

Python programming language

Week 6. Python dictionary, sets

Alisher Ismailov

Lecturer

Department of Information technology
Andijan Branch of Tashkent Financial Institute

Email: alisherismailov534@gmail.com

Python dasturlash tili

6-Mavzu. Python dictionary, sets

Ma'ruzachi: Alisher Ismoilov

Axborot texnologiyalari kafedراسي
Toshkent moliya instituti Andijon filiali

Elektron pochta: alisherismailov534@gmail.com

6-Mavzu. Python sets, dictionary

Reja:

- 🐍 Set to‘plami elementlari
- 🐍 Set to‘plami elementlari tartibsizdir
- 🐍 Set to‘plami elementlari o‘zgarmasdir
- 🐍 Set to‘plamida dublikatlarga ruxsat berilmagan
- 🐍 Set to‘plami uzunligini aniqlash
- 🐍 Set to‘plami elementlarini chop etish
- 🐍 Set to‘plamlarni birlashtirish
- 🐍 Set to‘plamiga har qanday to‘plamni qo‘shish mumkin
- 🐍 Set to‘plami elementini o‘chirib tashlash
- 🐍 Set to‘plami elementlari sikl orqali chop etish
- 🐍 Ikkita set to‘plamini birlashtirish
- 🐍 Set to‘plamlaridan dublikatlardan boshqa elementlarni chop etish
- 🐍 Set to‘plami funksiyalari

6-Mavzu. Python sets, dictionary

Reja:

- 🔗 Dictionary (Lugʻat) yaratish va chop etish
- 🔗 Dictionary (Lugʻat) toʻplami elementlari
- 🔗 Dictionary (Lugʻat) toʻplamida dublikatlarga ruxsat berilmagan
- 🔗 Dictionary (Lugʻat) toʻplami uzunligi
- 🔗 Dictionary (Lugʻat) toʻplamidagi elementlarini chop etish
- 🔗 Dictionary (Lugʻat) toʻplamida kalit soʻzlarni chop etish
- 🔗 Dictionary (Lugʻat) toʻplamida qiymatlarni chop etish
- 🔗 Dictionary (Lugʻat) toʻplami elementlarni chop etish
- 🔗 Dictionary (Lugʻat) toʻplamida qiymatlarni oʻzgartirish
- 🔗 Dictionary (Lugʻat) toʻplamini elementini yangilash
- 🔗 Dictionary (Lugʻat) toʻplamiga elementlarni qoʻshish
- 🔗 Dictionary (Lugʻat) toʻplamidan elementlarni oʻchirib tashlash
- 🔗 Dictionary (Lugʻat) toʻplamini for sikli orqali chop etish
- 🔗 Dictionary (Lugʻat) toʻplamidan nusxa koʻchirish
- 🔗 Dictionary (Lugʻat) toʻplami funksiyalari

Python dasturlash tilida Set

Python dasturlash tilidagi yana bir to'plam **Set** deb ataladi. **Set** to'plamlar figuralik qavslar `{}` orqali yaratiladi.

Python dasturlash tilida Set

- Set to'plami elementlari
- Set to'plami elementlari tartibsiz, o'zgarmas va takroriy qiymatlarga ruxsat bermaydigan to'plam hisoblanadi.

Python dasturlash tilida Set

- Set to'plamida bir xil qiymatga ega ikkita element bo'lishi mumkin emas.

```
dasturlar = {"MS Word", "Photoshop", "SolidWorks", "Photoshop"}
```

```
print(dasturlar)
```

```
{'SolidWorks', 'Photoshop', 'MS Word'}
```

- Agar bir xil qiymatga ega ikkita element bo'lsa, chop etilganda faqat oxirgi element chop etiladi.

Python dasturlash tilida Set

- Set to'plamida nechta element borligini aniqlash uchun `len()` funksiyasidan foydalanish mumkin.

```
In [72]: futbol_jamoa = {"MU", "Paxtakor", "Neftchi"}
```

```
print(len(futbol_jamoa))
```

```
3
```

Python dasturlash tilida Set

- Set to‘plami turli xil berilganlar turlarini o‘z ichiga olishi mumkin:

```
In [75]: futbol_jamoa = {"MU", 1930, "Paxtakor", "Neftchi", True}
print(futbol_jamoa)
{'Neftchi', True, 'Paxtakor', 1930, 'MU'}
```

Python dasturlash tilida Set

- Python dasturlash tilida, Set to'plami berilganlar turini aniqlash uchun `type()` funksiyasidan foydalanish mumkin:

```
In [24]: meva = {"olma", "behi", 55, "gilos", True}
```

```
print(type(meva))
```

```
<class 'set'>
```

Python dasturlash tilida Set

- Siz set toʻplami indeksiga murojaat qilib, toʻplamdagi elementlarni chop etishingiz mumkin. Agar Set toʻplami barcha elementlarini chop etmoqchi boʻlsangiz **for** siklidan foydalanish mumkin.

```
In [76]: shaxarlar = {"Andijon", "Toshkent", "Farg`ona"}  
  
for x in shaxarlar:  
    print(x)
```

```
Toshkent  
Andijon  
Farg`ona
```

Python dasturlash tilida Set

- Set to'plamida "Andijon" qiymati mavjudligini tekshirish:

```
In [78]: shaxarlar = {"Andijon", "Toshkent", "Farg`ona"}  
  
print("Andijon" in shaxarlar)  
  
True
```

Python dasturlash tilida Set

- Set to'plamiga bitta element qo'shish uchun **add()** funksiyasidan foydalanish mumkin.

```
In [80]: shaxarlar = {"Andijon", "Toshkent", "Farg`ona"}
          shaxarlar.add("Nukus")
          print(shaxarlar)
          {'Toshkent', 'Nukus', 'Andijon', 'Farg`ona'}
```

Python dasturlash tilida Set

- Mavjud set to'plamiga boshqa set to'plamidan elementlar qo'shish uchun update() funksiyasidan foydalanish mumkin.

```
Uzbek_shaxarlar = {"Andijon", "Toshkent", "Farg`ona"}  
Amerika_shaxarlar = {"LA", "New york", "Chicago"}
```

```
Uzbek_shaxarlar.update(Amerika_shaxarlar)  
print(Uzbek_shaxarlar)
```

```
{'Toshkent', 'Andijon', 'Chicago', 'LA', 'Farg`ona', 'New york'}
```

Python dasturlash tilida Set

- **update() funksiyasi** har qanday takrorlanadigan obyekt bo‘lishi mumkin (tuple, list, dictionaries).

```
In [84]: avtomobil = {"Nexia", "Matiz", "Lacetti"}
          zavot = ("GM", "Deawoo", "Ferrari")

          avtomobil.update(zavot)

          print(avtomobil)

          {'GM', 'Matiz', 'Lacetti', 'Nexia', 'Deawoo', 'Ferrari'}
```

Python dasturlash tilida Set

- Set to'plamidagi elementni o'chirib tashlash uchun **remove()** yoki **discard()** funksiyasidan foydalanish mumkin.

```
In [87]: zavot = {"GM", "Deawoo", "Ferrari"}  
  
zavot.remove("GM")  
  
print(zavot)  
  
{'Deawoo', 'Ferrari'}
```

Python dasturlash tilida Set

- Quyidagi mashgʻulotda **Discard()** funksiya yordamida "gilos" elementini oʻchirib tashlanadi:

```
In [88]: xona_nomer = {"101", "207", "709"}  
  
xona_nomer.discard("207")  
  
print(xona_nomer)  
  
{'101', '709'}
```

Python dasturlash tilida Set

- Set to‘plamidagi elementni o‘chirish uchun **pop()** funksiyasidan ham foydalanish mumkin, ammo bu funksiya faqatgina oxirgi elementni o‘chirib tashlaydi. Shuni esda tutish kerakki, set to‘plami tartibsiz, shuning uchun qaysi elementni o‘chirib tashlanishini oldindan bilmaymiz.
- Quyidagi suratda **pop()** funksiyasi orqali qaysi element o‘chirilganini ekranga chop etiladi va set to‘plami elementlarini bitta element o‘chirilganidan keyingi holatini chop etiladi.

```
In [89]: ijtimoiy_tarmoqlar = {"Instagram", "Facebook", "Twitter"}

ijtimoiy_tarmoqlar.pop()

print(ijtimoiy_tarmoqlar)

{'Twitter', 'Instagram'}
```

Python dasturlash tilida Set

- `del` kalit soʻzi set toʻplamni butunlay oʻchirib tashlaydi:

```
ijtimoiy_tarmoqlar = {"Instagram", "Facebook", "Twitter"}  
  
del ijtimoiy_tarmoqlar  
  
print(ijtimoiy_tarmoqlar)
```

```
-----  
NameError                                Traceback (most  
~\AppData\Local\Temp\ipykernel_6668\521383281.py in <modu  
      3 del ijtimoiy_tarmoqlar  
      4  
----> 5 print(ijtimoiy_tarmoqlar)  
  
NameError: name 'ijtimoiy_tarmoqlar' is not defined
```

Python dasturlash tilida Set

- **for** siklidan foydalanib set to'plami elementlarini ekranga chop etish ham mumkin:

```
In [91]: choy_turlari = {"Ko`k", "Qora", "Pamil"}  
  
for i in choy_turlari:  
    print(i)
```

Ko`k

Pamil

Qora

Python dasturlash tilida Set

- Python dasturlash tilida ikki yoki undan ortiq set to'plamlarini birlashtirishning bir necha yo'li mavjud. Siz ikkala to'plamdagi barcha elementlarni o'z ichiga olgan yangi to'plamni qaytaradigan **union()** funksiyasidan yoki bir to'plamdagi barcha elementlarni boshqasiga kiritadigan **update()** funksiyasidan foydalanishingiz mumkin:

```
In [94]: choy_turlari = {"Ko`k", "Qora", "Pamil"}
         taom_turlari = {"Lag`mon", "Dimlama", "Osh"}

         birlashtirish = choy_turlari.union(taom_turlari)

         print(birlashtirish)

{'Dimlama', 'Osh', 'Qora', 'Lag`mon', 'Ko`k', 'Pamil'}
```

Python dasturlash tilida Set

- Quyidagi misolda **update()** funksiyasi sabzavot set to'plamidagi elementlarni meva set to'plamiga kiritadi:

```
In [97]: choyxona = {"Cho`ntak", "Majnuntol", "Vodiy"}
shaxarlar = {"Asaka", "Andijon", "Farg`ona"}

choyxona.update(taom_turlari)

print(choyxona)

{'Majnuntol', 'Vodiy', 'Osh', 'Dimlama', 'Lag`mon', 'Cho`ntak'}
```

Python dasturlash tilida Set

- `intersection_update()` funksiyasi faqat ikkala to'plamda mavjud bo'lgan elementlarni chop etadi.

```
In [98]: Futbolchi_nomi = {"Ronaldo", "Messi", "Beckham"}
         jamoa = {"Real Madrid", "MU", "Barcelona", "Ronaldo"}

         Futbolchi_nomi.intersection_update(jamoa)

         print(Futbolchi_nomi)

         {'Ronaldo'}
```

Python dasturlash tilida Set

- **intersection()** funksiyasi faqat ikkala to'plamda mavjud bo'lgan elementlarni o'z ichiga olgan yangi to'plamni qaytaradi.

```
In [99]: Futbolchi_nomi = {"Ronaldo", "Messi", "Beckham"}
         jamoa = {"Real Madrid", "MU", "Barcelona", "Ronaldo"}

         Futbolchi_nomi.intersection(jamoa)

         print(Futbolchi_nomi)

         {'Messi', 'Beckham', 'Ronaldo'}
```

Python dasturlash tilida Set

- `symmetric_difference_update()` funksiyasi ikkala set to'plamida duplikat bo'lmagan elementlarni o'zida saqlaydi.

```
Futbolchi_Chet_El = {"Ronaldo", "Ahmedov", "Messi", "Beckham"}
Futbolchi_Uzbek = {"Ahmedov", "Qosimov", "Jeparov"}

Futbolchi_Chet_El.symmetric_difference_update(Futbolchi_Uzbek)

print(Futbolchi_Chet_El)

{'Beckham', 'Ronaldo', 'Qosimov', 'Jeparov', 'Messi'}
```

Python dasturlash tilida

Dictionary

Python dasturlash tilida kalit soʻzlari ingliz tilida boʻlgani uchun bu kitobda kalit soʻzlarni ingliz tilidagi holatida ishlatiladi. Dictionary oʻzbekchada lugʻat degan maʼnoni anglatadi. Dictionary (Lugʻat) toʻplami berilganlar qiymatlarini

Python dasturlash tilida Dictionary

kalit: qiymat juftlarida saqlash uchun ishlatiladi. Dictionary (Lugʻat) toʻplami oʻzgaruvchan va dublikatlarga ruxsat bermaydigan toʻplam hisoblanadi. Dictionary (Lugʻat) toʻplami figuralik qavslar {} bilan yaratiladi, kalit va qiymatlarga ega boʻladi:

Python dasturlash tilida Dictionary

```
In [103]: lugat = {  
    "name": "ism",  
    "home": "uy",  
    "study": "o`qimoq"  
}
```

```
print(lugat)
```

```
{'name': 'ism', 'home': 'uy', 'study': 'o`qimoq'}
```

Python dasturlash tilida Dictionary

- Dictionary (Lugʻat) toʻplamida bir xil kalitga ega ikkita element boʻlishi mumkin emas:

```
In [105]: lugat = {  
    "name": "ism",  
    "home": "uy",  
    "study": "o`qimoq",  
    "home": "hovli",  
}
```

```
print(lugat)
```

```
{'name': 'ism', 'home': 'hovli', 'study': 'o`qimoq'}
```

Python dasturlash tilida Dictionary

- **Dictionary** to'plamida nechta element borligini aniqlash uchun **len()** funksiyasidan foydalanish mumkin:

```
In [107]: lugat = {  
           "name": "ism",  
           "home": "uy",  
           "study": "o`qimoq",  
           }  
  
print(len(lugat))
```

3

Python dasturlash tilida Dictionary

- Dictionary (Lugʻat) toʻplamida elementlardagi qiymatlar har qanday turdagi berilganlar turiga ega boʻlishi mumkin:

```
lugat = {  
    "subject": "ot (gapdagi ot so`z turkumi)",  
    "adjective": "sifat",  
    "verbs": ["to do", "jump", "run"]  
}
```

```
print(lugat)
```

```
{'subject': 'ot (gapdagi ot so`z turkumi)', 'adjective': 'sifat', 'verbs': ['to do', 'jump', 'run']}
```

Python dasturlash tilida Dictionary

- Dictionary to'plamining berilganlar turini aniqlash uchun type() funksiyadan foydalanish mumkin:

```
In [12]: lugat = {  
        "komputer lingivistika": "Python lingivistlar uchun"  
        }  
  
print(type(lugat))  
  
<class 'dict'>
```

Python dasturlash tilida Dictionary

- Toʻrtburchak qavs [] ichidagi kalit nomiga murojaat qilib, **Dictionary** (lugʻat) elementlarini ekranga chop etish mumkin:

```
In [16]: lugat = {
           "komputer lingivistika": "Python lingivistlar uchun",
           "dasturlash": "Python dasturlash tili"
         }

print(lugat["dasturlash"])

Python dasturlash tili
```

Python dasturlash tilida Dictionary

- Xuddi shu natijani beradigan `get()` deb nomlangan funksiya ham bajarishi mumkin:

```
In [109]: lugat = {  
           "computer": "kompyuter",  
           "technology": "texnologiya",  
           }  
  
print(lugat.get("technology"))  
  
texnologiya
```

Python dasturlash tilida Dictionary

- keys() funksiyasi **dictionary** to'plamida barcha kalitlar ro'yxatini chop etadi.

```
In [110]: lugat = {  
           "computer": "kompyuter",  
           "technology": "texnologiya",  
           }  
  
           kalit = lugat.keys()  
           print(kalit)  
  
           dict_keys(['computer', 'technology'])
```

Python dasturlash tilida Dictionary

- Mavjud **Dictionary** to'plamiga yangi element qo'shish va kalitlar ro'yxati ham yangilanishini ko'rish mumkin:

```
lugat = {  
    "komputer lingivistika": "Python lingivistlar uchun",  
    "dasturlash": "Python dasturlash tili",  
    "amaliy mashg`ulotlar": "Python dasturida stemming yaratish"  
}
```

```
kalit = lugat.keys()  
print(kalit)
```

```
lugat["so`z turkumi"] = "fe'l"
```

```
kalit = lugat.keys()  
print(kalit)
```

```
dict_keys(['komputer lingivistika', 'dasturlash', 'amaliy mashg`ulotlar'])  
dict_keys(['komputer lingivistika', 'dasturlash', 'amaliy mashg`ulotlar', 'so`z turkumi'])
```

Python dasturlash tilida Dictionary

- `values()` funksiyasi **Dictionary** (lugʻat)dagi barcha qiymatlar roʻyxatini ekranga chop etadi.

```
In [111]: lugat = {
            "Ronaldo": "MU",
            "Mbappe": "PSG",
            "Benzema": "Real Madrid",
        }

qiymat = lugat.values()
print(qiymat)

dict_values(['MU', 'PSG', 'Real Madrid'])
```

Python dasturlash tilida Dictionary

- Qiymatlar ro'yxati **Dictionary** (lug'at) to'plamiga kiritilgan har qanday o'zgarishlar qiymatlar ro'yxatida aks etadi. Quyidagi suratda mavjud **Dictionary** (lug'at) to'plamiga o'zgartirish kiritilishi va qiymatlar ro'yxati ham yangilanishini ko'rishingiz mumkin:

```
In [112]: lugat = {
            "Ronaldo": "MU",
            "Mbappe": "PSG",
            "Benzema": "Real Madrid",
        }

qiymat = lugat.values()
print(qiymat)

lugat["Maradona"] = "Napoli"

qiymat = lugat.values()
print(qiymat)

dict_values(['MU', 'PSG', 'Real Madrid'])
dict_values(['MU', 'PSG', 'Real Madrid', 'Napoli'])
```

Python dasturlash tilida Dictionary

- `items()` funksiyasi **Dictionary** (lugʻat)dagi har bir elementni roʻyxatdagi tuple (kortej) sifatida qaytaradi.

```
lugat = {  
    "MS Word": "MS Office",  
    "Windows": "Microsoft",  
    "Macintosh": "Apple",  
}
```

```
qiymat = lugat.items()  
print(qiymat)
```

```
dict_items([('MS Word', 'MS Office'), ('Windows', 'Microsoft'), ('Macintosh', 'Apple')])
```

Python dasturlash tilida Dictionary

- Dictionary (Lugʻat)da "dasturlash" kalit soʻzi mavjudligini tekshirish:

```
In [114]: lugat = {
            "MS Word": "MS Office",
            "Windows": "Microsoft",
            "Macintosh": "Apple",
        }

        if "Windows" in lugat:
            print("Windows kalit soʻzi lugʻatda mavjud")
```

```
Windows kalit soʻzi lugʻatda mavjud
```

Python dasturlash tilida Dictionary

- Muayyan elementning qiymatini uning kalit nomiga murojaat qilib o'zgartirish mumkin:

```
lugat = {  
    "MS Word": "MS Office",  
    "Macintosh": "Apple",  
    "Windows": "Microsoft",  
}  
  
lugat["MS Word"] = "Windows"  
  
print(lugat)
```

```
{'MS Word': 'Windows', 'Macintosh': 'Apple', 'Windows': 'Microsoft'}
```

Python dasturlash tilida Dictionary

- **update()** funksiya **Dictionary** (lugʻat)ni berilgan argument elementlari bilan yangilaydi. Argument **Dictionary** (lugʻat) yoki kalit: qiymat juftliklari bilan takrorlanadigan obyekt boʻlishi kerak.

```
lugat = {  
    "O`zbekiston": "Tashkent",  
    "Amerika": "Washington",  
    "Rossiya": "Moskva",  
}  
  
lugat.update({"Amerika": "LA"})  
print(lugat)
```

```
{'O`zbekiston': 'Tashkent', 'Amerika': 'LA', 'Rossiya': 'Moskva'}
```

Python dasturlash tilida Dictionary

- **Dictionary** (Lugʻat) toʻplamiga element qoʻshish yangi indeks kaliti yordamida va unga qiymat belgilash orqali amalga oshiriladi:

```
lugat = {  
    "O`zbekiston": "Tashkent",  
    "Amerika": "Washington",  
    "Rossiya": "Moskva",  
}  
  
lugat["Malayziya"] ="Kuala Lumpur"  
print(lugat)
```

```
{'O`zbekiston': 'Tashkent', 'Amerika': 'Washington', 'Rossiya': 'Moskva',  
'Malayziya': 'Kuala Lumpur'}
```

Python dasturlash tilida Dictionary

- **Dictionary** (Lugʻat) toʻplamidan elementlarni oʻchirib tashlashning bir necha funksiyalari mavjud:
- **pop()** funksiyasi belgilangan kalit nomi bilan elementni oʻchirib tashlaydi:

```
In [120]: lugat = {  
           "GM Uzbekistan": "Tracker",  
           "Mercedes": "630",  
           "Tesla": "Tesla22",  
           }  
  
           lugat.pop("GM Uzbekistan")  
           print(lugat)  
  
           {'Mercedes': '630', 'Tesla': 'Tesla22'}
```

Python dasturlash tilida Dictionary

- `popitem()` funksiyasi oxirgi kiritilgan elementni o'chiradi.

```
In [123]: lugat = {
            "GM Uzbekistan": "Tracker",
            "Mercedes": "630",
            "Tesla": "Tesla22",
        }

lugat.popitem()
print(lugat)

{'GM Uzbekistan': 'Tracker', 'Mercedes': '630'}
```

Python dasturlash tilida Dictionary

- **del** kalit soʻzi koʻrsatilgan kalit nomi bilan elementni oʻchirib tashlaydi:

```
In [124]: lugat = {  
    "laptop": "Notebook",  
    "PC": "Shaxsiy Kompyuter",  
    "Phone": "Teelefon",  
}
```

```
del lugat["PC"]
```

```
print(lugat)
```

```
{'laptop': 'Notebook', 'Phone': 'Teelefon'}
```

Python dasturlash tilida Dictionary

- **del** kalit soʻzi ham **Dictionary** (lugʻat)ni butunlay oʻchirib tashlashi mumkin:

```
lugat = {  
    "laptop": "Notebook",  
    "PC": "Shaxsiy Kompyuter",  
    "Phone": "Teelefon",  
}
```

```
del lugat
```

```
print(lugat)
```

```
NameError                                Traceback (most recent call last)
```

```
~\AppData\Local\Temp\ipykernel_6668\482999755.py in <module>
```

```
    7 del lugat
```

```
    8
```

```
----> 9 print(lugat)
```

```
NameError: name 'lugat' is not defined
```

Python dasturlash tilida Dictionary

- `clear()` funksiyasi **Dictionary** (lugʻat)ni barcha elementlarini oʻchirish orqali toʻplamni boʻshatadi:

```
In [126]: lugat = {  
           "Amerika": "Dollar",  
           "O`zbekiston": "So`m",  
           "Qozog`iston": "Tenge",  
           }  
  
           lugat.clear()  
  
           print(lugat)  
  
           {}
```

Python dasturlash tilida Dictionary

- **Dictionary** (Lugʻat)dagi barcha asosiy kalit soʻzlar nomlarini birma-bir chop etadi:

```
In [127]: lugat = {  
    "Amerika": "Dollar",  
    "O`zbekiston": "So`m",  
    "Qozog`iston": "Tenge",  
    }  
  
for x in lugat:  
    print(x)
```

```
Amerika  
O`zbekiston  
Qozog`iston
```

Python dasturlash tilida Dictionary

- **Dictionary** (Lugʻat)dagi barcha qiymatlarni birma-bir chop etish uchun qiymatlarning indekslaridan foydalanamiz:

```
In [128]: lugat = {  
    "Amerika": "Dollar",  
    "O`zbekiston": "So`m",  
    "Qozog`iston": "Tenge",  
}  
  
for x in lugat:  
    print(lugat[x])
```

Dollar

So`m

Tenge

Python dasturlash tilida Dictionary

- **Dictionary** (Lugʻat) toʻplamida qiymatlarini qaytarish uchun siz **values()** funksiyasidan ham foydalanishingiz mumkin:

```
In [129]: lugat = {  
           "bir": "1",  
           "ikki": "2",  
           "uch": "3",  
           }  
  
for x in lugat.values():  
    print(x)
```

```
1  
2  
3
```

Python dasturlash tilida Dictionary

- **Dictionary** (Lugʻat) kalit soʻzlarini qaytarish uchun **keys()** funksiyasidan foydalanishingiz mumkin:

```
In [130]: lugat = {  
           "bir": "1",  
           "ikki": "2",  
           "uch": "3",  
           }  
  
           for x in lugat.keys():  
               print(x)
```

```
bir  
ikki  
uch
```

Python dasturlash tilida Dictionary

- `items()` funksiyasidan foydalanib, ikkala kalit va qiymatlarini ekranga chop etish mumkin:

```
In [131]: lugat = {  
           "suv": "water",  
           "havo": "air",  
           "tuproq": "soil",  
           }  
  
for x, y in lugat.items():  
    print(x, y)
```

```
suv water  
havo air  
tuproq soil
```

Python dasturlash tilida Dictionary

- **Dictionary** to'plamlaridan nusxa olish funksiyalari mavjud bo'lib, ulardan biri **copy()** funksiyasidan foydalanishdir.

```
lugat = {  
    "application": "ilova",  
    "software": "dastur",  
    "web-software": "web dastur",  
}
```

```
lugat2 = lugat.copy()  
print(lugat2)
```

```
{'application': 'ilova', 'software': 'dastur', 'web-software': 'web dastur'}
```

Python dasturlash tilida Dictionary

- Nusxa olishning yana bir usuli - **dict()** funksiyasidan foydalanishdir.

```
lugat = {  
    "my": "mening",  
    "yours": "sizning",  
    "ours": "biznig",  
}
```

```
lugat2 = dict(lugat)  
print(lugat2)
```

```
{'my': 'mening', 'yours': 'sizning', 'ours': 'biznig'}
```

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!**