**MA’RUZA MASHG‘ULOTI UCHUN O‘QUV MATERIALLARI**

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

**1-mashg‘ulot.** Python dasturlash tilining klassifikatsiyasi va rivojlanish tarixi. Python dasturlash tilining asosiy tushunchalari.

**O‘quv savollari:**

1. Fanning mazmuni, maqsadi, vazifalari;
2. Pythonni o‘rnatish. PyCharm dasturini o‘rnatish;
3. Pythonda “Hello world!” dasturini tuzish.
4. Python tilining asosiy operatorlari bilan tanishish

**1. O‘quv fanining maqsad va vazifalari .**

“Python dasturlash tili” fanining maqsadi kursantlarga dasturlash texnologiyalari haqida tushuncha berish va O‘zbekiston Respublikasi Qurolli Kuchlari tizimida qo‘llaniladigan dasturlash tillarini o‘rgatish, ularga operatsion tizim tomonidan qo‘llaniladigan tayyor dasturiy ta’minotni yaratish uchun zarur bo‘lgan dasturiy vositalardan foydalanishni o‘rgatish.

Kredit-moduli tizimi asosida fan bo‘yicha o‘quv kursi ma’ruza, amaliy mashg‘ulotlar, shuningdek, mavzu bo‘yicha topshiriq va mustaqil topshiriqlarni o‘z ichiga oladi. Ma’ruza o‘quv materiali, amaliy ishlar ko‘rsatilgan mavzular bo‘yicha nazariy va amaliy ma’lumotlar beradi, amaliy ish, mustaqil ish va ishlarni bajarish tartibini tushuntiradi. Kursda belgilangan o‘quv materiali kursantlar tomonidan mustaqil ravishda o‘rganiladi, testlar, amaliy ishlar individual ravishda amalga oshiriladi.

Kurs 1 semestrga mo‘ljallangan bo‘lib, umumiy auditoriya 60 soatni tashkil qiladi. 7 Semestrda, 6 soat nazariya, 54 soat amaliyot, jami 60 soat auditoriya soati mavjud.

Sinfdagi yuklamadan tashqari mustaqil ishlarga ham vaqt ajratiladi. Shunga ko‘ra, 6-semestrda 60 soat mustaqil ish.

Mustaqil tayyorgarlik mashg‘ulotlatlarida kursantdan mustaqil ravishda turli xil dasturlarni yaratishni va yaratgan dasturi haqida gapirib berishi talab qilinadi.

**Kursantlarni baholash**

Kursantlarning bilimini baholash semestr davomida va yakuniy nazorat jarayonida o‘quv materialini o‘zlashtirish ko‘rsatkichi asosida amalga oshiriladi.

7-semestr davomida “Python dasturlash tili” fanini o‘rganishda kursantlar 100 ballik tizim bo‘yicha baholanadi. Shundan joriy va oraliq nazoratga 20 ball, yakuniy nazoratga esa 40 ball beriladi. Joriy va oraliq nazorat ballarining umumiy natijasi 30 balldan past boʻlgan kursantlar yakuniy nazorat imtihoniga qoʻyilmaydi. Yakuniy nazoratga kiritilgan, unda 30 va undan ortiq ball to‘plagan kursantlar o‘tgan semestrda fanni o‘zlashtirgan deb hisoblanadi.

Joriy, oraliq va yakuniy nazorat ballari quyidagicha taqsimlanadi:

|  |  |
| --- | --- |
| Joriy nazorat | 40 ball |
| Oraliq nazorat | 20 ball |
| Yakuniy nazorat | 40 ball |
| Fan bo‘yicha jami: | 100 ball |

**Python dasturlash tili va uning rivojlanish tarixi.**

**Python dasturlash tili.** Python yuqori darajadagi dasturlash tili boʻlib, u AT ning turli sohalarida, masalan, mashinani oʻrganish, turli ilovalarni ishlab chiqish, web illovalari ishlab chiqish, tahlil qilish va boshqalarda keng qoʻllaniladi.

2019-yilda Python Java tilini 10 foizga ortda qoldirib, eng ommabop dasturlash tiliga aylandi. Bu juda ko‘p sabablarga bog‘liq bo‘lib, ulardan biri malakali mutaxassislarga to‘lanadigan ish haqining yuqoriligidir (yiliga taxminan 100 ming dollar).

Hozirgi kunda Turli xil dasturlash tillari mavjud bo‘lib, odatda ular ma’lum bir sohada (yoki bir nechta) liderlik qilishadi. Python dasturlash tili bo‘lsa, turli xil sohalarda ilovalar yaratish imkomiyatiga egaligi, dasturchilarni o‘ziga jalb qilmasdan qolmaydi.

Web ilovalar ishlab chiqish bozorining kichik qismini Python tili yaratilgan web ilovalari egallagan. Python tilini desktop ilovalarini yozish uchun ham ishlatilsa, mashinani o‘rganish (machine learning) sohasida to‘liq yetakchilik qiladi.

**Python nomining kelib chiqish tarixi.**

Gido van Rossum python dasturlash tilini yaratganidan keyin, uni nomlamasdan turib tarqatishni hohlamadi. Ammo mukammal nom tanlash uchun vaqt sarflash Gvido uchun vaqtni behuda o‘tkazish hisoblanardi. Shunda uning xayoliga kelgan birinchi fikr “Python” bo‘ldi va dasturlash tilining nomini “Python”deb nomlab qo‘yaqoldi. Chunki 1970-yillarning boshida mashhur bo‘lgan, “Monty Python’s Flying Circus” komediya seryali uning sevimli shoularidan biri edi.

Yangi til nomi bilan bir qatorda logotip o‘ylab topish kerak edi va Guido tasodifiy shrift tanladi va “Python” so‘zini shu shriftda yozdi. Gvido bu nomning ilonlarga aloqasi yo‘qligini qayta-qayta ta’kidlagan bo‘lsada, ommaning fikriga ta’sir qilishning ilojisi yo‘q edi.

Ko‘pchilik dasturchilar Python tilini ilonlar bilan bog‘lab, uning nomini emas turli turdagi ilonlar rasmlari bilan kitoblar chop qilishar edi. Bu holat 2006 yilgacha davom etdi.

**Payton** (python) **yoki Piton** (питон) **– bu qanday talaffuz qilinadi?**

Bu ingliz teleko‘rsatuvining nomi bo‘ladimi yoki “ilon” so‘zining inglizcha tarjimasi bo‘ladimi,  “Python” nomi Pyton kabi to‘g‘ri talaffuz qilinadi . Biroq, rus dasturchilarining taxminan 80% “Piton” (питон)deb talaffuz qilishga o‘rganib qolishgan.

 Ushbu variantlardan birini aniq to‘g‘ri ikkinchisini xato deb aytish qiyin. Biroq, “Piton” deb talaffuz qilish varianti faqat rus tilida so‘zlashuvchi suhbatdoshlar bilan suhbatda foydalanish uchun mos keladi, chunki har qanday xalqaro konferentsiyada “Piton” so‘zi tushunarsiz bo‘ladi. Chunki ingliz tilida bunday so‘z mavjud emas.

Bunday konferensiyalarda faqat “Python (Python)” deb talaffuz qilish kerak bo‘ladi.

**Logotip.** Logotipda ikkita ilon kvadrat shaklda tasvirlangan, bu esa ko‘pincha foydalanuvchilarni til nomini sudralib yuruvchi bilan bog‘lashga undaydi.



*1-rasm. Python dasturlash tili logotipi*

Bu logotip muallifning ukasi, dasturchi va tip dizayneri Joost van Rossum tomonidan yaratilgan. U logotip dizaynini ishlab chiqshda kvadrat shaklini egallagan ikkita ilon va Flux Regular matn shriftida yozilgan Python so‘zidan foydalandi.

**Yaratilish tarixi.** Til 1980-yillarning oxirida dasturchi Guido van Rossum tomonidan ishlab chiqilgan. O‘sha paytda u Niderlandiyadagi matematika va informatika markazida ishlayotgan edi.

Gvido van Rossum maktab yillaridanoq apparat vositalari bilan ishlashni yaxshi ko‘rardi va u tengdoshlari tomonidan qo‘llab-quvvatlanmasa ham, ma’qullamasa ham, bu uning mustaqil ravishda dasturlash tilini rivojlantirishiga to‘sqinlik qilmadi.

Rossum bo‘sh vaqtlarida Pythonda ishlagan va u bir paytlar rivojlanishiga yordam bergan ABC dasturlash tiliga asoslanib ishlagan.

**Python dasturlash tili tarixidagi bosqichlar** :

* **1991-yilning fevralda** python tilining kodlari *alt.sources* saytida chop etildi.  Til ob’ektga yo‘naltirilgan yondashuvga amal qildi va obyekt, class, metodlar, merosxorlik, funktsiyalar, istisnolarni qayta ishlash va barcha asosiy ma’lumotlar tuzilmalari bilan ishlash imkoniyatiga ega edi.
* **2000 yilda Pythonning ikkinchi versiyasi chiqdi**. Unga ko‘plab muhim vositalar, jumladan Unicode yordami va qoldiqlarni yig‘uvchi funksiyasi qo‘shildi.
* **2008-yil 3-dekabrda Pythonning uchinchi versiyasi ishlab chiqildi.** Bu versiyasi hozircha oxirgi versiya hisoblanadi. Bu versiyada tilning ko‘pgina xususiyatlari qayta ishlangan va oldingi versiyalar bilan mos kelmaydigan holga keltirildi. Uchinchi versiyaning funksionalligi ikkinchisidan kam bo‘lmasada, tilning rivojlanishi ikki tarmoqqa bo‘lingan. Kimdir eski loyihalarni qo‘llab-quvvatlash uchun Pythonning 2-versiyasida foydalanishni davom ettirsa, kimdir butunlay uchinchi versiyada ishlashni tanladi.

Ikkinchi versiyaning tugatilish vaqti 2015 yilga belgilangan edi. Biroq barcha mavjud kodlarni Python 3 ga o‘tkazishga ulgurmaslikdan qo‘rqib,  Python 2 ning ishlash muddati 2020 yilgacha uzaytirilgan.

Pythonning sintaksisi uni har doim boshqa dasturlash tillaridan ajratib turadi. U ortiqcha operatorlardan xalos qilingan. Oddiy ingliz tili bilan sintaksisning o‘xshashligi hatto oddiy foydalanuvchiga ham kodni tushunishga imkon beradi. Bundan tashqari, dasturchi kamroq kod yozadi, chunki ortiqcha belgilarni ishlatishning hojati bo‘lmaydi. Masalan ; { }….

Pythonning oddiyligi qisman tilning ABC tiliga asoslanganligi bilan bog‘liq bo‘lib, u dasturlashni va dasturchi bo‘lmaganlarning kundalik ishlarini o‘rgatish uchun ishlatilgan.

**Python kod yozishni soddalashtiradi va rivojlanishni tezlashtiradi, chunki u quyidagi xususiyatlarga ega:**

* **Dinamik tip.** Dasturchi o‘zgaruvchilar tipini ko‘rsatishi shart emas, tilning o‘zi o‘zgaruvchiga qanday ma’lumot yuklanganiga qarab tipini aniqlab oladi.
* **Funktsiya orqali bir nechta qiymatlarni qaytarish.** Ushbu qiymatlarni vergul bilan ajratilgan holda to‘rtburchak qavslar ichiga yoziladi va avtomatik ravishda ro‘yxatga aylantiriladi. Funktsiyadan [massivni](https://all-python.ru/osnovy/massivy.html) qaytarish uchun “return ro‘yxat\_nomi” yozish kifoya.
* **Avtomatik xotiradan joy ajratish.** Dasturchi o‘zgaruvchilarga xotiradan joy ajratish uchun ortiqcha kod yozishi talab qilinmaydi. Bu, bir tomondan, dasturchining dastur ustidan nazoratini kamaytiradi, ikkinchi tomondan, rivojlanish sezilarli darajada tezlashadi.
* **Chiqindilarni yig‘uvchi**. Agar ob’ekt keraksiz bo‘lib qolsa, u chiqindi yig‘uvchi tomonidan avtomatik ravishda o‘chiriladi. Chiqindi yig‘uvchi sizga xotiradan foydalanishni optimallashtirish va keraksiz narsalarni qo‘lda o‘chirmaslik imkonini beradi.
* **a, b = b, a**  **ifodasi**. Bu ifoda o‘zgaruvchilarning qiymatlarini o‘zaro almashtiradi. Endi **a** o‘zgaruvchining qiymati **b** o‘zgaruvchisiga yuklandi va aksincha. Shunday qilib, siz nafaqat ikkita o‘zgaruvchining, balki uch va undan ortiq ob’yektlarning qiymatlarini mos ravishda bir-biriga almashtirishingiz mumkin. Faqat taraflardagi o‘zgaruvchilar soni teng bo‘lishi talab qilinadi.
* **O‘zgaruvchi tipining qiymatga bog‘liqligi**.  O‘zgaruvchining qiymati - bu uning turini va boshqa xususiyatlarini belgilaydigan atributlarga ega bo‘lgan qandaydir ob’ekt. O‘zgaruvchi esa bu ob’ektning nomidir. Ushbu yondashuv qat’iy tipdagi ta’riflarga bo‘lgan ehtiyojni yo‘qga chiqaradi va o‘zgaruvchilarga boshqa turdagi qiymatni qayta tayinlashni sezilarli darajada soddalashtirdi.
* **For tsikli** . Pythonda massivlar, ro‘yxatlar va boshqa konteynerlar bilan ishlash oson va qulay. Uning barcha elementlarini takrorlash zarur bo‘lganda, konstruktsiya quyidagicha ko‘rinadi: **for x in konteyner:** *(iteratsiya 0 dan oxirgi -1 elementgacha elementlarni x ga qaytaradi)*. Agar siz tsiklni aniq bir miqdorda takrorlanishini amalga oshirish uchun **range**() funksiyasidan foydalanish kerak bo‘ladi: **for x in range(1,9):** *(tsikl  1 dan 8 gacha bo‘lgan qiymatlarini x ga uzatadi)*.
* **Machine learning sohasidagi yetakchiligi.** Python dasturlash tili “Mashina o‘rganish” (Machine learning) sohasida yaqqol yetakchilik qilib kelmoqda. Ilm-fan bilan u yoki bu tarzda shug‘ullanadigan odamlar kod yozish kabi narsalarga ko‘p vaqt sarflamaslikni afzal ko‘radilar, shuning uchun Python ularga yuklangan vazifalarni amalga oshirish uchun juda mos keladi.

**Python ham soddalik, ham kuchli vositalarni o‘zaro birlashtiradi. U deyarli har qanday dasturning prototipini yaratish uchun ishlatilishi mumkin.**

**Kod misoli** :

def kattasi(a, b):

if a > b:

print(a, “katta”, b, “dan”)

else:

print(b, “kichik”, a, “dan”)

def max\_mass (massiv):

max = 0

for x in massiv:

if massiv[x-1] > max:

max = massiv[x]

return max

def 2mass(massiv):

massiv = massiv \* 2

return massiv

print(“Oddiy Python dasturi”)

a = [1,2,3,6,1,6]

kattasi (1,5)

r1 = max\_mass(a)

r2 = 2mass (a)

print(“Massiv max elementi –”, r1)

print(“Ikkilangan massiv – ”, r2)

Dastur natijasi:

Oddiy Python dasturi

5 katta 1 dan

Massiv max elementi -  6

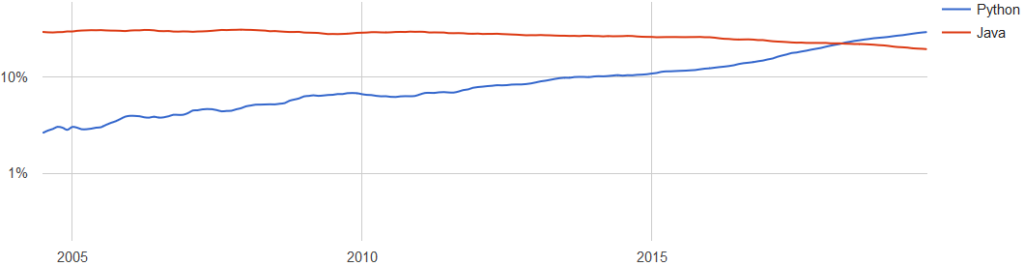
Ikkilangan massiv - [1, 2, 3, 6, 1, 6, 1, 2,3,6,1,6]

**Ommalashganligi.** Til 30 yoshdan oshgan bo‘lsada, butun dunyodagi dasturchilar orasida juda keng tarqalgan. Python deyarli har bir o‘rta yoki katta loyihada, asosiy ishlab chiqish vositasi sifatida bo‘lmasada, prototiplash yoki uning bir qismini yozish uchun vosita sifatida ishlatiladi.

U o‘z atrofida juda katta dasturchilar guruhini to‘pladi.

Stackoverflow saytida o‘tkazilgan so‘rov natijalariga ko‘ra, Python deyarli 39% ovoz bilan 7-o‘rinni egalladi.

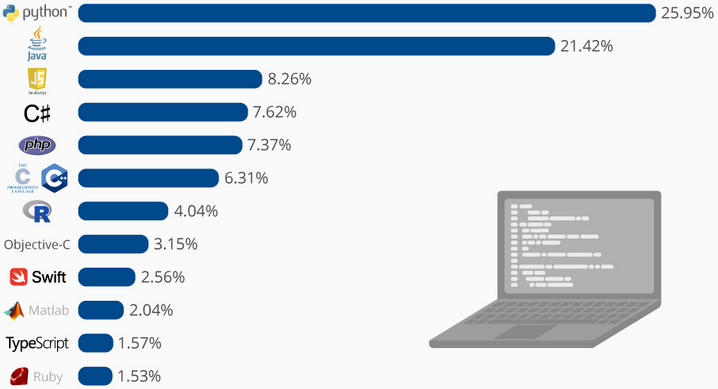
**PYPL.** Ushbu indeks tilni o‘rganish materiallari bilan bog‘liq qidiruv so‘rovlari soniga asoslanadi.



*2-rasm. PYPl indekmsi bo‘yicha solishtirma sxemasi*

PYPL ma’lumotlariga ko‘ra, Python mashhurligi bo‘yicha Javadan 10% ga oldinda.

**statista.com.** Xizmat turli xil statistik ma’lumotlarni taqdim etadi, ular orasida dasturlash tillarining mashhurligi bo‘yicha statistikalar ham mavjud.



*3-rasm.* ***Statista.com*** *saytidagi solishtirish diagrammasi*

**Ish tezligi.** Dasturchilar ko‘pincha o‘zlariga savol berishadi: "Python-dan foydalanish ishlashning pasayishiga olib keladimi?". Batafsil tekshiruvsiz hech qanday xulosa chiqarmang.

Agar biz faqat kodni bajarish tezligini hisobga olsak, Python C kabi boshqa dasturlash tillaridan past ekanligi ayon bo‘ladi. Darhaqiqat, dinamik terish, izohlash va dasturchi ishini osonlashtiradigan boshqa xususiyatlar ishlashning pasayishiga olib keladi. Biroq, zamonaviy IT-da nafaqat dasturlarning tezligi, balki ularni ishlab chiqish tezligi ham muhimdir. Ishlab chiqish, sinovdan o‘tkazish, disk raskadrovka va qo‘llab-quvvatlash - bularning barchasi juda ko‘p pul talab qiladi. Agar Python dasturlarining tezligi bo‘yicha past bo‘lishi mukin, ammo rivojlanish tezligida bo‘yicha unga teng keladigani yo‘q.

**Dasturchilar Pythonda yaratilgan dasturlarni bajarilish tezligini oshirish uchun turli yo‘llardan foydalanadilar:**

* **C kodini o‘rnatish** . Ushbu texnikadan foydalanib, siz ishlashni sezilarli darajada yaxshilashingiz mumkin, odatda vaqt birligida ko‘p so‘rovlarni qayta ishlaydigan kod bo‘limlari C tilida yoziladi. Masalan, bitta ma’lumotlar bazasidan ma’lumotlarni qabul qiladigan, uni qayta ishlaydigan va boshqasiga yuboradigan funksiyalar, agar o‘tadigan ma’lumotlarning miqdori yetarlicha katta bo‘lsa, C tilida yozilgani afzal xisoblanasi.
* **ifodalarni optimallashtirish.** Python dasturlarining tezligi ifodalarning ishlash tezligiga juda bog‘liq. Quyidagi jadvalda tezroq va sekinroq ishlovshi ifodalar keltirilgan.

1-jadval

| **Tezroq** | **Sekinroq** |
| --- | --- |
| a, b = c, d | a = c; b = d |
| a < b < c | a < b and b < c |
| not not a | bool(a) |
| a = 5 | a = 2 + 3 |

* **Dasturni testlash uchun tayyor modullar.** Kodning qaysi bo‘limlari umumiy ish faoliyatini sezilarli darajada kamaytirishini aniqlash uchun dasturchi testlash uchun maxsus modullardan foydalanishi mumkin. Shunday qilib, bu modullar yordamida qaysi kodni optimallashtirish yoki C kodi bilan almashtirish kerakligini bilib olishi mumkin.
* **Tayyor asboblar.** Aksariyat vazifalar uchun samarali echimlar allaqachon ishlab chiqilgan. O‘z yechimingizni noldan yozishdan ko‘ra, ba’zi kutubxonaning tayyor, tuzatilgan kodini ishlatish yaxshiroqdir, bu 100% samarali bo‘lmaydi.

**Pythonda qanday dasturlar yozish mumkin?**

Python tilida turli sferalarda dasturlar tuzish mumkin.

**Back-end – saytning dasturiy qismi.** Saytning server tomonini rivojlantirish uchun turli fremworklar qo‘llaniladi: **Django** va **Flask**. Bu fremworklar Pythonni boshqa mashhur vositalar bilan raqobatlashadigan imkoniyatlarga ega server tomonidagi dasturlash tiliga aylantiradilar.

PHP tili server tomonidagi web ishlab chiqish bozorining ko‘p qismini nazorat qilsada, tobora ko‘proq dasturchilar Pythonda ishlab chiqishni afzal ko‘rishmoqda.

**Blokcheyn.** Blokcheyn - bu ketma-ket bloklar zanjiri bo‘lib, unda har bir blok ma’lumotni o‘z ichiga oladi va har doim oldingisiga ulanadi. Texnologiya har qanday sohada qo‘llanilishi mumkin. Ayniqsa moliya sektorida va bitcoin kriptovalyutasi sohasida mashhurdir.

Blokcheyn axborot xavfsizligi va ochiqligini o‘zida mujassamlashtiradi. U foydalanuvchiga dunyoning istalgan nuqtasidan ma’lumotlarga kirish imkonini beradi.

**Bot.** Bu ma’lum bir vaqtda yoki berilgan signalga javoban ba’zi harakatlarni avtomatik ravishda bajaradigan dasturdir. Botlar inson xatti harakatlarini bevosita taqlid qilishi mumkin. Shuning uchun ular ko‘pincha texnik yordam ko‘rsatishda (chat botlarida), Internetda ma’lumot qidirishda (qidiruv botlarida), virtual dunyoda odam yoki boshqa mavjudotning harakatlarini taqlid qilishda (kompyuter o‘yinlari) zarur bo‘ladi.

Python sizga tezkor xususiyatga ega va nisbatan aqlli botlarni yaratish imkonini beradi. Shuni tushunish kerakki, botlar 500 qatorli kodlardan iborat oddiy dastur emas. Biznes uchun bot yaratish buyurtmasi bir necha millionga tushishi mumkin. Narx insondan farqlash qiyin bo‘ladigan botni loyihalash juda qiyin ekanligi bilan bog‘liq. Muloqotning ko‘plab variantlarini ta’minlash, inson xatti-harakatlarining omillarini tahlil qilish va ularni dasturda amalga oshirish kerak. Oddiy qilib aytganda, faqat nol va birlarni tushunadigan mashinadan siz ibtidoiy "miya" qilishingiz kerak bo‘ladi.

**Malumotlar bazasi.** Ma’lumotlar bazasi umumiy xususiyatlar va maxsus qoidalarga muvofiq tizimlashtirilgan ma’lumotlardir. Har qanday yirik loyihada ma’lumotlar bazalaridan foydalaniladi. Ular foydalanuvchilar haqidagi ma’lumotlarni, dasturdagi o‘zgarishlarni va hokazolarni saqlaydi.

Ma’lumotlar bazasini boshqarish tizimi Pythonda yozilishi mumkin.

**Kengaytirilgan haqiqat (Дополненная реальность).** “To‘ldirilgan reallik” virtual texnologiyalar yordamida jismoniy dunyoni to‘ldiradi. Ya’ni, virtual ob’ektlar real muhitga proyeksiya qilinadi va oddiy jismoniy ob’ektlarning xususiyatlari va xatti harakatlariga taqlid qiladi.

“Kengaytirilgan haqiqat”ni turli 3D filmlarda guvohi bo‘lish mumkin. Haqiqiy dunyoda u, masalan, jangovar jangchilarda (maqsad tizimi) ishlatiladi.

“Kengaytirilgan haqiqat” ishi teglar bilan o‘zaro ta’sirga asoslangan. Elektron qurilma ma’lumot oladi va atrofdagi makonni tahlil qiladi, kompyuter ko‘rish yordamida u odamning oldida nimani ko‘rayotganini “Tushunadi”. Keyin qurilma real dunyoda “Virtual qatlam”ni qoplaydi.

“Kengaytirilgan haqiqat” sferasida yaratilgan mukammal dasturlar taxminan 7-10 ming AKSH dollarini tashkil qiladi. Ularni loyihalash va yozish oson emas, 3D-dizaynerlardan dasturchilargacha turli mutaxassislar ishlab chiqish jarayonida tinmasdan mehnat qilishadi.

Python – bunday sferadagi loyihalarni yaratish uchun ajoyib tanlov.

**BitTorrent mijozi.** BitTorrent - bu Internet orqali katta hajmdagi ma’lumotlarni tezlikda uzatish, qabul qilish imkonini beruvchi noyob texnologiya.

BitTorrentning 6-versiyasidan keyingi versiyalari butunlay Pythonda yozilgan. Keyinchalik u C++ da butunlay qayta yozilgan bo‘lsa-da, bu Pyton tilidan xuddi shunga o‘xshash vazifalarni bajarish uchun foydalanish mumkinligini anglatadi.

**Neyron tarmoq.** “Neyron tarmoq” tushunchasi dasturlashga biologiyadan kirib kelgan. Biologiyada neyron tarmoq bir-biriga bog‘langan neyronlar ketma-ketligidir. Dasturiy ravishda yaratilgan neyron tarmoqlar nafaqat ma’lumotni tahlil qilish va saqlash, balki uni xotira qayta ishlab chiqishga qodir.

Ular inson miyasi tomonidan bajariladigan hisob-kitoblarni talab qiladigan murakkab masalalarni hal qilish uchun ishlatiladi. Odatda, neyron tarmoqlar biror narsani xususiyatlari bo‘yicha tasniflash, bashorat qilish, masalan, fotosurat yoki videodan odamni tanib olish kabi funksiyalarni amalga oshirish uchun foydalaniladi.

**Python neyron tarmoqlarni rivojlantirishda yaqqol yetakchi hisoblanadi. Standart vositalardan tashqari, u *“Mashinani o‘rganish”* sohasida juda ko‘p kutubxonalar mavjud. Buning yordamida hatto katta va murakkab loyihalar nisbatan tez yozilishi mumkin.**

**Parser -** bu ma’lumotlarni yig‘ish va qayta ishlash uchun mo‘ljallangan dastur. Foydalanuvchi valyuta kursi kabi ma’lumotlarni tahlil qilishi yoki turli kompaniyalar aktsiyalaridagi o‘zgarishlarni kuzatishi va tahlil qilishi mumkin.

Parser turli xil tillarda yozilishi mumkin. Ya’ni parser dasturlar boshqa dasturlash tillari yordamida ham yaratish mumkin. Lekin parser (ma’lumot yig‘uvchi) dasturlarni python tilida boshqa dasturlash tillariga qaraganda tez va samarali yozish mumkin.

**O‘yin yaratish.** Pythonda katta o‘yinlar yaratilmaydi. U prototipni (demo versiyasini) ishlab chiqish yoki ba’zi bir qismini amalga oshirish uchun ishlatiladi (masalan, o‘yinning server qismidagi logik qismi).

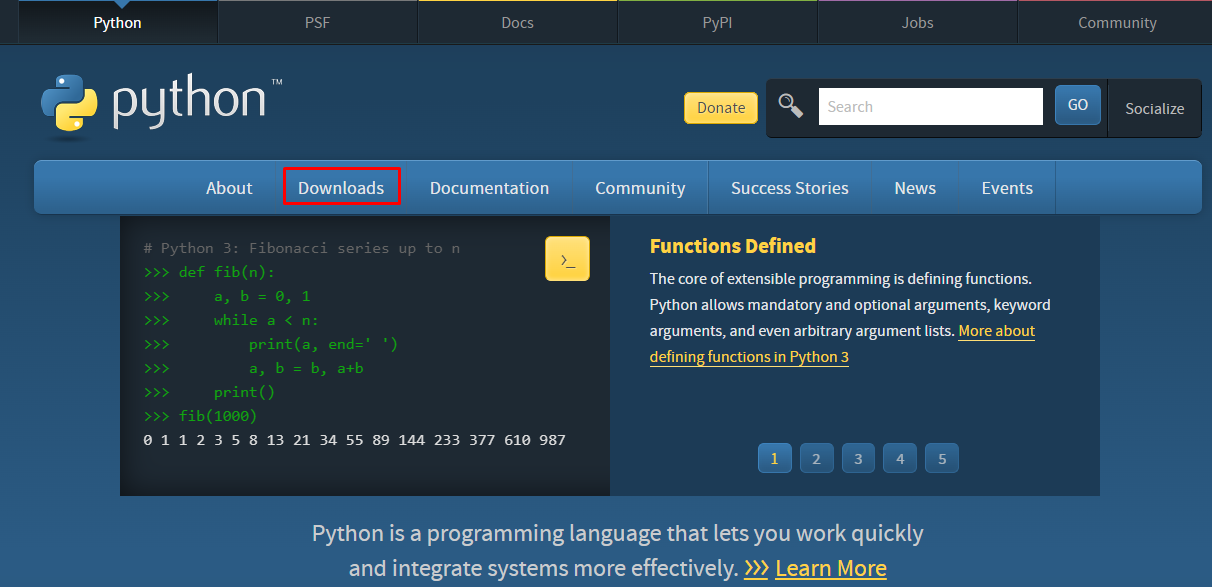
**Kichik loyihani yozish uchun siz kichik 2D o‘yinni yaratish uchun barcha kerakli vositalarni taqdim etadigan Pygame kutubxonasidan foydalanishingiz mumkin.**

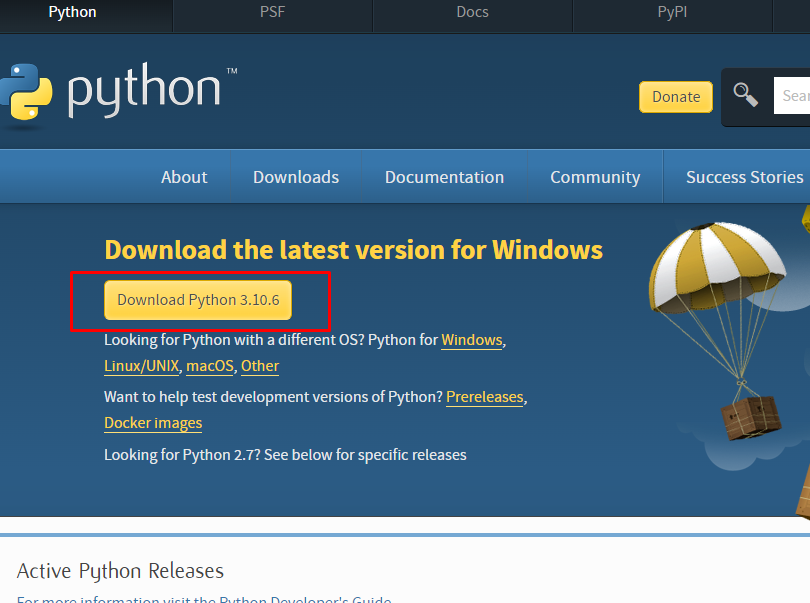
**2. Pythonni o‘rnatish. PyCharm dasturini o‘rnatish**

**PyCharm** - JetBrains tomonidan ishlab chiqilgan kross-platforma muharriri. Pycharm samarali Python rivojlanishi uchun zarur bo‘lgan barcha vositalarni taqdim etadi . Quyida Windows-da Python va PyCharm-ni o‘rnatish bo‘yicha batafsil ko‘rsatmalar mavjud.

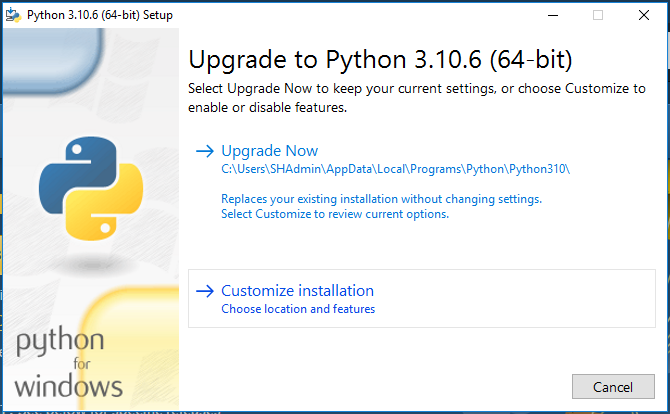
**Python o‘rnatilmoqda**

**1-qadam**. Python-ni yuklab olish va o‘rnatish uchun rasmiy Python veb-saytiga tashrif buyuring  //www.python.org/downloads/  va tegishli versiyani tanlang. Biz Python 3.10.6 versiyasini tanladik

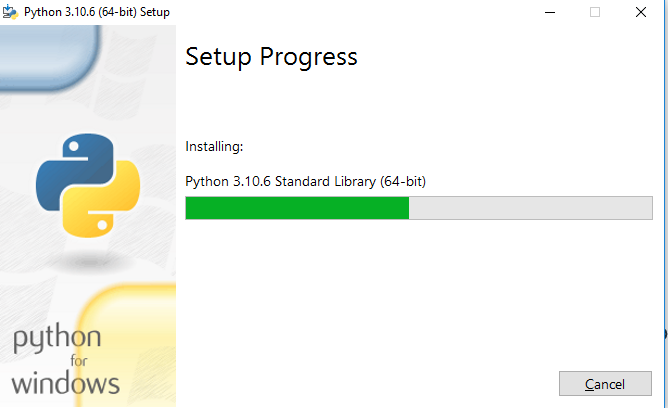




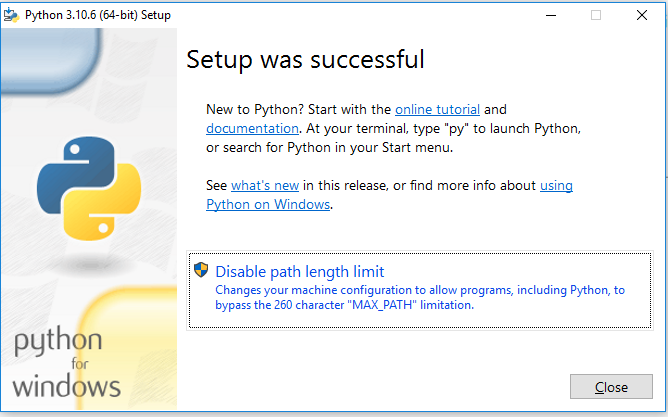
**2-qadam**  Yuklab olish tugallangach, Python-ni o‘rnatish uchun .exe faylini ishga tushiring. Keyin "Hozir o‘rnatish" tugmasini bosing.



**3-qadam**  Keyingi bosqichda Python o‘rnatilishi jarayonini ko‘rishingiz mumkin.

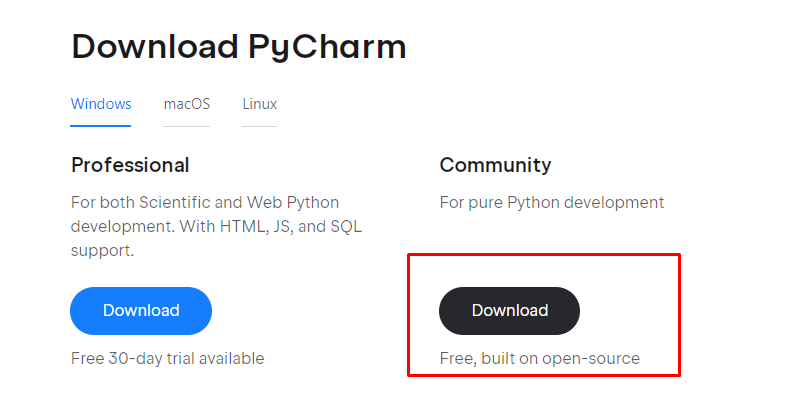


**4-qadam**  O‘rnatish tugallangach, o‘rnatish muvaffaqiyatli bo‘lganligi haqida panelni ko‘rasiz. Endi "Yopish" tugmasini bosing.

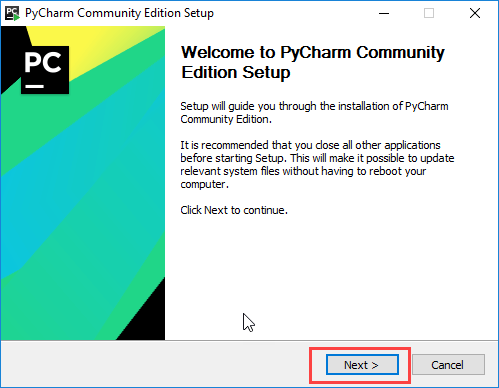


**Pycharm o‘rnatilmoqda**

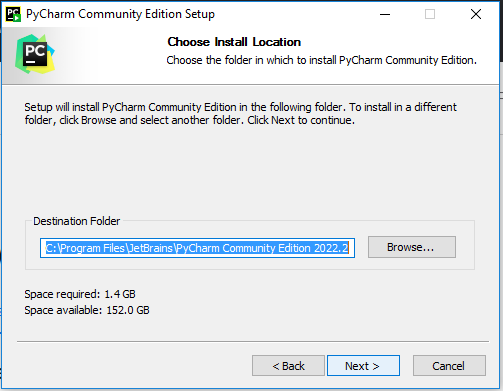
**1-qadam**  PyCharm-ni yuklab olish uchun //www.jetbrains.com/pycharm/download/ veb-saytiga tashrif buyuring va jamoat bo‘limidagi YUKLASH havolasini bosing.



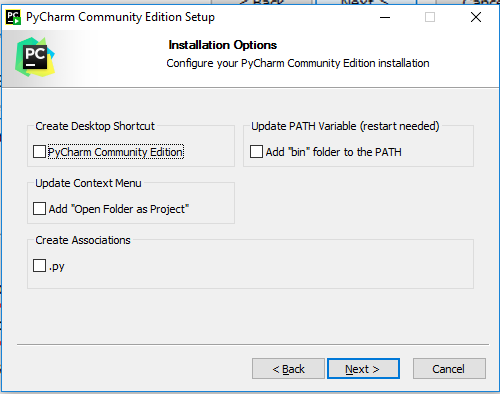
**2-qadam**  Yuklab olish tugallangach, PyCharm-ni o‘rnatish uchun .exe faylini ishga tushiring. O‘rnatish ustasi ishga tushishi kerak. "Keyingi" tugmasini bosing.



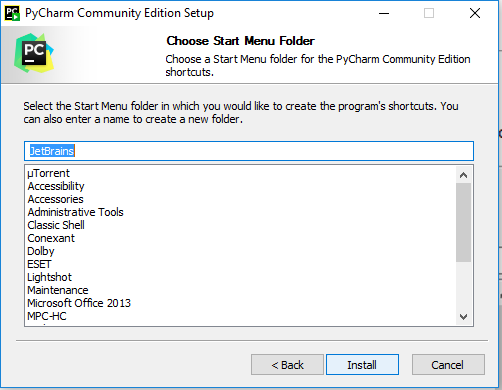
**3-qadam**  Keyingi bosqichda, agar kerak bo‘lsa, o‘rnatish yo‘lini o‘zgartiring. "Keyingi" tugmasini bosing.



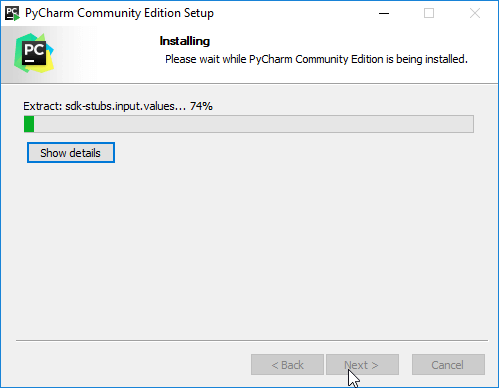
**4 -qadam)** Keyingi bosqichda, agar xohlasangiz, ish stolida yorliq yaratishingiz mumkin, so'ngra "Next" tugmasini bosing..



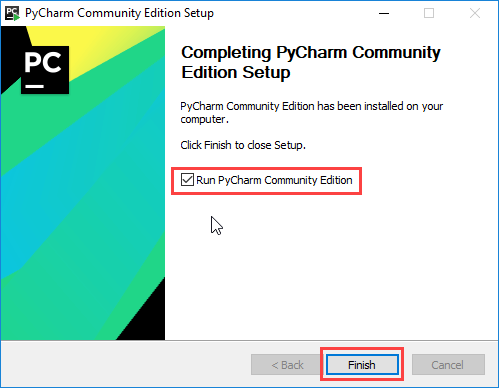
**5-qadam)**  Boshlash menyusi papkasini tanlang. JetBrains-ni sukut bo‘yicha qoldiring va "O‘rnatish" tugmasini bosing.



**6-qadam)**  O‘rnatish tugashini kuting.



**7-qadam)**  O‘rnatish tugallangandan so‘ng siz PyCharm o‘rnatilganligi haqida xabar olasiz. Agar siz uni ishga tushirishni xohlasangiz, avval "PyCharm Community Edition-ni ishga tushirish" katagiga belgi qo‘ying va "Finish" tugmasini bosing.

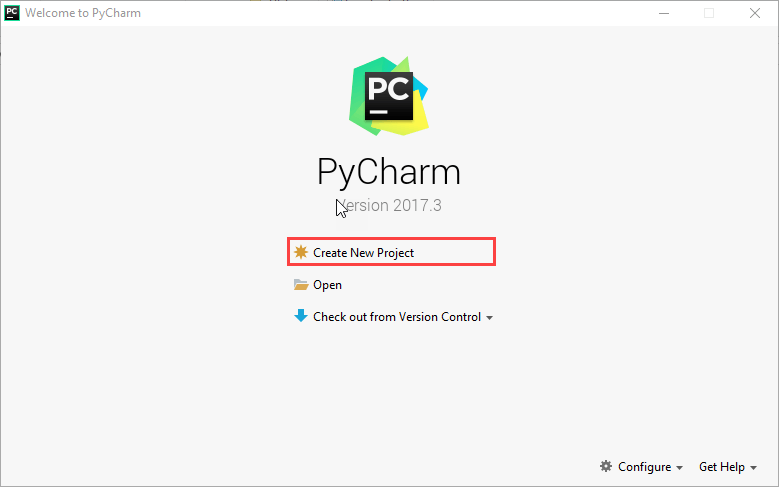


**8-qadam)**  Finish tugmasini bosganingizdan so‘ng quyidagi ekran paydo bo‘ladi.



**3. Pythonda “ Salom dunyo !” dasturini yaratish.**

**1-qadam)**  PyCharm muharririni oching. PyCharm xush kelibsiz panelini ko‘rasiz. Yangi loyiha yaratish uchun "Yangi loyiha yaratish" tugmasini bosing.

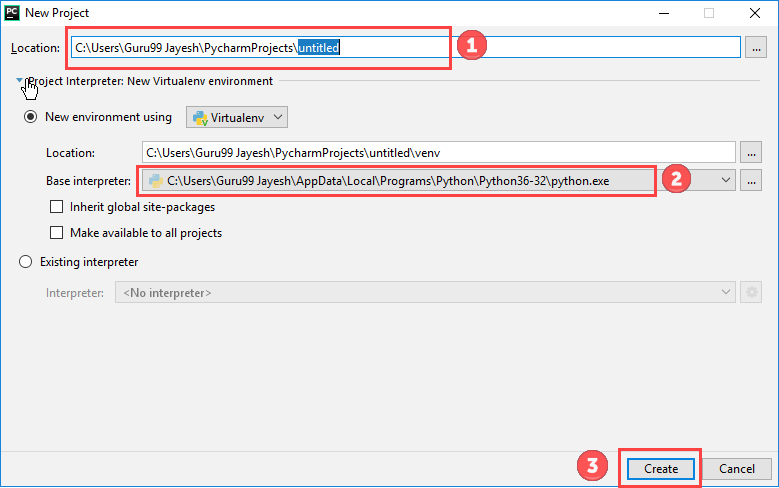


**2-qadam)**  Siz o‘rindiqni tanlashingiz kerak.

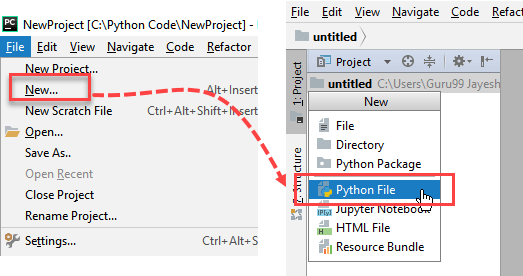
Siz loyihani yaratmoqchi bo‘lgan joyni tanlashingiz mumkin. Agar siz joylashuvni o‘zgartirishni xohlamasangiz, uni avvalgidek qoldiring, lekin hech bo‘lmaganda "nomsiz" nomini "Birinchi loyiha" kabi mazmunliroq narsaga o‘zgartiring.

PyCharm siz avval o‘rnatgan Python tarjimonini topgan bo‘lishi kerak edi.

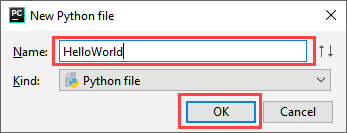
Keyin, "Yaratish" tugmasini bosing.



**3-qadam)**  Endi "Fayl" menyusiga o‘ting va "Yangi" ni tanlang. Keyin, "Python fayli" ni tanlang.



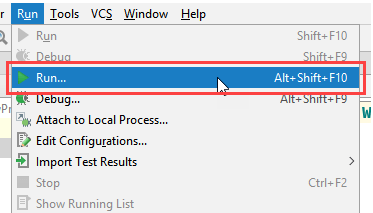
**4-qadam)**  Yangi qalqib chiquvchi oyna paydo bo‘ladi. Endi fayl nomini kiriting (bu erda biz "HelloWorld" ni o‘rnatamiz) va "OK" tugmasini bosing.



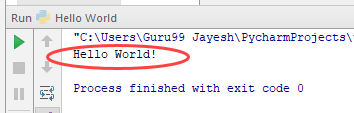
**5-qadam)**  Endi oddiy dastur yozing - print("Salom Dunyo!").



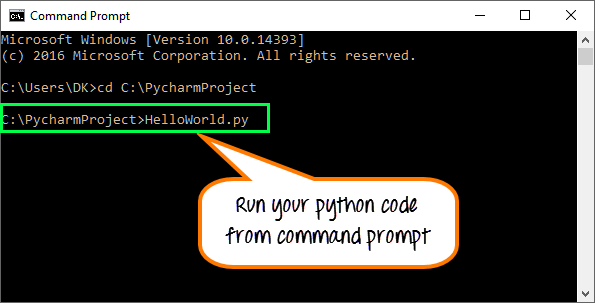
**6-qadam)**  Endi Run menyusiga o‘ting va dasturingizni ishga tushirish uchun Run-ni tanlang.



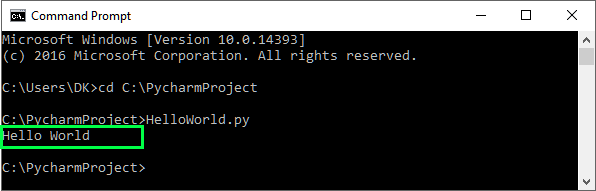
**7-qadam)**  Dasturingizning chiqishini ekranning pastki qismida ko‘rishingiz mumkin.



**8-qadam)**  Agar sizda Pycharm Editor o‘rnatilmagan bo‘lsa, tashvishlanmang, kodni buyruq satridan ishga tushirishingiz mumkin. Dasturni ishga tushirish uchun buyruq satriga to‘g‘ri fayl yo‘lini kiriting.



Kodning chiqishi quyidagicha bo‘ladi:



**9-qadam)**  Agar siz hali ham dasturni ishga tushira olmasangiz, bizda siz uchun Python muharriri mavjud. Iltimos, ushbu kodni   ishga tushiring.

**4. Python tilining asosiy operatorlari bilan tanishish**

Python dasturlash tilida yozilgan dastur ifodalar jamlanmasidan iborat. Har bir ifoda yangi qatorga joylashtiriladi. Masalan:

print(2 + 3)

print("Hello")

Python dasturlash tili kodida ifodalarning oldidan qoldirilgan oraliq masofalar katta ro‘l o‘ynaydi. Agar dasturchi tomonidan bilmasdan yoki chiroy uchun qoldirilgan har qanday oraliq masofa xatolik kelib chiqishiga sabab bo‘lishi mumkin. Masalan, garchi kod yuqoridagi bilan deyarli bir xil bo‘lsada, quyidagi holatda dasturda oraliq masofalar bilan bog‘liq xatolik sodir bo‘ladi:

print(2 + 3)

print("Hello")

Shuning uchun, python dasturlash tilida dastur kodining boshi satrning boshidan boshlanishi kerak. Bu Python bilan C# yoki Java kabi boshqa dasturlash tillari o‘rtasidagi muhim farqlardan biridir.

Ammo shuni yodda tutish kerakki, tilning ba’zi operatorlarining sintaksisi bir necha qatordan iborat bo‘ladi. Masalan, shart operatori if:

if 1 < 2:

print("salom")

Bu dasturda, agar 1 soni 2 dan kichik bo‘lsa, u holda “**salom**” so‘zi ekranga chiqariladi. Bu dasturda print(“Salom”) ifodasining oldida bo‘sh joy bo‘lishi kerak, chunki bu ifoda if shartli konstruktsiyasining bir qismi sifatida ishlatiladi.

Bundan tashqari, kodni yozish mobaynida qoldirib ketiladigan bo‘shliqlar 4 ga karrali probellar soniga (ya’ni 4, 8, 16 va boshqalar) teng bo‘lishi maqsadga muvofiqdir. Ammo 4 ga karrali bo‘lmay qolishi xatolik keltirib chiqarmaydi. Ifoda oldida qoldiriladigan masofalarning 4 ta probelga yoki 4ga karrali probellar soniga teng bo‘lishining sababi shundaki, 1 ta tab 4 ta probelga tengligida.

**Harflar katta-kichikligini farqlanishi**

**(Регистрозависимость)**

Python katta-kichik harflarni turli belgilan sifatida qabul qiladigan tildir. Shuning uchun **print** va **Print** yoki **PRINT** ifodalari turli ifodalarni anglatadi. Shunday ekan agar konsolga “Salom dunyo” xabarini Print funksiyasi orqali chiqarmoqchi bo‘lsak, hech qanday narsa chiqara olmaymiz, aksincha xatolik sodir bo‘ladi:

Print("Hello World")

**Izohlar**

Izohlar – kodning ma’lum bir qismida nima amalga oshirilayotganligini ifodalab foydalanuvchi uchun tushunchalar berib ketish uchun foydalaniladi. Dastur interpritatsiya qilinganda (ishga tushirilganda) interpritator izohlarni e’tiborsiz qoldiradi, shuning uchun ular dasturning ishlashiga ta’sir qilmaydi. Pythondagi izohlar blok va inline izohlarda keladi.

Bitta satrda joylashuvchi izohlardan oldin panjara belgisi- # qo‘yiladi:

# Konsolga chiqish

# Salom Dunyo xabarlari

print("Salom Dunyo ")

# belgisidan keyingi har qanday belgilar to‘plami izohni bildiradi. Ya’ni, yuqoridagi misolda kodning birinchi ikki qatori izohlar hisoblanadi.

Shuningdek izohlar ifodalardan so‘ng, til ifodalari bilan bir qatorda joylashgan bo‘lishi ham mumkin:

print("Salom dunyo")   # Xabarni konsolga chop etish

Blok izohlarida sharh matnidan oldin va keyin uchta bitta tirnoq qo‘yiladi: *‘‘‘sharh matni’’’*. Masalan:

‘‘‘

    Konsol chiqishi

    Salom dunyo xabarlari

‘‘‘

print ("Salom dunyo")

**Asosiy funksiyalari**

Python ko‘plab o‘rnatilgan, standart funktsiyalarni taqdim etadi. Ulardan ba’zilari, ayniqsa, til o‘rganishning dastlabki bosqichlarida juda tez-tez qo‘llaniladi, shuning uchun ularni ko‘rib chiqaylik.

  Konsolga ma’lumot chiqarishning asosiy funksiyasi - bu print() funksiyasi hisoblanadi. Biz chiqarmoqchi bo‘lgan satr ushbu funktsiyaga argument sifatida uzatiladi:

print ("Salom Python")

Agar konsolga bir nechta qiymatlarni chop etish kerak bo‘lsa, ularni vergul bilan ajratib, print() funktsiyasiga beriladi:

print("To‘liq ismi:", "Tom", "Smit")

Natijada, vergul bilan ajratilgan joylarga probel bilan ajratilgan holatda, barcha berilgan qiymatlar, bir qatorga joylashtirilib konsolga chiqariladi:

To‘liq ismi: Tom Smit

Agar **print()** funksiyasi chiqarish uchun javobgar bo‘lsa, u holda **input()** funktsiyasi axborotni kiritish uchun xizmat qiladi. **input()** funksiyasi kiritilgan ma’lumotni satr tipida qaytaradi va bu satrni o‘zgaruvchiga saqlanadi va shu qiymatga o‘zaruvchi orqali murojaat qilinadi:

name = input("Ismni kiriting:")

print ("Salom", ism)

Konsol chiqishi:

Ismni kiriting: Eugene

Salom Eugene

**Nazorat savollari:**

1. Python paketini qanday o‘rnatish mumkin ?

2. Python kodini oddiy matnli faylga saqlash orqali ishga tushirish uchun qanday qadamlarni bajarishim kerak ?

3. Python dasturlash tili bilan ishlash uchun qanday tarjimonlardan foydalanish mumkin ?

4. Python dasturlash tilining nomi qayerdan kelib chiqqan ?

5. Python dasturlash tilining tarixi haqida gapirib bering .

6. Python dasturlash tili logotipini kim ixtiro qilgan .

7. Pythonni oddiy va tushunarli dasturlash tili nima qiladi ?

8. Pythonning boshqa dasturlash tillaridan qanday afzalliklari bor ?

9. Python dasturlash tili yordamida qanday dasturlar yaratish mumkin ?

10.Pythonda dastur strukturasi qanday tuzilgan?

11.Satrning oldida qoldirilgan oraliq masofalar (indents) nimani anglatadi?

12.Pythonning sintaksisi boshqa dasturlash tillarining sintaksisidan eng katta farqli jihati nimada?