

3-mashg'ulot. Satrlar bilan ishlovchi operatorlar va metodlar.

O'quv savollari:

1. Satrlar bilan ishlovchi operatorlar va metodlar.
2. str.format() metodi yordamida satrlarni formatlash

Satrlar bilan ishlovchi operatorlar va metodlar

Satr deb mavjud belgilar ketma-ketligiga aytiladi. Satrlarni pythonda aniqlash uchun ham bittalik, ham qo'shtirnoqlardan foydalaniladi

```
msg = 'Salom Dunyo!'
print(msg)    # Salom Dunyo!
```

```
nomi="Jeyson"
print(nomi)   # Jeyson
```

```
text = ("Salom hurmatli kursantlar"
        "Bugun biz sizlar bilan"
        "Satrlar ustida amallar bajarib ko'ramiz")
print(text)
```

```
'''
Bu izoh
'''
text = '''Salom hurmatli kursantlar
Bugun biz sizlar bilan
Satrlar ustida amallar
bajarib ko'ramiz
'''
print(text)
```

Satrdagi qo'llaniladigan maxsus simvollar

- `\` : satr tarkibiga slesh symbolini qo'shish imkonini beradi
- `\'` : satr ichiga bitta tirnoq qo'shish imkonini beradi
- `\"` : qatorga qo'sh tirnoq qo'shish imkonini beradi
- `\n` : Yangi qatorga o'tadi
- `\t` : Tabulyatsiya qo'shadi (4 ta probelga teng bo'sh joy)

```
text = "Xabar:\n\"Salom Dunyo\""
print(text)
```

```
Xabar:
"Salom Dunyo"
```

URL ni o'z xolatida satr sifatida foydalanish

```
URL = "D:\python_course\name.txt"  
print(URL)
```

```
c:\python_course  
ame.txt
```

```
URL = r"C:\python_course\name.txt"  
print(URL)
```

```
c:\python_course\name.txt
```

Qatorga qiymatlarni kiritish

Python boshqa o'zgaruvchilarning qiymatlarini satrga joylashtirish imkonini beradi. Buning uchun satr ichida o'zgaruvchilar jingalak qavslarga {} joylashtiriladi va satr oldidagi qo'shtirnoq belgisi oldiga f harfi qo'yiladi

```
user_name = "Jeyson"
user_age = 22
user = f"Ismi: { user_name }   Yoshi: { user_age }"

print(user_age)      # Ismi: Jeyson   Yoshi: 22
```

Satr tarkibidagi belgilarga murojaat qilish

Musbat indeks

```
str = "hello world!"  
belgi = str[0]    # h  
print(belgi)  
  
belgi = str[6]    # w  
print(belgi)
```

Manfiy indeks


```
str = "salom dunyo!"  
belgi = str[-1]  
print(belgi)      #!  
  
belgi2 = str[-5]  
print(belgi2)     #u
```

Matndan belgilar ketma-ketligini olish

str[:end] – ushbu ifoda 0 - indeksdan to end - indeksgacha belgilar ketma-ketligini chiqaradi. end – elementni o‘zini olmaydi. End o‘rnida son beriladi. str[:10]

str[start:end] – belgilar ketma-ketligi **start** dan to **end** - indeksigacha bo‘lgan belgilar ketma-ketligini qaytaradi. Ammo ikkinchi index – end xisobga olinmaydi.

string[start:end:step] – belgilar ketma-ketligini **start** dan to **end** - indeksigacha bo‘lgan belgilar ketma-ketligini **step** – qadami bilan qaytaradi



```
str = "salom dunyo"
```

```
# 0-indeksdan 5 - indeksgacha  
sub_str1 = str[:5]  
print(sub_str1)           # salom
```

```
# 2-indeksdan 5-indeksgacha  
sub_str2 = str[2:5]  
print(sub_str2)           #llo
```

```
# 2 - indeksdan 9-indeksgacha oraliqda 2 qadam bilan  
sub_str3 = str[2:9:2]  
print(sub_str3)           # lmdn
```


Satrlarni birlashtirish

```
name = "Jeyson"
surname = "Stethom"

full_name = name + " " + surname

print(full_name)  # Jeyson Stethom
```

```
name=" Jeyson "
age = 38

info_per = "Ism: " + name + " Yoshi: " +
str(age)

print(info_per)  # Ism: Jeyson Yoshi: 38
```

Satrnı takrorlash

Satrlarga arifmetik ko‘paytirish amalini qo‘llash, ularning shuncha marta takrorlaydi

```
print("char" * 3)    # charcharchar  
print("u" * 4)       # uuuu
```

Satrlarni taqqoslash

Satrlarni taqqoslashda asosiy diqqatni belgiga va ularning katta yoki kichik harfligiga qaratiladi. Demak, raqamli belgi shartli ravishda har qanday alifbo belgisidan kichik hisoblanadi. Katta harfdagi alifbo belgilari shartli ravishda kichik alifbo belgilaridan kichik hisoblanadi.

```
str1 = "1a"  
str2 = "aa"  
str3 = "aa"  
  
print(str1>str2)  #False  
print(str2>str3)  #True
```

Satr bilan ishlovchi operatorlar va metodlardan foydalanish

isalpha(): agar satr faqat alfavit belgilaridan iborat bo'lsa, True qaytaradi;

islower(): Agar satr faqat kichik harflardan iborat bo'lsa, True qiymatini qaytaradi;

isupper(): Agar satrdagi barcha belgilar katta harflardan iborat bo'lsa, True qiymatini qaytaradi;

isdigit(): agar satrning barcha belgilari raqam bo'lsa, True qiymatini qaytaradi;

isnumeric(): agar satr raqam bo'lsa, True qiymatini qaytaradi;

startswith(str): agar satr pastki qator str bilan boshlansa, True qiymatini qaytaradi;



endswith(str): agar satr oxiri **str** bilan tugasa, True qiymatini qaytaradi;

low(): satrni kichik harfga o'zgartiradi;

upper(): satrni katta harfga aylantirish;

title(): Satrdagi barcha so'zlarning birinchi harflarini bosh harfga o'zgartiradi;

capitalize(): satrning faqat birinchi so'zining birinchi harfini bosh harf bilan yozadi;

lstrip(): satr boshidagi probellarni o'chirib, satrni tozalaydi;

rstrip(): satr oxiridagi probellarni o'chirib, satrni tozalaydi;

strip(): satrdan oldingi va keyingi bo'shliqlarni olib tashlaydi;

ljust(width): agar satr uzunligi width parametridan kichik bo'lsa, kenglik qiymatini to'ldirish uchun satrning o'ng tomoniga bo'shliqlar qo'shiladi va satrning o'zi oqlanadi;

rjust(width): agar satr uzunligi width parametridan kichik bo'lsa, kenglik qiymatini to'ldirish uchun satrning chap tomoniga bo'shliqlar qo'shiladi va satrning o'zi o'ng tomonga asoslanadi;


center(width): agar satrning uzunligi kenglik parametridan kichik bo'lsa, kenglik qiymatini to'ldirish uchun satrning chap va o'ng tomoniga bo'shliqlar teng ravishda qo'shiladi va satrning o'zi markazlashtiriladi;

find(str[, start [, end]]): satrning ko'rsatilgan intervaldan qism qatorini qidirish. Agar satrdan qism qator topilsa indeksini qaytaradi. Agar qator topilmasa, -1 raqami qaytariladi;

replace(old, new[, num]): satrdagi bir qism qatorni boshqasiga almashtiradi;

split([delimiter[, num]]): ajratuvchiga qarab qatorni qism qatorlarga ajratadi;

join(strs): bir nechta satrni ular orasiga ma'lum ajratuvchi qo'yib, bir qatorga birlashtiradi.



```
str = input(" Raqamni kiriting :")  
if str.isnumeric() :  
    raqam = int(str)  
    print(raqam)
```

```
f = "hello.py"  
startsWithHello = f.startswith("hello") #True  
endsWithExe = f.endswith("exe") # False
```

```
str = "    Salom    dunyo!    "  
str = str.strip()  
print(str)                # salom    dunyo!
```



```
print("Redmi 7:", "500".rjust(10))  
print("Nokia P10:", "360".rjust(10))
```

```
Hi_bye = "Salom dunyo! Xayr dunyo!"  
indeks = Hi_bye.find("dun")  
print(indeks)          #6  
  
# 10 indeksdan qidirish  
indeks = Hi_bye.find("dun",10)  
print(indeks)          #21  
  
# 10 dan 15 gacha indekslarni qidirish  
indeks = Hi_bye.find("dun",10,15)  
print(indeks)          # -1 - topilmaganini anglatadi
```


Satrdagi almashtirish (*replace*).

replace (old, new): satrdagi **old** satrlarni **new** satr bilan almashtiradi;

replace(old, new, num): **num** parametri **old** satrlarning nechtaisini **new** satri bilan almashtirishni belgilab beradi. Odatda **num** parametridan foydalanilmasa, u -1 qiymatni qabul qiladi, bu esa barcha **old** satrlarni **new** satri bilan almashtirilishini anglatadi

```
tel_num = "+99-899-667-89-10"

# defislarni bo'sh joylar bilan almashtirish
tahrirlangan_telefon = tel_num.replace("-", " ")
print(tahrirlangan_telefon)           # +99 899 667 89 10
```



```
# defislarni olib tashlash
```

```
tahrirlangan_telefon = tel_num.replace("-", "")
```

```
print(tahrirlangan_telefon)           # +998996678910
```

```
# faqat birinchi defisni almashtirish
```

```
tahrirlangan_telefon = tel_num.replace("-", "", 1)
```

```
print(tahrirlangan_telefon)           # +99899-667-89-10
```


Qism qatorlarga ajratish

split(): ajratuvchi sifatida bo'sh joy qilinadi. Va probeldan probelgacha bo'lgan joylarni element qilib oladi.

split(delimeter): cheklovchi sifatida delimiter dan foydalaniladi.

split(delimeter, num): num parametri bo'linish uchun delimiter (ajratuvchi) ning necha marta ishlatilishini belgilaydi

```
text = "Bu katta, ikki bo'yli eman, shoxlari singan va po'stlog'i  
singan"  
# probellardan ajratiladi  
  
splitted_text = text.split()  
print(splitted_text)  
print(splitted_text[6])    #singan
```



```
# vergul ustida tanaffus
splitted_text = text.split(",")
print(splitted_text)
print(splitted_text[1])    # ikki bo'yli eman
```

```
# birinchi besh bo'shliqqa bo'lingan
splitted_text = text.split(" ", 5)

print(bo'lingan_matn)

print(bo'lingan_matn[5])

# shoxlari singan va po'stlog'i singan
```

Satrlarni birlashtirish

Satrlar ustidagi eng oddiy amallarni ko‘rib chiqqan turib, qo‘shish amali yordamida qatorlarni birlashtirishni ko‘rsatdik. Satrlarni birlashtirishning yana bir imkoniyati - ***join()*** ***usuli***: u bir nechta satrlarni birlashtirish uchun ishlatiladi

```
strings = ["Let's", "go", "speaking", "from", "my", "heart", "into", " My box"]
sentence = " ".join(strings)
print(sentence)

# ajratuvchi - vertikal chiziq
sentence = " | ".join(strings)
print(sentence)

# Let's | go | speaking | from | my | heart | into | My box
```

Satrlarni birlashtirish

Ro'yxat o'rniga oddiy satr qo'shilish usuliga o'tkazilishi mumkin, keyin ajratuvchi ushbu satr belgilari orasiga kiritiladi

```
str = "hello"  
joined_str = "\".join(str)  
  
print(joined_str)          # h\e\l\l\o
```

str.format() metodi yordamida satrlarni formatlash

```
first_name="Jeyson"
text = f"Hello, {first_name}."
print(text)          # Hello, Jeyson.

name="Bobo"
age=73
info = f"Ismi: {name}\t Yoshi: {age}"
print(info)         # Ismi: Bobo  Yoshi: 73
```

Nomlangan parametrlar

```
txt = "Salom, {first_name}.".format(first_name="Jeyson")  
print(txt)      # Salom, Jeyson.
```

```
info = "Ismi: {name}\t Yoshi: {age}".format(name="Bobo", age=73)  
print(info)     # Ismi: Bobo Yoshi: 73
```


Joylashgan o'рни bo'yicha parametrlar

```
info = "Ismi: {0}\t Age: {1}".format("Bobo", 73)  
print(info)          # Ismi: Bobo  Age: 73
```

Bunday holda, argumentlar satrga bir necha marta chaqirilishi ham mumkin:

```
txt = "Salom, {0} {0} {0}.".format("Jeyson")
```

```
Salom, Jeyson Jeyson Jeyson
```

Almashtirishlar

s: String tipli ma'lumot kiritish uchun;

d: Double tipli sonlarni kiritish uchun;

f: Float tipli sonlarni kiritish uchun. Bu tur uchun nuqta orqali kasr sonini aniqlash ham mumkin;

%: qiymat 100 ga ko'paytiriladi va foiz belgisini qo'shib qo'yiladi.

```
hi = "Salom sizga {:s}"  
name = "Jeyson"  
formatted_txt = hi.format(name)  
print(formatted_txt)           # Salom sizga Jeyson
```



```
n = 300
```

```
txt = f"{n:,d} belgi"
```

```
print(txt)    # 200 belgi
```


```
num = 115.2548695
```

```
print("{:.2f}".format(num))    # 115.25
```

```
print("{:.3f}".format(num))    # 115.254
```

```
print("{:.4f}".format(num))    # 115.2548
```

```
print("{:,.2f}".format(12582.23664))    # 12,582.23
```



```
print("{:10.2f}".format(115.2548695))    #115.25
print("{:8d}".format(25))                # 25
```

```
n1 = 115.2548695
print(f"{n1:10.2f}")    #115.25
n2 = 26
print(f"{n2:8d}")       #26
```

```
num = .123456
print("{:%}".format(num))    # 12.3456000%
print("{:.0%}".format(num))  # 12%
print("{:.1%}".format(num))  # 12.3%

print(f"{num:%}")            # 12.3456000%
print(f"{num:.0%}")           # 12%
print(f"{num:.1%}")           # 12.3%
```

Format metodini ishlatmasdan formatlash

```
string%(param1, param2,..paramN)
```

```
info = " Ism : %s\t Yoshi:%d"%( "Jeck", 32)  
print(info)      # Ism : Jeck      Yoshi : 32
```

```
raqam = 23.8689578  
print("%0.2f    - %e" % (raqam, raqam))  
# 23.87    - 2.386896 e +01
```



Thank you.

See you next lesson

