Avvalgi darsimizda yaratgan barcha funksiyalarimiz konsolga ma'lumot chiqarayotgan edi. Aslida, aksar holatlarda bu g'ayritabiiy. Sababi, dasturchi sifatida biz konsolga chiqqan ma'lumotdan unumli foydalana olmaymiz. Konsoldagi qiymatni o'zgaruvchiga yuklab, undan kelajakda foydalanib ham bo'lmaydi. Mana shunday holatlarda, funksiyadan unumli foydalanish uchun undan biror qiymatni qaytarish maqsadga muvofiq bo'ladi.

**FUNKSIYADAN ODDIY QIYMAT QAYTARISH**

Keling ism va familiya degan parametrlarni olib, toliq\_ism qaytaradigan sodda funksiya yasaymiz. **Kod\_1** Yuqoridagi funksiyamizga ahamiyat bersangiz, uning badanida endi print() funksiyasi yo'q. Buning o'rniga, funksiyamiz return operatori yordamida toliq\_ism degan o'zgaruvchining qiymatini qaytaradi. Endi funksiyadan to'g'ri foydalanish uchun u qaytargan qiymatni biror o'zgaruvchiga yuklashimiz kerak. Kod\_1 Line 5, 6

Yuqoridagi kodlarni bajarganimizda konsolga hech narsa chiqmaydi. talaba1 va talaba2 o'zgaruvchilarining qiymatini ko'rish uchun esa print() funksiyasidan foydalanamiz. **Kod\_1 Line 7**

Demak, qiymat qaytaradigan funksiyaning afzalligi shundaki, biz bu qiymatlardan keyin ham bemalol foydalanishimiz mumkin. Funksiya ichidagi o'zgaruvchilar mahalliy yoki ichki o'zgaruvchilar deyiladi (local variables). Ichki o'zgaruvchilar faqatgina funksiya ichida mavjud bo'ladi, ularga tashqaridan murojat qilib bo'lmaydi. Shuning uchun ham funksiya o'zgaruvchi emas qiymat qaytaradi.

**IXTIYORIY ARGUMENTLAR**

Avvalgi darsizmida funksiyalarga standart parametr berishni ko'rgan edik. Huddi shu usul bilan, ba'zi argumentlarni ixtiyoriy qilishimiz mumkin. Ya'ni funksiya ishlashi uchun bu agrumentarni kiritish majburiy emas, ixtiyoriy bo'ladi. Keling avvalgi funksiyamizni o'zgartiramiz va unga yana bitta otasiningismi degan paramter qo'shamiz, lekin bu parametr ixtiyoriy bo'ladi. Buning uchun funksiya yaratishda otasining\_ismi='' deb yozib ketamiz. Kod\_2

Yuqoridagi funksiyani tahlil qiladigan bo'lsak, 3-qatorda biz otasining ismi parametri bo'sh yoki yo'qligini tekshiramiz. Pythonda if dan so'ng bo'sh bo'lmagan matn (string) yozsak, bu shart True qaytaradi. Demak, bu ixtiyoriy parametr kiritilgani yoki yo'qligiga qarab, funksiyamiz turlicha qiymat qaytaradi.

**FUNKSIYADAN LUG'AT QAYTARAMIZ**

Funksiyadan sodda qiymat emas, ro'yxat, lu'gat va boshqa ma'lumot turlarini ham qaytarishimiz mumkin. Quyidagi funksiya ham mashina haqidagi ma'lumotlarni jamlab, ularni lug'at ko'rinishida qaytaradi: Kod\_3 E'tibor bering, narhi nomli parametrga None standart qiymatini berib ketdik. None Pythonda mavjud emas ma'nosini beradi, va if yordamida tekshirganda False mantiqiy qiymatini qaytardi.

FUNKSIYADAN RO'YXAT QAYTARAMIZ

Biz avvalroq range() funksiyasi bilan tanishgan edik. Bu funksiya 2 ta son qabul qilib, shu ikki son orali'g'idagi sonlarni qaytaradi. Keling biz oraliq() degan yangi funksiya yaratamiz. range() dan farqli ravishda, funksiyamiz 2 son oralig'idagi sonlarni ro'yxat ko'rinishida qaytarsin.Kod\_4 Yuqoridagi funksiyaga uchinchi, qadam deb nomlangan ixtiyoriy parameterni qo'sha olasizmi? Bajarishga harakat qilib koramiz

**FUNKSIYALARNI TSIKLDA ISHLATISH**

Funksiyalarni takrorlash uchun tsikldan foydalanishimiz mumkin. Quyidagi misolda biz while yordamida avvalroq yaratgan avto\_info funksiyamizni bir necha bor chaqiramiz va salondagi avtolar ro'yxatini shakllantiramiz. Bunda, ro'yxatning har bir elementi avto\_info funksiyasidan qaytgan lug'at bo'ladi. Kod\_5 Ro'yxatga bir necha qiymatlar qo'shing. Natijalarni konsolga chiroyli qilib chiqaring: Kod\_5

**FUNKSIYAGA RO'YXAT UZATISH**

Biz avvalgi darslarimizda funksiyaga parametr sifatida yagona qiymat berayotgan edik. Aslida, bu bilan cheklanmasdan, funksiyaga ro'yxat (list) ham berishimiz mumkin. Bunda, funksiya ro'yxat qiymatlariga to'g'ridan-to'g'ri murojat qila oladi. Kod\_6 Keling talabalarni baholaydigan funksiya yozamiz. Funksiyamiz talabalar ro'yxatini qabul qilib oladi, ro'yxatdan har bir talabani sug'urib olib (.pop()), bahosini kiritishni so'raydi. Talaba ismi va bahosini lug'atga joylab, yakuniy lug'atni foydalanuvchiga qaytaradi. Kod\_6

**RO'YXATGA O'ZGARTIRISH KIRITISH**

Funksiyaga ro'yxat uzatganimizda, funksiya ro'yxat elementlariga to'g'ridan-to'g'ri murojat qila oladi. Ro'yxatga funksiya ichida kiritilgan o'zgartirishlar asl ro'yxatga ham ta'sir qiladi. Avvalgi misolimizga qaytaylik: Kod\_7

talabalar = ['ali', 'vali', 'hasan', 'husan']

baholar = bahola(talabalar)

print(talabalar)

**Natija: []**

Yuqoridagi funksiya unga uzatilgan ro'yxat ichidagi talabalarning ismini .pop() yordamida sug'urib olgani uchun bizning asl ro'yxatimiz ham bo'shab qoldi. E'tibor bering, funksiya tashqarisidagi va ichidagi ro'yxatlar ikki hil nomlangan bo'lsada (talabalar va ismlar), ikkalasi ham xotiradagi bitta ro'yxatga bog'langani sabab ulardan biriga o'zgartirish kiritilishi bilan, ikkinchisi ham o'zgaradi.

**ASL RO'YXATGA O'ZGARTIRISH KIRITISHNING OLDINI OLISH**

Agar funksiya asl ro'yxatga o'zgartirish kiritishini istamasangiz, funksiyaga ro'yxatning o'zini emas, uning nusxasini uzatish mumkin. Buning uchun funksiya parametrini **royxat\_nomi[:]** ko'rinishida yozish kifoya. Bunda [:] operatori ro'yxatdan nusxa olishni bildiradi: Kod\_8

talabalar = ['ali', 'vali', 'hasan', 'husan']

baholar = bahola(talabalar[:])

print(talabalar)

Natija: ['ali', 'vali', 'hasan', 'husan']