

PYTHON DASTURLASH TILI TARIXI VA IMKONIYATLARI**Mahmudov Muhammadxon Nuralixon o'g'li**

Andijon davlat universiteti, Axborot texnologiyalari va kompyuter injiniringi fakulteti, Axborot texnologiyalari kafedrası o'qituvchisi

Python - bu har xil turdagi ilovalarni yaratish uchun mo'ljallangan mashhur yuqori darajadagi dasturlash tili. Bularga veb-ilovalar, o'yinlar, ish stoli dasturlari va ma'lumotlar bazalari bilan ishlash kiradi. Python mashinani o'qitish va sun'iy intellektni tadqiq qilish sohasida ancha keng tarqaldi.

Python tili birinchi marta 1991 yilda gollandiyalik dasturchi Guido Van Rossum tomonidan e'lon qilingan. O'shandan beri bu til rivojlanishda uzoq yo'lni bosib o'tdi. 2000 yilda 2.0 versiyasi, 2008 yilda esa 3.0 versiyasi nashr etildi. Versiyalar orasidagi katta farqlarga qaramay, subversiyalar doimiy ravishda chiqariladi. Shunday qilib, ushbu materialni yozish paytidagi joriy versiya 2022 yil oktyabr oyida chiqarilgan 3.11 .

Python juda oddiy dasturlash tili bo'lib, u ixcham va ayni paytda juda sodda va tushunarli sintaksisga ega. Shunga ko'ra, uni o'rganish oson va aslida bu o'rganish uchun eng mashhur dasturlash tillaridan biri bo'lishining sabablaridan biridir. Xususan, 2014-yilda u AQShda o'rganish uchun eng mashhur dasturlash tili sifatida e'tirof etilgan.

Python nafaqat ta'lim sohasida, balki maxsus dasturlarni, shu jumladan tijorat dasturlarini yozishda ham mashhur. Shuning uchun biz foydalanishimiz mumkin bo'lgan ushbu til uchun ko'plab kutubxonalar yozilgan.

Bundan tashqari, ushbu dasturlash tilida dasturchilarning juda katta jamoasi mavjud, Internetda siz ushbu til bo'yicha juda ko'p foydali materiallar va misollarni topishingiz va mutaxassislardan malakali yordam olishingiz mumkin.

Python dasturlash tilining asosiy xususiyatlari:

- Skript til ekanligi. Dastur kodi skriptlar shaklida yoziladi (*.py fayllar);
- Turli xil dasturlash paradigmalarni qo'llashi, shu jumladan obyektga yo'naltirilgan va funktsional paradigmalarni qo'llab-quvvatlaydi;
- Interpretator dasturlash tili ekanligi. Skriptlar bilan ishlash uchun skriptni ishga tushiradigan va bajaradigan interpretator kerak. Bunday dasturlarni virtual mashina deb ataladi;
- Portativlik va platforma mustaqilligi. Bizda qanday operatsion tizim mavjudligi muhim emas - Windows, Mac OS, Linux, biz faqat interpretator bo'lsa, ushbu barcha operatsion tizimlarda bizning dastur skriptimiz ishlaydi;
- Avtomatik xotirani boshqarish tizimi, Garbage collector (GC);

- Dinamik ma'lumot turli dasturlash tili hisoblanadi.

Python dasturining bajarilishi quyidagicha (1-rasm):

- 1) Biror matn muharririda berilgan dasturlash tilida ko'rsatmalar to'plamiga ega dastur yozamiz. (Текстовый редактор – matn muharriri);
- 2) Matn muharririda yozilgan dasturni instruksiyalarini *.py faylga saqlaymiz.(Файл с исходным кодом, скрипт – birlamchi kod fayli, skript);
- 3) Biz ushbu skriptni bajarish uchun interpretatorga topshiramiz. (Интерпретатор Python – Python interpretatori);
- 4) Tarjimon kodni oraliq baytkodga tarjima qiladi. (Байткод – baytkod);
- 5) Tayyor baytkodni virtual mashina joriy operatsion tizim tomonidan bajariladigan ko'rsatmalar to'plamiga aylantiradi. (Виртуальная машина Python – Python virtual mashinasi);
- 6) Tayyor mashina ko'rsatmalariga joriy operatsion tizim bajaradi. (Операционная система – operatsion tizim);



1-rasm. Python dasturlash tili ishlash prinsipi.

Bu yerda shuni ta'kidlash joizki, rasmiy ravishda interpretator tomonidan manba kodini baytekodga tarjima qilish va virtual mashina tomonidan baytekodni mashina ko'rsatmalari to'plamiga tarjima qilish ikki xil jarayonni ifodalasa ham, aslida ular interpretatorning o'zida birlashtirilgan. Ushbu ikkala vazifani Python dasturini o'zi bajaradi.

Python dasturlash tili qo'llanilish sohalari va kutubxonalari:

Python interpretatori ushbu tilda ilovalar yaratish imkonini beruvchi yetarli funktsionallik bilan birga keladi. Biroq, bu funktsiya bir qator vazifalar uchun yetarli bo'lmashligi mumkin. Ammo butun dunyo bo'ylab tilni dasturchining katta hamjamiyati tufayli turli maqsadlar uchun ishlatilishi mumkin bo'lgan turli xil paketlar va kutubxonalarning katta ekotizimlari mavjud. Ularni quyidagi sohalar keng qo'llaniladi:

- 1) Grafik ilovalar yaratish uchun:
 - a) **Tkinter**- python dasturlash tili dasturi bilan keluvchi standart kutubxona;
 - b) **PyQt, PySide** – Dunyoda eng mashhur C++ tili uchun yozilgan Qt kutubxonasining Python dasturlash tili talqini;
 - c) **wxPython** – C++ tili uchun yozilgan wxWidget kutubxonasining Python dasturlash tili talqini;
- 2) Mobil ilovalar yaratish uchun:
 - a) **Kivi** – krossplatformali ilovalarni yaratish uchun freymwork;
 - b) **Toga**
- 3) Veb-ilovalarni yaratish uchun:
 - a) **Django** – Murakkab, yuqori trafikga ishlay oladigan web-saytlar yaratish uchun freymwork;
 - b) **Flask** – oddiy, tezkor web-saytlarni yaratish uchun ishlatiluvchi tezkor, yengil freymwork;
 - c) **FastAPI** – Server tomonida joylashgan ma'lumot bazasidan boshqa klient ilovalarga ma'lumotlarni uzatish uchun ishlatiluvchi kutubxona.
- 4) Matematik jarayonlarni avtomatlashtirish uchun:
 - a) **Numpy** – matematik formulalar, massivlar va matrissalar bilan tezkor ishlash kutubxonasi;
 - b) **Pandas** – ma'lumotlar tahlil va statistik ma'lumotlar bilan ishlash kutubxonasi;
 - c) **Matplotlib** – turli xil statistik grafiklarni ko'rsatish kutubxonasi;
- 5) Mashinani o'qitish, sun'iy intellekt uchun:
 - a) **SciPy**
 - b) **PyTorch**
 - c) **Tensorflow**
 - d) **OpenCV**
 - e) **Keras**
 - f) **NLTK**

Python dasturlash muhiti

Integrated development environment (Dasturlash muhit) – integrallashgan dasturlash muhiti (*qisqacha dasturlash muhiti*) deb yuritilib,

dasturchilarga dasturiy ta'minot kodini samarali ishlab chiqishda yordam beradigan dasturiy ta'minot hisoblanadi. Dasturiy ta'minot kodlarini tahrirlash, qurish, sinovdan o'tkazish kabi imkoniyatlarni birlashtirib, Dasturchilarni dasturiy ta'minot ishlab chiqish samaradorligini oshiradi. Yozuvchilar matn muharrirlaridan va buxgalterlar elektron jadvallardan foydalanganidek, dasturiy ta'minot dasturchii ham o'z ishlarini osonlashtirish uchun dasturlash muhitlaridan foydalanadilar.

Matn muharrirlaridan dasturlash muhitlarining farqlari:

Kod yozish uchun istalgan matn muharriridan foydalanishingiz mumkin. Biroq, ko'pchilik dasturlash muhitlari matnni tahrirlashda, tashqarida bo'lgan funktsionallikni o'z ichiga oladi. Ular dasturiy ta'minotni ishlab chiqish jarayonini ancha samaraliroq qilish uchun umumiy ishlab chiquvchi vositalari uchun markaziy interfeysni ta'minlaydi. Ishlab chiquvchilar turli xil dasturlarni qo'lda integratsiyalash va sozlash o'rniga tezda yangi ilovalarni dasturlashni boshlashlari mumkin. Shuningdek, ular barcha vositalarni o'rganishlari shart emas va buning o'rniga faqat bitta dasturga e'tibor qaratishlari mumkin.

Dasturlash tillarida quyidagi imkoniyatlar mavjud:

- **Kodni tahrirlashni avtomatlashtirish** (Code editing automation) - Dasturlash tillarida ko'rsatmalarni qanday tuzilishi kerakligi haqida qoidalar mavjud. Dasturlash muhiti ushbu qoidalarni bilganligi sababli, u manba kodini avtomatik ravishda yozish yoki tahrirlash uchun ko'plab aqlli xususiyatlarni o'z ichiga oladi;

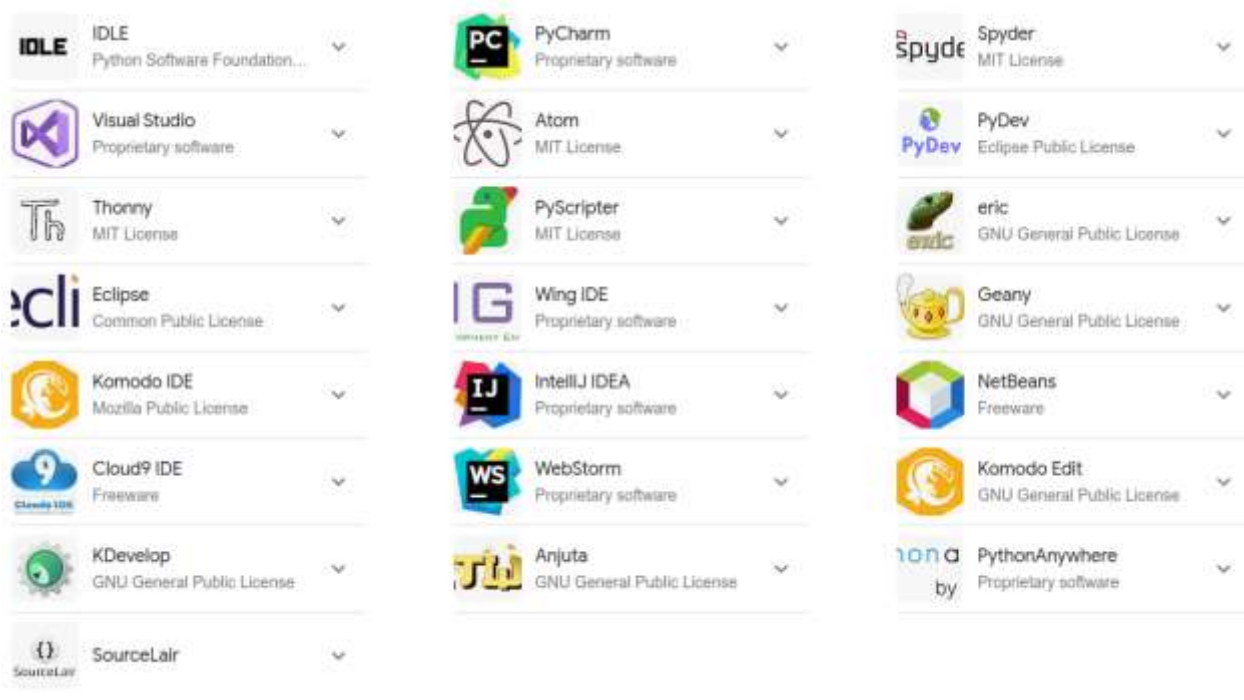
- **Sintaksisni formatlash** (Syntax highlighting) – Dasturlash muhiti yozma matnni avtomatik ravishda ba'zi so'zlarni qalin yoki kursiv qilish yoki turli shrift ranglarini qo'llash orqali formatlashi mumkin. Ushbu vizual signallar manba kodini yanada o'qilishini soddala qiladi va tasodifiy sintaksis xatolari haqida tezkor fikr bildiradi.

- **Intellektual kodni to'ldirish** (Intelligent code completion) – Qidiruv tizimida so'zlarni yozishni boshlaganingizda turli xil qidiruv so'zlari paydo bo'ladi. Xuddi shunday, Dasturlash muhitida dasturchi kod yozishni boshlaganida kod ko'rsatmasini bajarish uchun takliflar berishi mumkin.

- **Mahalliy qurilishni avtomatlashtirish** (Local build automation) – Dasturlash muhitlar odatda har bir kod o'zgarishining bir qismi bo'lgan takrorlanadigan ishlab chiqish vazifalarini bajarish orqali dasturchining mahsuldorligini oshiradi.

Shuningdek hozirgi kunda maxsus faqat dasturlash tilida yozish uchun qaratilgan matn muharrirlari ham yaratilmoqda ularni **Kod muharriri(Code editor)** deb nomlanib, dasturlash muhitlarining imkoniyatlarni o'z ichida faqat

qisman olgan dasturiy ta'minot hisoblanadi. (2-rasmda IDE va Code editor

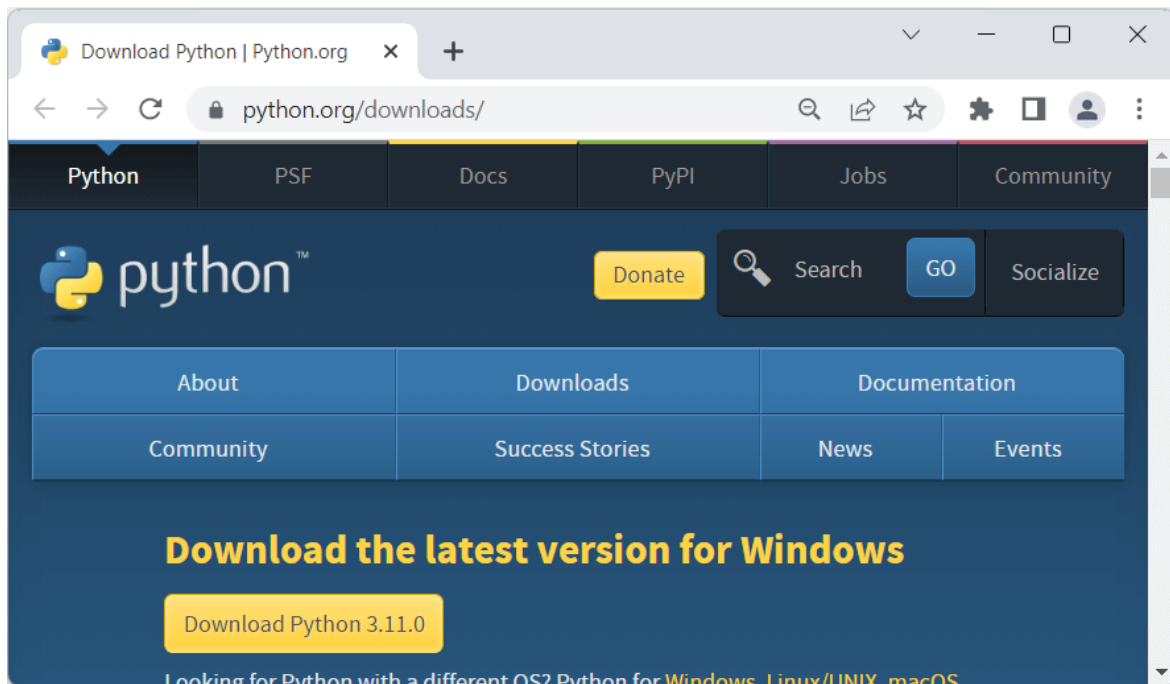


dasturlarini ko'rishingiz mumkin).

2-rasm. IDE va Code Editor dasturlar bir qismi.

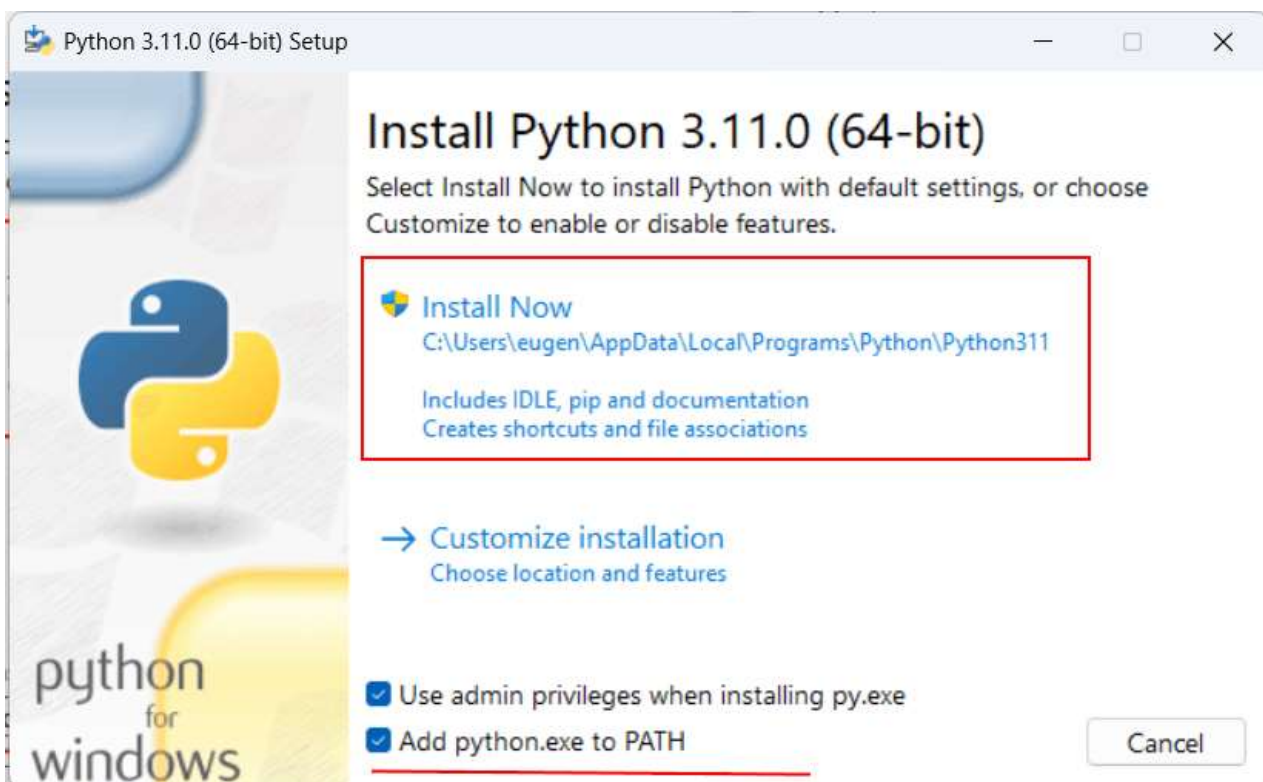
Python dasturlash muhiti

Pythonda dasturlar yaratish uchun bizga interpretator kerak. Uni o'rnatish uchun <https://www.python.org/downloads> sahifasiga o'ting va filning so'nggi versiyasini yuklab olish uchun havolani toping. (3-rasmda veb-sahifa ko'rsatilgan).



3-rasm. python.org web-sayti.

Tugmani bosish orqali joriy OT ga mos keladigan Python oʻrnatuvchisi yuklanadi. Eʼtibor bering, Windows 7 va undan oldingi versiyalari qoʻllab-quvvatlanmaydi. Windows OS da, oʻrnatuvchini ishga tushirganingizda, oʻrnatish ustasi oynasi ochiladi:



3-rasm. Python interpretator dasturini oʻrnatish jarayoni.

Bu erda biz interpretator oʻrnatiladigan yoʻlni belgilashimiz mumkin. Install Now tugmani tanlaymiz va dasturimiz **C:\Users\[foydalanuvchi nomi]\AppData\Local\Programs\Python\Python311** \ papkasiga oʻrnaydi.

Bundan tashqari, eng pastki qismida atrof-muhit o'zgaruvchilariga interpretatorga yo'l qo'shish uchun "Add python.exe to PATH" bayroqchani belgilashimiz kerak.

Shundan so'ng **Windows + R** klavishalarni bosing va o'chilgan buyruqlar oynasida **cmd.exe** deb yozib bajarish tugmasini yozing. Ochilgan qora oynada **python -version** buyrug'ini yozing.

Til alfaviti – Python dasturlash tili til alfaviti katta-kichik(A-Z, a-z) lotin harflari, raqamlar(0-9) va maxsus belgilardan iborat.

Identifikator – katta-kichik(A-Z, a-z) lotin harflari, raqamlar(0-9) va tag chiziq(_) belgilarining uzliksiz ketma-ketligidan iborat so'z tushuniladi. Identifikator deb dasturlash tilida keluvchi o'zgaruvchilar, obyektlar, funksiyalar nomlari hisoblanadi. Shuningdek katta-kichik harflar bilan yozilgan bir xil so'z ikki xil identifikator deb hisoblanadi. 2-rasmda berilgan **_ism, yosh, Yosh, funksiya, funksiya1** identifikator deb hisoblanadi.

Identifikatorni yozishda quyidagilar taqiqlanadi:

- Birinchi belgi raqam bilan boshlash;
- Ikkita bir xil nomlik o'zgaruvchi, obyekt yoki funksiya yozish;

Literal – Dastur kodida yozilgan yozilgan o'zgarmas skalar qiymalarni literallar deb ataladi. 2-rasmda berilgan **"Muhammadxon", 50, 100** qiymatlar letterallar deb hisoblanadi.

```

main.py - /home/atkif/projects/main.py (3.10.12)
File Edit Format Run Options Window Help
# Identifikatorlar
_ism = "Muhammadxon"
yosh = 50
Yosh = 100

def funksiya():
    pass

def funksiya2():
    pass
    
```

Ln: 3 Col: 9

2-rasm. Identifikator va literallarga misol.

Ko'rsatma, instruksiya – Python dasturlash tili bajaruvchi buyruq yoki buyruqlar guruhi bo'lib, aniq bir ishni yakunlanganligini bildiruvchi dastur kodi.

Kalit so'z – Python dasturlash tili uchun maxsus ajratilgan so'zlar bo'lib, maxsus dasturlash tilida yozilgan ishni bajaradi. Bunday so'zlarni identifikator sifatida foydalanish taqiqlanadi.

Python tilidagi kalit so'zlar						
and	as	assert	async	await	break	class
continue	del	elif	else	except	finally	false
for	from	global	if	import	in	
is	lambda	nonlocal	not	or	pass	
raise	return	true	try	while	with	yield

Operator – Python dasturlash tilida hisob-kitob ishini bajarib, natija qaytaruvchi amallar hisoblanadi. Operatorlar **unari, binar va trinar** turlarga bo'linadi. Operatorlarning bajarilish tartib kuchi bo'lib, eng yuqoridan boshlab eng pastiga qarab bajariladi. Shuningdek operatorlarni chapdan yoki o'ng bajarilish yo'nalishlariga ham bo'linadi.

Ifoda – Bir nechta operatorlar, funksiyalar ketma-ketligidan iborat ko'rsatma hisoblanadi.

Izoh – Dasturlash tiliga hech qanday aloqasi yo'q, kodda yozilgan matnlar hisoblanadi.

Sintaksis – Dasturlash tilida buyruq, ko'rsatma, operatorlarni yozish qoidasi.

Butun, haqiqiy va kompleks sonlar.

Berilganlarni qayta ishlash, arifmetik amallarni bajarish uchun Python tilida butun, haqiqiy va kompleks sonlardan foydalaniladi. Python tilining kutubxonasida berilganlarning bir qator son tiplari bilan ishlashga mo'ljallangan metodlar mavjud.

Butun sonlar. Python dasturlash tilida odatdagidek yoziladi. Ular yordamida odatdagi matematik amallar (*qo'shish, ayirish, ko'paytirish, bo'lish, darajaga ko'tarish, sonning ishorasini o'zgartirish, sonning moduli, qoldiq, to'liqsiz bo'linma va b.*) bajariladi.

Haqiqiy sonlar. Haqiqiy sonlarning butun va kasr qismi nuqta orqali ajratib yoziladi. Juda katta va juda kichik sonlarni yozishda "qozg'almas nuqtali" va "suzuvchi nuqtali" yozuvdan foydalaniladi. Bunda holda sonlar

ikki qismga ajratilib, birinchi qismi o'nli kasr ko'rinishida bo'lib sonning **mantissasi** deyiladi. Ikkinchi qismi o'nning butun darajalari ko'rinishida ifodalaniib, **sonning tartibi** deyiladi. Bu qismlar bir-biriga ko'paytirish belgisi orqali yoziladi. Masalan, yerning massasi $5.9742 * 10^{24}$, 1 gramm suvdagi molekulalar soni $3.7 * 10^{22}$ taga, molekula massasi $2.99 * 10^{-23}$ grammga teng.

Python tilida sonlarning suzuvchi nuqtali ko'rinishi quyidagicha yoziladi: avvalo sonning mantissasi, so'ngra *e* harfi, undan so'ng sonning tartibi yoziladi. Yuqorida ko'rsatilgan sonlar 5.9742e24, 3.7e22 va 2.99e – 23 ko'rinishida yoziladi.

Kompleks sonlar. Matematika kursida kompleks sonlar bilan ish ko'riladi. $a + bi$ ko'rinishidagi sonlar kompleks sonlar deb yuritiladi. Bunda *a* kompleks sonning haqiqiy qismi, *b* esa mavxum qismi deyiladi. Python dasturlash tilida kompleks sonlar ustida bevosita amallar bajarish imkoni berilgan.

Kompleks sonlar bilan ishlashda `complex(a, b)` funksiyasidan foydalaniladi. Bunda birinchi argument sonning haqiqiy qismi ikkinchisi mavxum qismi bo'ladi.

O'zgaruvchilar. O'zlashtirish operatori.

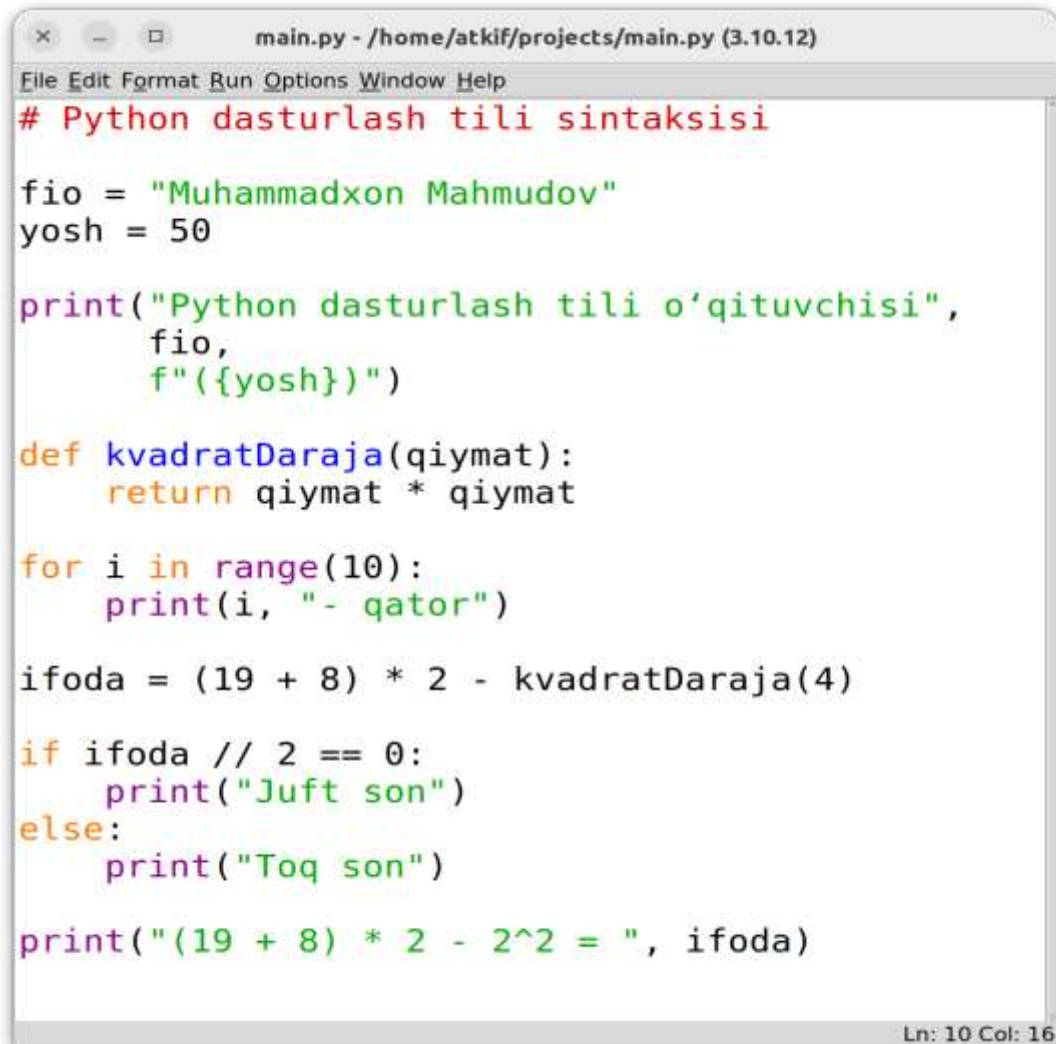
Pythondagi o'zgaruvchilar C++ yoki Delphi dasturlash tillaridagi o'zgaruvchilardan tubdan farq qiladi. Python'da o'zgaruvchilar o'rniga "nom"lar, boshqacha aytganda "ism"lar atamasidan ham foydalaniladi.

Dasturlash tillarini o'zgaruvchilarsiz tasavvur qilib bo'lmaydi. Bu xuddi insonlarning familiyasi yoki ismisiz ularga murojaat qilib bo'lmasligiga o'xshatish mumkin. Dasturlash tilida sonlarni, matnlarni, belgilarni, sonlar ketma-ketligi ro'yxatini ifodalash uchun o'zgaruvchilardan foydalaniladi. O'zgaruvchilar deb ma'lumotlarni saqlash uchun xotiraning nomlangan qismiga aytiladi.

O'zgaruvchilarni ifodalashda lotin alifbosining katta (A-Z) va kichik (a-z) harflaridan, raqamlardan (0-9), tag chizig'i (_) belgilar ketma-ketligidan iborat (*kiril alifbosining harflaridan ham foydalanish mumkin*). Ketma-ketlikning birinchi belgisi lotin alifbosining harflaridan so'ngra harf yoki raqamlardan tashkil topadi. Masalan, **a, ism, yusa123, maktab_64** kabi yozish mumkin. Nomlayotganda ayrim xatoliklarga yo'l qo'yilishi mumkin, bular: **1son** yoki **1_son**. Bunday xato qilmaslik uchun doim nomlanayotgan o'zgaruvchi nomi oldidan harf yoki '_' belgisi foydalaniladi.

O'zgaruvchilarga qiymat berishda o'zlashtirish operatori teng(=) belgisidan foydalaniladi. Qiymat berish operatori, ya'ni teng belgisining chap tomonida **o'zgaruvchi** o'ng tomoniga **qiymat** (*son, arifmetik ifoda, boshqa o'zgaruvchi nomi, funksiya yozilishi mumkin*). Masalan, $x = 5$, $yuza = s$, $y = z$, `ism = "Xasanboy"` va boshqalar.

O'zgaruvchilar faqat raqamlardan iborat bo'lishi mumkin emas. Ularning uzunligi uchun chegara qo'yilmagan. O'zgaruvchilarni ifodalashda xizmatchi so'zlardan foydalanish tavsiya qilinmaydi.



```
main.py - /home/atkif/projects/main.py (3.10.12)
File Edit Format Run Options Window Help
# Python dasturlash tili sintaksisi

fio = "Muhammadxon Mahmudov"
yosh = 50

print("Python dasturlash tili o'qituvchisi",
      fio,
      f"({yosh})")

def kvadratDaraja(qiymat):
    return qiymat * qiymat

for i in range(10):
    print(i, "- qator")

ifoda = (19 + 8) * 2 - kvadratDaraja(4)

if ifoda // 2 == 0:
    print("Juft son")
else:
    print("Toq son")

print("(19 + 8) * 2 - 2^2 = ", ifoda)

Ln: 10 Col: 16
```

3-rasm. Python dasturlash tili sintaksisiga misol.

FOYDALANILGAN ADABIYOTLAR

1. Mark Lutz (Марк Лутц), "Learning Python" ("Изучаем python"), 3rd edition, 2009, 845 p.
2. Mark Lutz (Марк Лутц), "Programming Python" ("Программирование на python" (том I, II)), 4th edition, 2011, 993 p.