

Python programming language

Week 3. Python string

Alisher Ismailov

Lecturer

Department of Information technology
Andijan Branch of Tashkent Financial Institute

Email: alisherismailov534@gmail.com

Python dasturlash tili

3-Mavzu. Python String

















Ma'ruzachi: Alisher Ismoilov

Axborot texnologiyalari kafedrası
Toshkent moliya instituti Andijon filiali

Elektron pochta: alisherismailov534@gmail.com

3-Mavzu. Python string

Reja:

-  String o'zgaruvchi yaratish
-  String berilganlar turi Array hisoblanadi
-  String uzunligi aniqlash
-  Stringni tekshirish
-  String o'zgaruvchilarni kesish
-  String boshidan belgilangan indeks gacha chop etish
-  Stringni belgilangan indeksdan string oxirigacha chop etish
-  Manfiy qiymat bilan indeksiyalash
-  String o'zgaruvchilarni berilganlarini o'zgartirish
-  String o'zgaruvchi berilganlari orasidagi bo'sh joyni o'chirib tashlash
-  String o'zgaruvchini berilganlarini almashtirish
-  String o'zgaruvchisini ajratish
-  String o'zgaruvchilarini birlashtirish
-  Stringni formatlash
-  String o'zgaruvchiga belgi qo'shish
-  String funksiyalari

Python dasturlash tilida String

Python dasturlash tilidagi **String** o'zgaruvchi yaratish uchun qo'shtirnoq "" yoki bittalik tirnoq " belgisi orqali yaratiladi.

String o'zgaruvchi yaratish

- String o'zgaruvchi yaratish uchun o'zgaruvchi nomidan keyin teng(=) belgisi va "" qo'shtirnoq belgisi bilan yaratiladi.

```
In [136]: matn = "Salom komputer lingivistikasi"
```

```
print(matn)
```

```
Salom komputer lingivistikasi
```

Python dasturlash tilida String

Ko'p qatorli **String** o'zgaruvchisini yaratish

O'zgaruvchiga ko'p qatorli String yaratish
uchun uchtalik qo'shtirnoq

```
"""
```

```
matn
```

```
"""
```

String o'zgaruvchi yaratish

```
In [17]: a = """String ma'lumot turini
ko`p qatorli o`zgaruvchi ma'lumoti
yaratish."""
print(a)
```

```
String ma'lumot turini
ko`p qatorli o`zgaruvchi ma'lumoti
yaratish.
```

String o'zgaruvchi yaratish

- Yoki “” **matn** “” belgisi orqali yaratish mumkin:

```
In [17]: a = """String ma'lumot turini
ko`p qatorli o`zgaruvchi ma'lumoti
yaratish."""
print(a)
```

```
String ma'lumot turini
ko`p qatorli o`zgaruvchi ma'lumoti
yaratish.
```

String berilganlar turi Array(massiv) hisoblanadi

- Boshqa ko‘plab mashxur dasturlash tillari singari, Python dasturlash tilidagi **String** o‘zgaruvchilar unicode belgilarni ifodalovchi **bayt Array** hisoblanadi. [] qavslar array elementlariga kirish uchun ishlatilishi mumkin

```
In [137]: matn = "Python Dasturlash Tili"  
  
print(matn[0])
```

P

String uzunligi aniqlash

- String uzunligini aniqlash uchun **len()** funksiyasidan foydalaniladi.

```
In [138]: matn = "Python Dasturlash Tili"  
  
print(len(matn))
```

22

Stringni tekshirish

- String o'zgaruvchida ma'lum bir ibora yoki belgi mavjudligini tekshirish uchun **in** kalit so'zidan foydalanish mumkin

```
In [23]: txt = "Yashash yaxshi, yaxshi yashash undanda yaxshi!"  
print("undanda" in txt)  
  
True
```

String o'zgaruvchilarni kesish. (ma'lum qismini print qilish va ma'lum qismini ishlatmaslik)

- 0-pozitsiyadan 15-pozitsiyaga belgilarni ekranga chop etish

```
In [140]: matn = "Dastur kodini yozish qiziqarli"  
  
print(matn[0:15])  
  
Dastur kodini y
```

String boshidan belgilangan indeksgacha qiymatlarni ekranga chop etish

- Boshlanish indeksini bo'sh qoldirsangiz, diapazon birinchi elementdan boshlanadi

```
In [142]: matn = "Men sport bilan shug`ullanaman"  
  
print(matn[:10])
```

```
Men sport
```

String boshidan belgilangan indeksgacha qiymatlarni ekranga chop etish

- Boshlanish indeksini bo'sh qoldirsangiz, diapazon birinchi elementdan boshlanadi

```
In [142]: matn = "Men sport bilan shug`ullanaman"  
  
print(matn[:10])  
  
Men sport
```

- Yuqoridagi misolda, boshlanish indeksini bermaganimiz uchun **stringni** boshidan boshlandi.

Stringni belgilangan indeksdan string oxirigacha chop etish

- Belgilarni 2-pozitsiyadan oxirigacha ekranga chop etish

```
In [26]: b = "Salom Python!"  
print(b[2:])
```

```
lom Python!
```

Manfiy qiymat bilan indekslash

- Qismni **String** oxiridan boshlash uchun manfiy indekslardan foydalanish mumkin:

```
In [32]: b = "Salom Python!"  
print(b[-7:-1])
```

```
Python
```

String o'zgaruvchilarni qiymatini o'zgartirish

- Python dasturlash tilida **String** qiymatini o'zgartirish uchun tayyor yaratilgan funskiyalar mavjud. Bularni bir qanchasini birga ko'rib chiqamiz. **upper()** funksiyasi String o'zgaruvchi qiymatini katta harf holatiga keltiradi:

```
matn = "kitob o`qish inson taffakkurini oshiradi"  
  
print(matn.upper())
```

```
KITOB O`QISH INSON TAFFAKKURINI OSHIRADI
```

- Yuqoridagi misol **upper()** funksiyasi **b** o'zgaruvchisining qiymatini hamma harflarini katta harfga o'zgartirdi.

String o'zgaruvchilarni qiymatini o'zgartirish

- **lower()** funksiyasi String o'zgaruvchi qiymatini katta harfdan kichik harfga o'gartiradi:

```
In [145]: matn = "BUGUN QOR YOG`DI"  
  
print(matn.lower())  
  
bugun qor yog`di
```

String o'zgaruvchi qiymati orasidagi bo'sh joyni o'chirib tashlash

- Bo'sh joy haqiqiy matndan oldin yoki keyin bo'sh joy bo'lib, ko'pincha bu bo'shliqni olib tashlash kerak bo'ladi. **strip()** funksiyasi **String** boshidagi yoki oxiridagi har qanday bo'shliqni o'chirib tashlaydi

```
matn = " Bo`sh joyni o`chirish uchun strip() funksiyasidan foydalaning"  
print(matn.strip())
```

```
Bo`sh joyni o`chirish uchun strip() funksiyasidan foydalaning
```

String o'zgaruvchini qiymatini almashtirish

- `replace()` funksiya String qiymatini boshqa matnga almashtiradi:

```
matn = "Malayziya Terengganu Universiteti"
```

```
print(matn.replace("Terengganu", "Putra"))
```

```
Malayziya Putra Universiteti
```

String o'zgaruvchisini ajratish

- Split() funksiya, agar ajratuvchining belgisini topsa, stringni pastki qatorlarga ajratadi:

```
matn = "bog`cha, maktab, litsey, universitet"
```

```
print(matn.split(","))
```

```
['bog`cha', ' maktab', ' litsey', ' universitet']
```

String o'zgaruvchilarini birlashtirish

- Ikkita o'zgaruvchini qiymatini birlashtirish uchun + operatoridan foydalanish mumkin.

```
In [151]: matn_bir = "stol"
          matn_ikki = "stul"

          print(matn_bir+matn_ikki)

          stolstul
```

String o'zgaruvchilarini birlashtirish

- Ularning orasiga bo'sh joy qo'shish uchun " " qo'shing:

```
In [152]: matn_bir = "stol"  
          matn_ikki = "stul"  
  
          print(matn_bir+" "+matn_ikki)  
  
          stol stul
```

Stringni formatlash

- Python dasturlash tilida o'zgaruvchilar String va Raqam berilganlar turi bo'lganda ularni bir-biriga qo'sha olmaymiz. Bu muammoni yechish uchun **format()** funksiyasi yordamida String va raqamlarni birlashtirish mumkin. **format()** funksiyasi o'tkazilgan argumentlarni oladi, ularni formatlaydi va ularni {} to'ldiruvchilari bo'lgan stringga joylashtiradi:

```
In [49]: yosh = 30
         txt = "Meni ismim ALisher, meni yoshim {}"
         print(txt.format(yosh))
```

```
Meni ismim ALisher, meni yoshim 30
```

String o'zgaruvchiga belgi qo'shish

- String o'zgaruvchiga belgilarni kiritish uchun maxsus \ belgisidan foydalaning. Masalan joy nomini qo'shtirnoq ichiga olmoqchi bo'lsangiz, quyidagidek bajarsak bo'ladi

```
In [60]: manzil = "Men \"Andijonda\" yashayman"  
print(manzil)
```

```
Men "Andijonda" yashayman
```

capitalize() funksiyasiga misol

```
In [8]: matn = "salom python"  
  
x = matn.capitalize()  
  
print (x)
```

```
Salom python
```

casefold() funksiyasiga misol

```
In [12]: matn = "SALOM STRING"  
  
x = matn.casefold()  
  
print (x)  
  
salom string
```

center() funksiyasiga misol

```
In [15]: matn = "O`rtaga surish"  
  
x = matn.center(50)  
  
print (x)
```

O`rtaga surish

count() funksiyasiga misol

```
matn = "Bobur, Ulug`bek, Yodgorbek, Alisher, Bobur"  
  
x = matn.count("Bobur")  
  
print (x)
```

2

encode() funksiyasiga misol

```
In [19]: matn = "Mening ismim Alisher"  
  
x = matn.encode()  
  
print (x)  
  
b'Mening ismim Alisher'
```

endswith() funksiyasiga misol

```
matn = "Gapni oxiri qanday belgi bilan tugaganini tekshirish."  
  
x = matn.endswith(".")  
  
print (x)
```

True

expandtabs() funksiyasiga misol

```
In [24]: matn = "T\ta\tb \tq\to`\tsh\ti\tsh"  
  
x = matn.expandtabs(2)  
  
print (x)
```

```
T a b   q o`  sh  i sh
```

find() funksiyasiga misol

```
In [25]: matn = "Python dasturlash tilini o`rganayapman"  
  
x = matn.find("tilini")  
  
print (x)
```

18

format() funksiyasiga misol

```
In [27]: matn = "Python dasturlash tili kitobi {narxi:.2f}"  
  
x = matn.format(narxi = 50)  
  
print (x)
```

```
Python dasturlash tili kitobi 50.00
```

index() funksiyasiga misol

```
In [29]: matn = "Python dasturlash tili kitobi"  
  
x = matn.index("kitobi")  
  
print (x)
```

23

isalnum() funksiyasiga misol

```
In [34]: matn = "Python2022"  
  
x = matn.isalnum()  
  
print(x)
```

```
True
```

isalpha() funksiyasiga misol

```
In [39]: matn = "Python"  
  
x = matn.isalpha()  
  
print(x)
```

```
True
```

isdecimal() funksiyasiga misol

```
In [41]: matn = "\u0033"  
  
x = matn.isdecimal()  
  
print(x)
```

True

isdigit() funksiyasiga misol

```
In [43]: matn = "2020"  
  
x = matn.isdigit()  
  
print(x)
```

```
True
```

isidentifier () funksiyasiga misol

```
In [45]: matn = "Salom"  
  
x = matn.isidentifier()  
  
print(x)
```

```
True
```

islower() funksiyasiga misol

```
In [47]: matn = "salom python"
```

```
x = matn.islower()
```

```
print(x)
```

```
True
```

isnumeric() funksiyasiga misol

```
In [48]: matn = "1234567"  
  
x = matn.isnumeric()  
  
print(x)
```

```
True
```

isprintable() funksiyasiga misol

```
In [51]: matn = "1234567#python"  
  
x = matn.isprintable()  
  
print(x)
```

```
True
```

isspace() funksiyasiga misol

```
In [54]: matn = " "  
  
x = matn.isspace()  
  
print(x)
```

```
True
```

isupper() funksiyasiga misol

```
In [60]: matn = "PYTHON DASTURLASH TILI"  
  
x = matn.isupper()  
  
print(x)
```

```
True
```

join() funksiyasiga misol

```
In [61]: matn = ("PYTHON", "JAVA", "C++")  
  
x = "#".join(matn)  
  
print(x)
```

```
PYTHON#JAVA#C++
```

ljust() funksiyasiga misol

```
In [65]: matn = "PYTHON"
```

```
x = matn.ljust(20)
```

```
print(x, "dasturlash tili")
```

```
PYTHON
```

```
dasturlash tili
```

lower() funksiyasiga misol

```
In [67]: matn = "PYTHON"  
x = x.lower()  
  
print(x)
```

```
python
```

partition() funksiyasiga misol

```
In [74]: matn = "SALOM PYTHON DASTURLASH"

x = matn.partition("PYTHON")

print(x)

('SALOM ', 'PYTHON', ' DASTURLASH')
```

replace() funksiyasiga misol

```
In [76]: matn = "SALOM PYTHON DASTURLASH"  
  
x =matn.replace("PYTHON", "JAVA")  
  
print(x)
```

```
SALOM JAVA DASTURLASH
```

rfind() funksiyasiga misol

```
In [77]: matn = "SALOM PYTHON DASTURLASH"  
  
x =matn.rfind("PYTHON")  
  
print(x)
```

rindex() funksiyasiga misol

```
In [79]: matn = "SALOM PYTHON DASTURLASH"  
  
x =matn.rindex("DASTURLASH")  
  
print(x)
```

13

rpartition() funksiyasiga misol

```
In [81]: matn = "SALOM PYTHON DASTURLASH"

x =matn.rpartition("PYTHON")

print(x)

('SALOM ', 'PYTHON', ' DASTURLASH')
```

rsplit() funksiyasiga misol

```
In [82]: matn = "PYTHON, JAVA, C++"  
  
x =matn.rsplit(", ")  
  
print(x)
```

```
['PYTHON', 'JAVA', 'C++']
```

split() funksiyasiga misol

```
In [84]: matn = "PYTHON Dasturlash tili"  
  
x =matn.split()  
  
print(x)
```

```
['PYTHON', 'Dasturlash', 'tili']
```

splitlines() funksiyasiga misol

```
In [86]: matn = "PYTHON Dasturlash tili\nva Java"

x =matn.splitlines()

print(x)

['PYTHON Dasturlash tili', 'va Java']
```

startswith() funksiyasiga misol

```
In [88]: matn = "PYTHON Dasturlash tili\nva Java"  
  
x =matn.startswith("PYTHON")  
  
print(x)
```

True

swapcase() funksiyasiga misol

```
In [91]: matn = "PYTHON Dasturlash Tili"
```

```
x =matn.swapcase()
```

```
print(x)
```

```
python dASTURLASH tILI
```

Foydalanilgan adabiyotlar

1. Mastering Object-Oriented Python: Build powerful applications with reusable code using OOP design patterns and Python 3.7, 2nd Edition, Steven F. Lott, Packt Publishing (June 14, 2019)
2. Learning Python, 5th Edition Fifth Edition, Mark Lutz , O'Reilly Media, June 12, 2013
3. Python Programming for Beginners: The Ultimate Guide for Beginners to Learn Python Programming: Crash Course on Python Programming for Beginners, AMZ Publishing, Independently published (July 13, 2021)
4. <https://www.python.org/>
5. <https://www.w3schools.com/>
6. <https://www.codecademy.com/catalog/language/python>
7. <https://realpython.com/>
8. <https://www.anaconda.com/>

**E'tiboringiz
uchun rahmat!**