Agar funksiyangiz bir nechta argument qabul qilishi kerak bo'lsa-yu, lekin siz argumentlar sonini aniq bilmasangiz, Pythonda istalgancha qiymat qabul qiluvchi funksiya yaratish imkoniyati bor.

**\*args USULI**

Agar funksiya qabul qiladigan parametrlar soni noaniq bo'lsa, va parametrlar yagona qiymatlar ko'rinishida uzatilsa, funksiya yaratishda argumentdan avval yulduzcha qo'yiladi (\*arguments).

**Kod\_1 ni** ko'raylik. summa() nomli funksiyamiz istalgancha sonlarni qabul qilib oladi, va ularning yi'gindisi hisoblaydi:

\*args usulida, bacha uzatilgan parametrlar (bir dona bo'lsa ham) funksiya ichida o'zgarmas ro'yxatga (tuple) joylanadi. Bundan kelib chiqib, yuqoridagi funksiyamizni yanada soddalashtirib yozishimiz mumkin: yani **Kod\_2**

Agar funksiya bir nechta argument qabul qilsa, \*args argument doim oxirida yoziladi.

**\*\*kwargs USULI**

Agar funksiyaga kalit so'z - qiymat ko'rinishidagi argumentlarni uzatish talab qilinsa, va bunday parametrlar soni noma'lum bo'lsa, argument oldidan ikkita yulduzcha qo'yiladi (\*\*kwargs).

**\*\*kwargs — keyword arguments (kalit so'zli argumentlar) Kod\_3**

Yuqoridagi funksiyamiz kompaniya va model degan ikki qiymatni qabul qiladi, undan keyin esa funksiyaga istalgancha parametr uzatish mumkin. Bunday funksiyaga parametrlar kalitso'z=qiymat ko'rinishida uzatiladi.

Funksiya ichida avval foydalanuvchi kiritgan qo'shimcha qiymatlardan iborat malumotlar deb nomlangan lug'at shakllantiriladi. Undan keyin esa majburiy parametrlarni lug'atga qo'shamiz. **Kod\_3**

**MODUL NIMA?**

Funksiyaning qulayliklaridan biri, ko'p takrorlanadigan kodlarni funksiya ichida yashirishimiz va kerak bo'lganda murojat qilishimiz mumkinligida. Maqsadimiz dasturimizni ixcham va tushunarli qilib, kelajakda o'zimiz yoki boshqalar uchun ham "toza" kod qoldrisih. Bu yo'nalishda yana bir qadam qo'yib, dasturimizni modullarga ajratimshimiz mumkin.

Modul bu loyihamiz ichidagi alohida fayl bo'lib, dasturimiz davomida ishlatiladigan funskyalarni (va o'zgaruvchilarni) mana shu faylga joylab, ko'zdan yashirib qo'yishimiz mumkin. Bu bizga asosiy dasturimizdan chalg'imasdan kod yozish imkoniyatini beradi.

Modul va uning ichidagi funksiyalarni istalgan payt asosiy dasturimizga yuklab olishimiz, modullarni boshqa dasturchilar bilan ulashishimiz yoki kelajakda o'zimizning boshqa loyihalarimizda foydalanishimiz mumkin.

Umuman olganda katta dasturlar bir nech o'nlab modullardan iborat bo'lishi tabiiy hol.

**MODUL YARATAMIZ**

Modul yaratish uchun asosiy dasturimizdagi funksiyalarni yangi faylga ko'chiramiz xolos. Modulga oson murojat qilishimiz uchun, faylimiz asosiy dasturimiz bilan bitta papkada bo'lgani afzal. Bunda adashib ketmaslik uchun, loyihangizning (dasturning) asosiy faylini main.py deb nomlash o'rinli.

Keling, biz ham avto\_info\_mod.py degan fayl yaratamiz va ichiga quyidagi 3 ta funksiyalarni joylaymiz: **Kod\_4**

**Kod\_4** funksiyalarga asosiy dasturdan murojat qlishning bir necha usuli bor.

**MODULNI CHAQIRIB OLISH**

Modul ichidagi istalgan funksiyaga murojat qilish uchun import modul\_nomi komandasidan foydalanamiz. Bunda modul ichidagi istalgan funksiyaga modul\_nomi.funksiya\_nomi() ko'rinishida murojat qilishimiz mumkin. Ya'ni avval modul nomi undan keyin esa nuqta qo'yilib, modul ichidagi funksiya nomi yoziladi.

Keling, **Kod\_4** modulimizdagi avto\_info() va info\_print() funksiyalariga murojat qilamiz: Murojat kodi: **Kod\_5**

**MODULGA QISQA NOM BERISH**

**Kod\_5** usul bilan modulni chaqirib olishda fayl nomi uzun bo'lsa bu o'ziga yarasha noqulayliklar tug'dirishi mumkin. Buning oldini olish uchun esa, modulni chaqirganda unga as operatori yordamida qisqa nom berishimiz, va modulga qisqa nom orqali murojat qilish mumkin. **Kod\_6** misolda avto\_info\_mod ni qisqa qilib aim deb nomlab oldik, va 3-4-qatorlarda modulga murojat qilishda qisqa nomidan foydalandik.

Modulga qisqa nom berganingizda bunday nomli boshqa o'zgaruvchi yoki funksiya yo'qligiga amin boling. Shunigdek, dastur davomida bu nomni boshqa o'zgaruvchilarga yoki funksiylarga berib yubormang.

**MODUL ICIHDAN MA'LUM FUNKSIYALARNI KO'CHIRIB OLISH**

Agar katta modullardan faqatgina ba'zi funksiyalarga murojat qilish talab qilinisa, kerakli funksiyalarni from modul\_nomi import funksiya1,

funksiya2 komandasi yordamida o'zimizning dasturimizga ko'chirib olishimiz mumkin. Bu usulning qulayligi, endi funksiyalarga to'g'ridan-to'g'ri murojat qilish mumkin (modul ismini yozmagan holda).

Misol uchun **Kod\_6** biz faqatgina avto\_info va info\_print funksiyalaridan foydalandik. Shu funksiyalarni asosiy kodimizga ko'chirib olaylik: **KOD\_7**

**FUNKSIYALARGA QISQA NOM BERISH**

Huddi avvalgidek, ko'chrib olgan funksiyamizga ham qisqa nom berishimiz mumkin. **Kod\_8**

**MODUL ICHIDAGI BARCHA FUNKSIYALARNI KO'CHIRIB OLISH**

Modul ichidagi barcha funksiyalarni asosiy dasturga ko'chirib olish uchun from modul\_nomi import \* komandasidan foydalanamiz. **Kod\_9**

**Diqqat! Bir necha sabablarga ko'ra bu usuldan foydalanish tavsiya qilinmaydi. Katta modullarda yuzlab funksiyalar bo'lishi mumkin, va funksiya nomi sizning dasturingizdagi boshqa funksiya yoki o'zgaruvchi bilan bir hil nomga ega bo'lsa, dastur xato ishlashiga olib keladi.**

**MODULDA O'ZGARUVCHI SAQLASH**

Modullarning ichida nafaqat funksiyalar, balki turli o'zgaruvchilarni ham saqlash mumkin. Modul ichidagi o'zgaruvchilarga ham huddi yuqoridagi usullar bilan murojat qilish mumkin.

PYTHON MODULLARI

Python dasturlash tili tayyor modullar bilan keladi, modullarning to'liq ro'yxatini **pypi.org** sayti orqali ko’rishimiz mumkin!

Keling ulardan ba'zilari bilan tanishamiz.

**math MODULI**

Bu modulda matematik hisob kitoblarni bajaruvchi funksiyalar va o'zgaruvchilar joylashgan. Keling ularga ba'zi misollarni ko'ramiz.

sqrt() - qavs ichida berilgan qiymatning kvadrat ildizini qaytaradi **Kod\_10**

pow(x,y) - x ning y-darajasini qaytaradi **Kod\_11**

pi - ning qiymatini saqlovchi o'zgaruvchi **Kod\_12**

log2(x) va log10(x) - x ning 2 va 10-lik logorifmini qaytaruvchi funksiyalar  **Kod\_13**

Bular math ichidagi ayrim funksiyalar edi xolos aslida bu modul ichida butun bir boshli matematika bor desak mubolag’a bo’lmaydi!

math moduli ichidagi barcha funksiyalar bilan Pythonning rasmiy sahifasida yoki **pypi.org** saxifalarida tanishishingiz mumkin.

**random MODULI**

Random moduli tasodifiy sonlar bilan ishlash uchun qulay funksiyalarga boy. Keing ulardan ayrimlari bilan tanishamiz.

randint(a,b) Bu funksiya a va b oraligi'da tasodifiy butun son qaytaradi. **Kod\_14**

x ning ichidan tasodifiy qiymatni qaytaruvchi funksiya. Bunda x bir necha elementdan iborat o'zgaruvchi (matn, ro'yxat) bo'lishi kerak.

**Kod\_15 Hamda Kod\_16**

x ichidagi elementlarni tasodifiy tartibda qaytaruvchi funksiya. Bunda x bir necha elementdan iborat o'zgaruvchi (matn, ro'yxat) bo'lishi kerak.

random moduli ichidagi boshqa funksiyalar haqida Python rasmiy sahifasidan ma'lumot olishingiz mumkin.