RO'YXATNI TARTIBLASH

Aksar holatlarda ro'yxat ichidagi elementlarni alifbo ketma-ketligida tartiblash talab qilinishi mumkin. Buning uchun list uchun maxsus `.sort()` metodidan foydalanamiz.

```
cars = ['bmw','mercedes benz', 'volvo', 'general motors', 'tesla', 'audi']
cars.sort()
print(cars)
```

Diqqat! Tartiblashda katta harflar kichik harflardan avval kelishini hisobga oling. Agar matndagi so'zlarning bosh harfi katta-kichik aralash yozilgan bo'lsa, ularni bir ko'rinishga keltirib olish maqsadga muvofiq bo'ladi.

```
cars = ['Bmw','mercedes benz', 'volvo', 'gm', 'tesla', 'audi']
cars.sort()
print(cars)
```

Yuqoridagi misolda 'Bmw' elementi katta harf bilan boshlangani uchun ro'yxatning boshidan joy oldi.

Ro'yxatni teskari tartibda saqlash uchun `.sort()` metodi ichida `reverse=True` argumentini ham kiritamiz.

```
cars = ['bmw','mercedes benz', 'volvo', 'general motors', 'tesla', 'audi']
cars.sort(reverse=True)
print(cars)
```

`.sort()` metodi ro'yxatni tartiblaydi. Ba'zida asl ro'yxat ichidagi elementlarning ketma-ketligini buzmagan holda ro'yxatni tartiblash talab qilinishi mumkin. Buning uchun `sorted()` funktsiyasidan foydalanamiz:

```
mehmonlar = ['Odil', 'Hamid', 'Temur', 'Avazbek', 'Farruh', 'Shamsiddin']
print("sorted() qaytargan ro'yxat:", sorted(mehmonlar))
print("Asl ro'yxat o'zgarmas qoldi:", mehmonlar)
```

`sorted()` funktsiyasi yordamida teskari tartiblash uchun ham `reverse=True` argumentini beramiz:

```
print(sorted(mehmonlar, reverse=True))
```

Yuoqridagi ikki usul bilan sonli ro'yxatlarni ham tartiblashimiz mumkin:

```
ages = [12, 98, 34, 65, 34, 76, 11]
ages.sort()
print(ages)
print(sorted(ages, reverse=True))
```

RO'YXATNI AYLANTIRISH

Ba'zida ro'yxatni aylantirish (boshini oxiriga, oxirini boshiga) talab qilinishi mumkin. Buning uchun `.reverse()` metodidan foydalanamiz.

```
fruits = ['pear','banana','apple','watermelon','lemon']
fruits.reverse()
print(fruits)
```

RO'YXATNING UZUNLIGINI BILISH

Ro'yxatning uzunligi, ya'ni uning ichidagi elementlar sonini aniqlash uchun `len()` funktsiyasidan foydalanamiz:

```
fruits = ['pear','banana','apple','watermelon','lemon']
print("Elementlar soni:",len(fruits)) # len(fruits) ro'yxat uzunligini qaytaradi
```

`range()` FUNKTSIYASI

Bu funktsiya yordamida biz ma'lum oraliqdagi sonlar ketma-ketligini yaratishimiz mumkin. `list()` funktsiyasi yordamida esa bu oraliqni ro'yxat shaklida saqlab olamiz:

```
sonlar = list(range(0,10)) #
print(sonlar)
```

Yuqoridagi misolda `range(0,10)` funktsiyasi `0` dan `9` gacha sonlar ketma-ketligini shakllantirdi, `list(range(0,9))` esa bu ketma-ketlikni ro'yxatga aylantirdi.

E'tibor qiling `range()` funsiyasi ikkinchi indeksdan bitta avval to'xtaydi.

`range()` yordamida qadamni ham berishimiz mumkin:

```
juft_sonlar = list(range(0,20,2)) # 0 dan 20 gacha 2 qadam bilan
toq_sonlar = list(range(1,20,2)) # 1 dan 20 gacha 2 qadam bilan
print("Juft sonlar: ", juft_sonlar)
print("Toq sonlar: ", toq_sonlar)
```

Agar sonlar ketma-ketligi 0 dan boshlansa, `range()` funktsiyasida yakuniy indeksni ko'rsatish kifoya. Misol uchun `range(0,10)` emas `range(10)` deb yozsak ham bo'lavradi.

SONLI RO'YXAT USTIDA SODDA AMALLAR

Pythonda ro'yxatlar ustida ba'zi sodda amallarni ham bajarish mumkin. Misol uchun ro'yxatdagi eng kichik sonni topish uchun `min()` funktsiyasidan, eng katta sonni topish uchun esa `max()` funktsiyasidan, sonlarning yig'indisini topish uchun esa `sum()` funktsyasidan foydalansak bo'ladi:

```
narhlar = [12000, 22500, 23456, 9800, 5600, 9934, 32874]
arzon = min(narhlar)
qimmat = max(narhlar)
```

jami = sum(narhlar)
print("Eng arzon narh ", arzon, ". Eng qimmati ", qimmat, ". Jami: ", jami)
RO'YXATNI KESISH
Ba'zida ro'yxatning ma'lum bir bo'lagini ajratib olish talab qilinishi mumkin, deylik biz quyidagi cars degan ro'yxatdan birinchi 3 ta elementni ajratib olmoqchimiz, buning uchun biz boshlang'ich va oxirgi indekslarni beramiz:
cars = ['bmw','mercedes benz', 'volvo', 'general motors', 'tesla', 'audi']
my_cars = cars[0:3] # 0-indeskdan boshlab 3 ta element ajratib olamiz
print(my_cars)
Python 2-indeksdan bitta avval to'xtaydi. Yuqoridagi misolda ham 0,1,2-elementlar ajratib olindi.
Bu usul bilan ro'yxatning istalgan joyidan bo'lishimiz mumkin:
print(cars[2:5]) # 2-3-4-elementlarni ajratib olamiz (5 kirmaydi)
Agar boshlang'ich indeksni bermasangiz, Python avtomat ravishda `0` indeksdan boshlab kesadi. Agar 2-indeksni kiritmasangiz, ro'yxat oxirigacha kesadi:
print(cars[:4]) # Ro'yxat boshidan 4-gacha kesadi (0,1,2,3)
print(cars[2:]) # 2-elementdan boshlab ro'yxat oxirigacha kesib oladi
RO'YXATDAN NUSXA OLISH
Dastur davomida biror ro'yxatdan nusxa olish talab qilinishi mumkin. Buning uchun biz tenglik(`=`) belgisidan foydalansak bo'ladimi? Quyidagi misolga e'tibor bering:
sonlar = [1, 2, 3, 4, 5] # donlar degan ro'yxat yaratamiz
sonlar2 = sonlar # sonlar2 degan ro'yxatni sonlar ga tenglaymiz
sonlar2.append(6) # sonlar2 ga 6 sonini qo'shamiz
sonlar2.append(7) # sonlar2 ga 7 sonini qo'shamiz
print("Bu sonlar ro'yxati:", sonlar)
print("Bu sonlar2 ro'yxati:", sonlar2)
Natija biz kutgandek chiqdimi? Yo'q. Biz 6 va 7 sonlarini `sonlar2` degan ro'yxatga qo'shgan edik, lekin bu ikki son `sonlar` degan asl ro'yxatga ham qo'shilib qoldi.
Demak yuqorida biz `sonlar2=sonlar` deb yozgan komandamiz yangi ro'yxat yaratish o'rniga, ikkala o'zgaruvchini ham bitta ro'yxatga bog'lab (yo'naltirib) qo'ydi. Biz `sonlar` yoki `sonlar2` ustida bajargan amallarimiz aslida bitta ro'yxat ustida bajarilyapti.
Ikki o'zgaruvchi, bir ro'yxat
Unda, qanday qilib ro'yxatdan nusxa olamiz? Buning uchun yuoqirdagi ka'bi ro'yxatni kesish usulidan foydalanamiz. Faqatgina, kvadrat qavs ichida ikkala indeksni ham ko'rsatmasdan, bo'sh qoldiramiz.
TUPLES - O'ZGARMAS RO'YXAT
Dastur yaratish davomida o'zgarmas ro'yxat tuzish talab qilinishi mumkin. Pythonda bunday ro'yxatlar **tuples** deb yuritiladi. Tuple ichidagi qiymatlarni bir marta, dastur boshida beriladi va so'ngra o'zgartirib bo'lmaydi. List dan farqli ravishda, Tuple e'lon qilishda kvadrat qavslar `[]` o'rniga oddiy qavslar `()` ishlatiladi:
tomonlar = (20, 30, 55.2)
print(tomonlar)
Tuple ichidagi elementlarga huddi ro'yxat elementlariga murojat qilingani kabi murojat qilinaveradi:
toys = ('bus','car','bear','dino','snake','lizard')
print(toys[0])
print(toys[-1])
print(toys[2:5])
Keling Tuple ichidagi biror elementning qiymatini o'zgartirib ko'ramiz:
toys = ('bus','car','bear','dino','snake','lizard')
toys[3] = 'dragon'
Agar Tuple ga o'zgartirish talab qilinsa, yagona yo'li o'zgarmas ro'yxatni `list()` funktsiyasi yordamida **List** (oddiy ro'yxat) ko'rinishiga keltirib olish, o'zgarishlarni bajarsih va qaytarib `tuple()` funktsiyasi yordamida o'zgarmas ro'yxatga o'tkazish mumkin:
toys = ('bus','car','bear','dino','snake','lizard') # o'zgarmas ro'yxat
toys = list(toys) # o'zgarmas ro'yxatni oddiy ro'yxatga (List) aylantiramiz
 # Ro'yxatga o'zgartirishlar kiritamiz
toys.append('dragon')
toys.remove('bus')
toys[1] = 'mcqueen'
toys = tuple(toys) # Ro'yxatni qaytadan o'zgarmas ro'yxatga (Tuple) aylantiramiz
print(toys