**AMALIY MASHG‘ULOT UCHUN O‘QUV MATERIALLARI**

**1-Mavzu:** “Python dasturlash tili” faniga kirish va asosiy tushunchalari.

**8-mashg‘ulot.** Pythonda Setlar ro‘yxati bilan ishlash.

**O‘quv savollari:**

1. Setlar va ularning qo‘llanilishi.
2. Setlarni yaratish usullari.
3. Setlar bilan ishlovchi metodlar.

**1. Setlar va ularning qo‘llanilishi.**

**To‘plam (set)** shunday to‘plamki, uning har bir elementi to‘plamda yagona, takrorlanmas hisoblanadi.Bunday turdagi to‘plamnianiqlash uchun jingalak qavslardan foydalaniladi:

set\_list= {"Tomson", "Bobbi", "Alisey", "Tomson"}

print(set\_list)  # {"Tomson", "Bobbi", "Alisey"}

E’tibor berish kerakki, print funktsiyasi yordamida to‘plamni ekranga chiqarilganda "Tomson" elementini bir marta chiqarildi. Ammo e’lon qilinganda "Tomson" ham, to‘plam ta’rifi ushbu elementni ikki marta o‘z ichiga oladi.

**2. Setlarni yaratish usullari**

Buning sababi shundaki, to‘plam faqat yagona qiymatlarni o‘z ichiga oladi.

Bundan tashqari, to‘plamni aniqlash uchun elementlar ro‘yxati yoki to‘plami uzatiladigan **set()** funktsiyasidan foydalanish mumkin**:**

set\_persons = ["Maya", "Billi", "Teddi"]

set\_users = **set**(set\_persons)

**print**(set\_users)     # {"Maya", "Billi", "Teddi"}

**set()** funktsiyasi bo‘sh to‘plam yaratish uchun quyidagicha foydalaniladi:

set\_users= **set**()

**3. Setlar bilan ishlovchi metodlar**

Boshqa turdagi toplamlar singari ***set()*** to‘plamida ham uzunligini aniqlovchi ***len()*** funksiyasi mavjud:

foydalanuvchilar = {"Tomson", "Bobi", "Alison"}

**print** (**len**(foydalanuvchilar))        # 3

**add() metodi**

To‘plamga bitta element qo‘shish uchun **add()** metodidan foydalaniladi:

set\_users = **set**()

set\_users.**add**("Sammi")

**print** (set\_users) # "Sammi"

**remove() metodi**

**Element olib tashlash.** Bitta elementni olib tashlash **remove()** metodi yordamida amalga oshiriladi va olib tashlanadigan element qaytariladi. Ammo shuni yodda tutish kerakki, agar bunday element to‘plamda bo‘lmasa, unda xatolik yuzaga keladi. Shuning uchun, o‘chirishdan oldin, in operatori yordamida element mavjudligini tekshirish kerak:

set\_users = {"Tomson", "Bobi", "Alison"}

set\_user = "Tomson"

**if** set\_user **in** set\_users:

    set\_users.**remove**(set\_user)

print(set\_users)     # {"Bobi", "Alison"}

**discard** () metodi

**Elementni olib tashlash uchun discard** () usulini ham qo‘llash mumkin. Ammo element mavjud bo‘lmasa, xatolik qaytariladi:

set\_users = {"Tomson", "Bobi", "Alison"}

set\_users.**discard**("Tim") *# element "Tim" mavjud emas va metod hech narsa qilmaydi*

print(set\_users)     *# {"Tomson", "Bobi", "Alison"}*

set\_users.**discard**("Tomson")  *# "Tomson" elementi mavjud va usul elementni olib tashlaydi*

print(set\_users)     *#   {"Bobi", "Alison"}*

**clear()**  metodi

Barcha elementlarni olib tashlash uchun **clear()** usuli chaqiriladi :

set\_users.**clear**()

**To‘plam elementlariga sikl yordamida murojaat qilish**

Elementlarga murojaat qilish uchun for siklidan foydalanish mumkin :

set\_users = {"Tomson", "Bobi", "Alison"}

**for** user **in** set\_users:

**print** (user)

Takrorlash paytida har bir element ***user*** o‘zgaruvchisiga uzatiladi va u o‘z navbatida ushbu elementlarni konsolga chiqaradi.

**copy** **() metodi**

**To‘plamlarni nusxalash. Copy** () usulidan foydalanib, bir to‘plam elementlarini boshqa o‘zgaruvchiga nusxalash mumkin:

set\_users = {"Tomson", "Bobi", "Alison"}

talabalar = set\_users.**copy**()

print (talabalar)      # {"Tomson", "Bobi", "Alison"}

**union() metodi**

**To‘plamlarni birlashtirish. union()** usuli ikkita to‘plamni birlashtiradi va yangi to‘plamni qaytaradi:

set\_users = {"Tomson", "Bobbi", "Alison"}

users2 = {"Semmi", "Keyt", "Bobbi"}

users3 = set\_users.**union** (users2)

print (users3)

# {"Tomson", "Bobi", "Alison","Semmi", "Keyt"}

**intersection() metodi**

**To‘plamlarning kesishishi.** To‘plamlarning kesishishi faqat ikkala to‘plamda bir vaqtning o‘zida bo‘lgan elementlarni olish imkonini beradi. **Intersection()** metodi to‘plamlar kesishish funksiyasini bajaradi va kesishmadan hosil bo‘lgan yangi to‘plamni qaytaradi:

set\_users = {"Tomson", "Bobbi", "Alison"}

users2 = {"Sammi", "Keyt", "Bobbi"}

users3 = set\_users.**intersection**(users2)

print (users3)    # {"Bobbi"}

**and** operatori

Kesishish usuli o‘rniga mantiqiy ko‘paytirish operatsiyasi (***and***) dan foydalanish ham mumkin:

set\_users = {"Tomson", "Bobbi", "Alison"}

users2 = {"Sammi", "Keyt", "Bobbi"}

print (set\_users and users2)    # {"Bobbi"}

Bu ikkala holda ham, bir xil natijaga erishiladi.

**intersection\_update()** metodi

**intersection\_update()** metodi birinchi to‘plamni kesishish elementlari bilan almashtiradi:

set\_users = {"Tomson", "Bobbi", "Alison"}

users2 = {"Sammi", "Keyt", "Bobbi"}

set\_users.**intersection\_update**(users2)

print(set\_users)    # {"Bobbi"}

**DIFFERENCE() metodi.**

Ushbu metod to‘plamlar orasidagi farqlanuvchi elementlarni qaytaradi. Ya’ni birinchi to‘plamda bo‘lgan, ammo ikkinchisida yo‘q bo‘lgan elementlarni qaytaradi. To‘plamlar farqini olish uchun difference usuli yoki ayirish operatsiyasidan foydalaniladi mumkin:

set\_users = {"Tomson", "Bobbi", "Alison"}

users2 = {"Sammi", "Keyt", "Bobbi"}

users3 = set\_users.**difference**(users2)

print(users3)            *# {"Tomson", "Alison"}*

print (set\_users - users2)    *# {"Tomson", "Alison"}*

**symmetric\_difference()**  metodi

To‘plam farqining alohida turi - simmetrik farq **symmetric\_difference()** usuli yoki ^ operatsiyasi yordamida amalga oshiriladi. U ikkala to‘plamning umumiy elementlardan tashqari barcha elementlarini qaytaradi:

set\_users = {"Tomson", "Bobbi", "Alison"}

users2 = {"Sammi", "Keyt", "Bobbi"}

users3 = set\_users.**symmetric\_difference**(users2)

print(users3)    *# {"Tomson", "Elison", "Sammi", "Keyt"}*

users4 = **set\_users ^ users2**

print(users4)    *# {"Tomson", "Alison", "Sammi", "Keyt"}*

**issubset** metodi

**issubset** metodi joriy toʻplam boshqa toʻplamining kichik toʻplami (yaʼni qismi) ekanligini aniqlash imkonini beradi:

set\_users = {"Tomson", "Bobbi", "Alison"}

users2={"Sammi", "Keyt", "Bobbi", "Alison", "Tomson"}

print (set\_users.**issubset**(users2))    # True

print(users2.**issubset**(set\_users))  # False

**issuperset** metodi

Agar joriy to‘plam boshqa to‘plamning super to‘plami (ya’ni o‘z ichiga olgan) bo‘lsa, **issuperset** usuli **True** qiymatini qaytaradi:

set\_users = {"Tomson", "Bobbi", "Alison"}

users2={"Sammi", "Keyt", "Bobbi", "Alison", "Tomson"}

print(set\_users.**issuperset**(users2))    *# False*

print (users2.**issuperset**(set\_users))    *# True*

**frozenset** matodi

**Muzlatilgan to‘plam** (*frozen set*) turidagi to‘plamlar o‘zgartirib bo‘lmaydigan to‘plamlar turiga kiradi. Uni yaratish uchun **frozenset** funksiyasidan foydalaniladi:

set\_user= **frozenset**({"Tommi","Bobbi","Elison"})

**Frozenset** funktsiyasiga elementlar to‘plami - ro‘yxat, kortej boshqa to‘plamlar uzatiladi.

  Bunday to‘plamga yangi elementlarni qo‘shish, undagi mavjud elementlarini o‘chirib tashlash kabi amallarni bajarishning ilojisi yo‘q. Aynana shuning uchun **frozenset** to‘plami  operatsiyalarning cheklangan to‘plamini qo‘llab-quvvatlaydi:

***len(s):*** to‘plam uzunligini aniqlash;

***x in s*** : agar ***s*** to‘plamida ***x*** elementi mavjud bo‘lsa, True qiymatini qaytaradi

***x not in s:*** agar ***x*** ***s*** to‘plamining tarkibida bo‘lmasa, True qaytaradi

***s.issubset(t):*** agar ***t s*** toʻplamini oʻz ichiga olgan boʻlsa, True qiymatini qaytaradi

***s.issuperset(t):*** agar ***t s*** ichida bo‘lsa, True qiymatini qaytaradi

***s.union(t):*** ***s*** va ***t*** to‘plamlarning birlashuvini qaytaradi

***s.intersection(t):*** ***s*** va ***t*** ning kesishishini qaytaradi

***s.difference(t):*** ***s*** va ***t*** to‘plamlari orasidagi farqni qaytaradi

***s.copy():*** ***s*** to‘plamining nusxasini qaytaradi

**Nazorat savollari:**

1. To‘plamlarni yaratishning qanday usullarini bilasiz?
2. set() metodining vazifasi qanday?
3. To‘plamlarni lug‘atlardan farqi nimada?
4. To‘plamga element qo‘shish qanday amalga oshiriladi?
5. To‘plamga element olib tashlash qanday amalga oshiriladi?
6. **discard**() metodining vazifasi qanday?
7. **clear**() metodining vazifasi qanday?
8. **copy**() metodi nima maqsadda ishlatiladi va uning afzallik hamda kamchiliklari qanday?
9. **union()** metodi haqida gapirib bering.
10. **intersection()** qanday metod?
11. **intersection\_update()** metodining vazifasi va **intersection()** metodidan farqi haqida gapirib bering.
12. **symmetric\_difference()**  metodi haqida gapirib bering.
13. **issubset()** – metodining ishlash prinsipi qanday?
14. **issuperset()** – metodidan qanday foydalaniladi?
15. **frozenset()** funksiyasining vazifasi nima?