
0
0
```

∞

Peki ya bu derste :

- string
- string
- string
- string
- string
- string
- string
- string
- string
- string
- string
- string
- string
- string
- string
- string
- string
- string
- string
- string
- string
- string
- string
- string
- string
- string
- string
- string
- string
- string
- string
- string
- string
- string
- string
- string
- string
- string
- string
- string
- string
- string
- string
- string
- string
- string
- string
- string
- string
- string
- string
- string
- string
- string
- string
- string
- string
- string
- string
- string
- string
- string
- string
- string
- string
- string
- string
- string
- string
- string

NOVA it academy

String Fonksiyonları...

- Fonksiyon, Metod… Aynı şeyi ifade eder… Ben karışık kuruşuk kullanabilirim… Emekliyim… :)
- Metin içeren bir değişken üzerinde çalışıtırılır ve sadece değişkenin çıktısı üzerinde değişiklik yaparlar. Değişkenin kendisinde DEĞİL!!!
- `stringVariable.fonksiyon()`
- `“Bu da böyle bir metin işte”.fonksiyon()`

NOVA it academy

String Fonksiyonları...

```
1  : capitalize
2  : casefold
3  : center
4  : count
5  : encode
6  : endswith
7  : expandtabs
8  : find
9  : format
10 : format_map
11 : index
12 : isalnum
13 : isalpha
14 : isascii
15 : isdecimal
16 : isdigit
17 : isidentifier
18 : islower
19 : isnumeric
20 : isprintable
21 : isspace
22 : istitle
23 : isupper
24 : join
25 : ljust
26 : lower
27 : lstrip
28 : maketrans
29 : partition
30 : removeprefix
31 : removesuffix
32 : replace
33 : rfind
34 : rindex
35 : rjust
36 : rpartition
37 : rsplit
38 : rstrip
39 : split
40 : splitlines
41 : startswith
42 : strip
43 : swapcase
44 : title
45 : translate
46 : upper
47 : zfill
```

```
1 : __add__
2 : __class__
3 : __contains__
4 : __delattr__
5 : __dir__
6 : __doc__
7 : __eq__
8 : __format__
9 : __ge__
10 : __getattribute__
11 : __getitem__
12 : __getnewargs__
13 : __getstate__
14 : __gt__
15 : __hash__
16 : __init__
17 : __init_subclass__
18 : __iter__
19 : __le__
20 : __len__
21 : __lt__
22 : __mod__
23 : __mul__
24 : __ne__
25 : __new__
26 : __reduce__
27 : __reduce_ex__
28 : __repr__
29 : __rmod__
30 : __rmul__
31 : __setattr__
32 : __sizeof__
33 : __str__
34 : __subclasshook__
```

NOVA it academy

String Fonksiyonları...

.capitalize()

Metnin ilk harfini BÜYÜK, geri kalanının da küçük olduğundan emin olur!

```
#1: capitalize()

sampleText = """bu BeNiM DünYam"""
newText = sampleText.capitalize()
print(newText)
print(sampleText.capitalize())
```

OUTPUT : “Bu benim dünyam”

NOVA it academy

String Fonksiyonları...

.casefold()

Alayını küçük harfe çevirir

```
#2: casefold()

sampleText = """bu BeNiM DünYam"""
print(sampleText.casefold())
```

OUTPUT : “bu benim dünyam”

NOVA it academy

String Fonksiyonları...

.center(int, char= “ “)

Şekilli Şüküllü metinler hazırlamamızı sağlar

```
#3: center()

sampleText = "bu BeNim dÜnyAm"
lenOfText = len(sampleText)
spaceBetween = 6 # number/2
newText = sampleText.center(lenOfText+spaceBetween)
newText2 = sampleText.center(lenOfText+spaceBetween, ".")
print(newText, "Sonraki Text")
print(newText2, "Sonraki Text")
```

OUTPUT : “___bu BeNim dÜnyAm___Sonraki Text”

NOVA it academy

String Fonksiyonları...

.count(str, start = 0, end = max)

Parametre olarak verilen stringin metin içindeki tekrar sayısını verir

```
#4: count()

sampleText = "abc_abc_abc_abc_abc_abc_abc"
textToSearch = "ab"
print(sampleText.count(textToSearch ), "Adet eşleşme bulundu")
```

OUTPUT : “7 Adet eşleşme bulundu”

NOVA it academy

String Fonksiyonları...

.encode(encoding = “UTF-8”, errors = “strict”)

Parametre olarak verilen stringin metin içindeki tekrar sayısını verir

```
#5: encode()

sampleText = "Merhaba, ü i ş ğ ç ö!"
encoded_text = sampleText.encode("UTF-8", "strict")
print(encoded_text)
```

OUTPUT : “Merhaba, \xc3\xbc i \xc5\x9f \xc4\x9f \xc3\xa7 \xc3\xb6!”

NOVA it academy

String Fonksiyonları...

.endswith(str, start=0, end = end)

Kontrol etmesi istenen metnin sonunun, parametre olarak verilen string ile bititp bitmediğinin kontrol sonucunu verir

```
#6: endswith()

sampleText = "bu BeNim dÜnyAm"
print(sampleText.endswith("yAm"))
print(sampleText.endswith("yam"))
print(sampleText.endswith(("m","s","e")))
```

OUTPUT :
True
False
True

NOVA it academy

String Fonksiyonları...

.expandtabs(tabsize=8)

Girdiler arası belirtilen “\t” Escape karakterinin farklı uzunluklardaki metinlerde düzgün görünmesini sağlar

```
#7: expandtabs()

sampleText = "Baslik1\tBaslik2\tBaslik3"
sampleText2 = "Uzun Baslik1\tUzun Baslik2\tUzun Baslik3"
print(sampleText)
print(sampleText2)
input("Enter tuşuna basınız...")
print(sampleText.expandtabs(20))
print(sampleText2.expandtabs(20))
```

OUTPUT : “7 Adet eşleşme bulundu”

NOVA it academy

String Fonksiyonları...

.find(str,start,end)

Parametre olarak verilen stringin metin içindeki İLK konumunu verir

```
#8: find()

sampleText = "bu BeNim dÜnyAm"
print("Aranan metin :" , sampleText.find("dÜn"), \
      ". karakterden başlıyor")
pos = sampleText.find("dÜn")
print(pos, ". karakter : ", sampleText[pos])
print(sampleText.find("Deniz"))
```

OUTPUT : "Aranan metin : 9 . karakterden başlıyor
9 . karakter : d"

Eğer herhangi bir eşleşme bulamaz ise : "-1" döndürür.

NOVA
it academy

String Fonksiyonları...

.format()

Degiskenlerin metin içinde yazdırılması için kullanılır.

```
#9: format()

sampleText = "{deg1} benim {deg2}"
print(sampleText.format(deg1 = "Bu", deg2 = "Dünyam"))

deg1 = "Bu"
deg2 = "dünyam"
sampleText = "{} benim {}"
print(sampleText.format(deg1, deg2))

sampleText = "{0} benim {1}"
print(sampleText.format(deg2, deg1))
```

NOVA it academy

String Fonksiyonları...

.format_map(Dict)

Format fonksiyonunun Dictionary adı verilen başka bir veri tipi ile birlikte çalışmasını sağlar

```
#10: format_map()

infoSet = {"Adı" : "Deniz", "Yas":45}
sampleText = "Ogretmen : {Adı},{Yas}"
print(sampleText.format_map(infoSet))
```

OUTPUT : “Ogretmen : Deniz,45”

NOVA it academy

String Fonksiyonları...

.index(int)

find() ile aynı, tek farkı aranan metin bulunamadığında hata verir.

```
#11: index()

sampleText = "Bilim adamları ay yüzeyinde muz buldular."
pos = sampleText.index("muz")
print("Konum : {}".format(pos))
print(sampleText[pos])
```

OUTPUT : “Konum : 28”

NOVA it academy

String Fonksiyonları...

.isalnum()

Metnin AlfaNumerik (Alfabe ve Sayılar) karakterlerden oluşup oluşmadığını belirtir.
! @ # & ( ) – [{ }] : ; ', ? / * ~ $ ^ + = < > 'boşluk'

```
#12: isalnum()

sampleText = "Bilimadamlarıayyüzeyindemuzbuldular"
sampleText2 = "Bilim adamları ay yüzeyinde muz buldular"

print(sampleText.isalnum())
print(sampleText2.isalnum())
```

OUTPUT : "True
False"

NOVA it academy

String Fonksiyonları...

.isalpha()

Alfabe elementlerinden oluşup oluşmadığını verir.

```
#13: isalpha()

sampleText = "Bilim adamları ay yüzeyinde üç adet muz buldular"
sampleText2 = "Bilimadamlarıayyüzeyindeüçadetmuzbuldular"

print(sampleText.isalpha())
print(sampleText2.isalpha())
```

OUTPUT : “False
True”

NOVA it academy

String Fonksiyonları...

.isascii()

Metin değişkeni içindeki verinin ascii tablosundaki verileri içerip içermediğini kontrol eder.

```
#14: isascii()

sampleText = "Made in Turkiye"
sampleText2 = "Made in Türkiye"

print(sampleText.isascii())
print(sampleText2.isascii())
```

OUTPUT : “True
False”

NOVA it academy

String Fonksiyonları...

.isdecimal()
.isdigit()
.isnumeric()

Parametre olarak verilen stringin metin içindeki tekrar sayısını verir

```
#15: isdecimal()
#16: isdigit()
#17: isnumeric()

sampleText = "177"          #Bir metinde decimal sayı varsa, hem digit hem de numeric dir
sampleText2 = "⓪①②③④⑤"    #Eger digitse, numrecdir de
sampleText3 = "壱弐参"       #Japonca ama sayı...

print(sampleText.isdecimal())
print(sampleText2.isdigit())
print(sampleText.isnumeric())
```

OUTPUT : "True
True
True"

NOVA it academy

String Fonksiyonları...

.isidentifier()

Metnin değişken ismi olarak kullanılıp kullanamayacağını verir.

```
#18: isidentifier()

var1 = "myVariable"
var2 = "_myVariable"
var3 = "1Variable"
var4 = "def"

print(var1.isidentifier())
print(var2.isidentifier())
print(var3.isidentifier())
print(var4.isidentifier())
```

NOVA it academy

String Fonksiyonları...

.islower()

Metnin küçük harflerden oluşup oluşmadığını kontrol eder.

```
#19: islower()

sampleText = "myVariable"
sampleText2 = "_myvariable"
sampleText2 = "myvariable"

print(sampleText.islower())
print(sampleText2.islower())
print(sampleText2.islower())
```

NOVA it academy

String Fonksiyonları...

.isprintable()

Ekrana yazdırılabilir kararter içerip içermediğini belirtir.

```
#20: isprintable()

sampleText = "myVariable\n"
sampleText2 = "_myvariable\t"
sampleText2 = "myvariable"

print(sampleText.isprintable())
print(sampleText2.isprintable())
print(sampleText2.isprintable())
```

NOVA it academy

String Fonksiyonları...

.isspace()

Metnin sadece boşluk karakterinden oluşup oluşmadığını kontrol eder.

```
#21: isspace()

sampleText = "     "
sampleText2 = "    e"

print(sampleText.isspace())
print(sampleText2.isspace())
```

OUTPUT : "True
False"

NOVA it academy

String Fonksiyonları...

.istitle()

Metnin her bir kelimesinin ilk harfinin büyük olup olmadığını kontrol eder.

```
#22: istitle()

sampleText = "merhaba Dünya"
sampleText2 = "Merhaba Dünya"

print(sampleText.istitle())
print(sampleText2.istitle())
```

OUTPUT : “7 Adet eşleşme bulundu”

NOVA it academy

String Fonksiyonları...

.isupper()

Metnin BÜYÜK harflerden oluşup oluşmadığını kontrol eder.

```
#23: isupper()

sampleText = "merhaba Dünya"
sampleText2 = "MERHABA DÜNYA"

print(sampleText.istitle())
print(sampleText2.istitle())
```

OUTPUT : “7 Adet eşleşme bulundu”

NOVA it academy

String Fonksiyonları...

.join()

Parametre olarak verilen stringlerin birleştirilmesi sırasında kullanılacak olan metnin, kelimelerin arasına yerleştirilmesini sağlar.

```
#24: join()

sampleText = "-"

t1="Kelime 1"
t2="Kelime 2"
t3="Kelime 3"

print(sampleText.join([t1,t2,t3]))
```

NOVA it academy

String Fonksiyonları...

.ljust()
.rjust()

Metnin belirtilen genişlik dahilinde, ve belirtilen karakter boşukları dolduracak şekilde sağa ya da sola hizalanmasını sağlar.

```
#25: ljust()
#26: rjust()

sampleText = "Merhaba Dünya"
print(sampleText.ljust(30, "_"))
print(sampleText.rjust(30, "_"))
```

OUTPUT : "Merhaba Dünya__________
__________Merhaba Dünya"

NOVA it academy

String Fonksiyonları...

.lower()
.upper()

Metnin tamamını küçük/BÜYÜK harflere çevirir.

```
#27: lower()
#28: upper()

sampleText = "Merhaba Dünya"
print(sampleText.lower())
print(sampleText.upper())
```

OUTPUT : “7 Adet eşleşme bulundu”

NOVA it academy

String Fonksiyonları...

.lstrip()
.rstrip()

Parametre olarak verilen stringin metin içindeki tekrar sayısını verir

```
#29: lstrip()
#30: rstrip()

sampleText = "Merhaba Dünya"
print(sampleText.lstrip("Merhaba"))
print(sampleText.rstrip("Dünya"))
```

OUTPUT : “7 Adet eşleşme bulundu”

NOVA it academy

String Fonksiyonları...

.maketrans()
.translate()

Metin içindeki özel karakterlerin belirtilen karakterler ile değiştirilmesini sağlayan tablodur.

```
#31: maketrans()
#32: translate()

sampleText = "Merhaba Dünya"
table = sampleText.maketrans("eby", "👋🌆🎓")
print(table)
print(sampleText.translate(table))
```

OUTPUT : “7 Adet eşleşme bulundu”

NOVA it academy

String Fonksiyonları...

.partition(str)
.rpartition(str)
.split(str, int)
.rsplit(str, int)

Metnin bölünmesinde kullanılır. split fonksiyonları için adım sayısı belirtilebilir.

```
#33: partition()
#34: rpartition()
#35: split()
#36: rsplit()



sampleText = "Merhaba_Bu Dünya_Benim"
print(sampleText.partition("_"))
print(sampleText.rpartition("_"))
print(sampleText.split("_",1))
print(sampleText.rsplit("_",1))
```

NOVA it academy

String Fonksiyonları...

.removeprefix(str)
.removesuffix(str)

Parametre olarak verilen değerleri metinden kaldırır.

```
#37: removeprefix(str)
#38: removesuffix(str)

sampleText = "Merhaba_Dünya"
print(sampleText.removeprefix("Me"))
print(sampleText.removesuffix("ya"))
```

OUTPUT : “7 Adet eşleşme bulundu”

NOVA it academy

String Fonksiyonları...

.replace(str,str,int)
.rfind(str)
.rindex(str)

Metinleri değiştirir
Bulma işlemini sağ taraftan başlatır

```
#39: replace()
#40: rfind()
#41: rindex()

sampleText = "Merhaba_Bu Dünya_Benim"
print(sampleText.replace("Bu", "0",  1))
print(sampleText.rfind("u"))
print(sampleText.rindex("e"))
```

OUTPUT : “7 Adet eşleşme bulundu”