

6-Mavzu: List elementlari bilan ishlash

List — bu bir nechta qiymatlarni bitta o'zgaruvchida saqlash imkonini beruvchi to'plam turi.

Python'da [] kvadrat qavslar yordamida yaratiladi.

Misol: mevalar = ["olma", "banan", "gilos"]

1. RO'YXAT ELEMENTLARINI TARTIBLASH

sort() — ro'yxatni o'sish tartibida saralaydi:

```
sonlar = [7, 2, 9, 1]
```

```
sonlar.sort()
```

Natija: [1, 2, 7, 9]

sort(reverse=True) — kamayish tartibida saralaydi.

sorted() — yangi saralangan ro'yxat qaytaradi (aslini o'zgartirmaydi).

2. RO'YXATNI KESISH (SLICING)

Ro'yxatdan ma'lum qismini olish: ro'yxat[boshlanish:oxiri]

Misol: mevalar[1:4] → ['banan', 'gilos', 'shaftoli']

Boshqa misollar:

mevalar[:3] - Dastlabki 3 ta element

mevalar[2:] - 2-indeksdan oxirigacha

mevalar[-2:] - Oxirgi 2 element

3. RO'YXATNI NUSXALASH (COPY)

Usullar:

- copy() metodi: b = a.copy()

- Slicing: b = a[:]

- list() funksiyasi: b = list(a)

Eslatma: b = a qilsangiz, ikkala o'zgaruvchi bir xil bo'ladi.

4. TUPLES HAQIDA QISQACHA

Tuple — o'zgarmas ro'yxat (immutable list): mevalar = ("olma", "banan", "gilos")

Farqlari:

- List o'zgaradi, Tuple o'zgarmaydi

6-Mavzu: List elementlari bilan ishlash

- List: [], Tuple: ()
- Tuple biroz tezroq ishlaydi

Tuple'dan faqat o'qish, qidirish, takrorlash mumkin.

MISOLLAR:

1. Ro'yxatni kamayish tartibida saralang

```
sonlar = [10, 3, 7, 1, 5]
```

```
sonlar.sort(reverse=True)
```

2. Ro'yxatdan oxirgi 3 ta elementni chiqaring: sonlar[-3:]

3. Ro'yxat nusxasini olib, unga yangi element qo'shing:

```
yangi = sonlar.copy()
```

```
yangi.append(100)
```