

**O‘ZBEKISTON RESPUBLIKASI AXBOROT TEXNOLOGIYALARI VA
KOMMUNIKATSIYALARINI RIVOJLANTIRISH VAZIRLIGI**

**MUHAMMAD AL-XORAZMIY NOMIDAGI
TOSHKENT AXBOROT TEXNOLOGIYALARI UNIVERSITETI
URGANCH FILIALI**

Ish ko‘rib chiqildi va himoyaga qo‘yildi

Axborot texnologiyalari

kafedra mudiri

_____ Xalmuratov O.U.

« ____ » _____ 2018 yil

Duschanov Mirzabek Egamberganovich

Bitiruvchining F.I.O.

Talabalarning ota-onalari bilan muloqot mobil ilovasini yaratish
mavzusidagi

5330500 – Kompyuter injiniringi (“Kompyuter injiniringi”) yo‘nalishi bo‘yicha
bakalavr darajasini olish uchun

BITIRUV MALAKAVIY ISHI

Bitiruvchi

(imzo)

Duschanov M

(f.i.o.)

Rahbar

(imzo)

O`razmatov T

(f.i.o.)

Maslahatchi

(imzo)

Allabergenova D.

(f.i.o.)

Taqrizchi

(imzo)

Baltayeva I

(f.i.o.)

Urganch – 2018 y.

O'ZBEKISTON RESPUBLIKASI AXBOROT TEXNOLOGIYALARI VA
KOMMUNIKATSIYALARINI RIVOJLANTIRISH VAZIRLIGI

TOSHKENT AXBOROT TEXNOLOGIYALARI UNIVERSITETI
URGANCH FILIALI

Kompyuter injiniringi fakulteti
Axborot texnologiyalari kafedrası

Kompyuter injiniringi yo'nalishi

TASDIQLAYMAN
Axborot texnologiyalari
kafedrası mudiri

_____ Xalmuratov O.U
«__» _____ 2018

Bitiruv malakaviy ishiga

T O P S H I R I Q

(familiyasi, ismi, otasining ismi)

1. Bitiruv malakaviy ishi mavzusi

2. Muhammad al-Xorazmiy nomidagi TATU Urganch filialining «__» _____
2018 yil sanadagi № __ sonli buyruqi bilan tasdiqlangan

3. Ishni himoyaga topshirish muddati _____

4. Ishga oid dastlabki ma'lumotlar: _____

5. Hisoblash-tushuntirish yozuvlarining mazmuni (ishlab chiqiladigan masalalar
ro'yxati): _____

6. Grafik materiallar ro'yxati: _____

7. Topshiriq berilgan sana _____

Rahbar _____
(imzo)

Topshiriq oldim _____
(imzo)

8. Ishning ayrim bo'limlari bo'yicha maslahatchilar

Qism	Maslahatchi o'qituvchining F.I.O.	Imzo, sana	
		Topshiriq berildi	Topshiriq Olindi
1. Tizimli tahlil va masalaning qo'yilishi			
2. Asosiy qism			
3. Mehnat muhofazasi va texnika xavfsizligi			

9. Ishni bajarish grafigi

No	Ish qismlarining nomi	Bajarish muddati	Rahbar (maslahatchi) Belgisi
1	Bitiruv malakaviy ishi mavzusini tasdiqlash		
2	Mavzu bo'yicha adabiyotlarni yig'ish va o'rganish		
3	Ta'lim muassasasi tizimini tubdan o'rganish		
4	Ta'lim muassasasiga axborot tizimlarini yanada ko'proq tadbiiq qilishni rejalashtirish		
5	Algoritm va dasturiy ta'minoti yaratish		
6	Mehnat muhofazasi va texnika xavfsizligi topshiriqlarini bajarish		
7	Xulosa qismini yozish		
8	Adabiyotlar ro'yxatini shakllantirish		
9	Chizma – grafik ishlar, prezentatsiya tayyorlash		
10	Bitiruv ishini rasmiylashtirish (perepletlash) va kafedraga topshirish		

Bitiruvchi _____ «__» _____ 2018
(imzo)

Rahbar _____ «__» _____ 2018
(imzo)

MUNDARIJA

KIRISH.....

I BOB TA'LIM MUASSASALARINING FAOLIYATINI TIZIMLI

TAHLILI VA MASALANING QO'YILISHI

1.1-§. Ta'lim muassasalari faoliyatini tahlil qilish.....

1.2-§. O'quv muassasalari axborot tizimlariga bo'lgan talablarni o'rganish

1.3-§. Axborot tizimi yaratishning asosiy qoidalari, dizayn bosqichlari va talablari

II BOB. OTA-ONALAR BILAN MULOQOT DASTURI

ALGORITMLARI VA UNI ISHLAB CHIQISH

2.1-§. Masalani yechilishining instrumental vositalarini tanlash

2.2-§. Dastur ishlab chiqish uchun python dasturlash tili imkoniyatlari.

2.3-§. Foydalanuvchi uchun dastur yo'riqnomasi.....

III BOB. MIQROIQLIM. MAXALLIY SHAMOLLATISHNI

XISOBLASH.....

3.1-§. Sanoat korxonalarini binolarini shamollatish.....

3.2-§. Ishlab chiqarish miqroiqlimining o'lchash usullari.....

3.3-§. Ishlab chiqarish miqroiqlimining gigienik normalari.....

XULOSA.....

FOYDALANILGAN ADABIYOTLAR RO'YHATI.....

ILOVALAR.....

KIRISH

Mamlakatimizda davlat organlari web-saytlari va hukumat portali orqali interaktiv davlat xizmatlari ko'rsatishni yanada kengaytirish borasida izchil ishlar amalga oshirilmoqda. Shu maqsadda iqtisodiyot va jamiyat hayotining barcha sohalariga axborot – kommunikatsiya texnologiyalarini joriy qilishni rag'batlantiradigan mustahkam qonunchilik bazasi yaratilyapdi.

Zamonaviy axborotni kommunikatsion axborot texnologiyalarsiz amalga oshirib bo'lmaydi. O'z vaqtida tezkor va sifatli axborotlarga ega bo'lish, to'g'ri qarorlar qabul qilishni ta'minlaydi. Natijada korxonalar raqobatbardoshligi ta'minlanadi. Rivojlangan davlatlar iqtisodiyotini muvaffaqiyati sabablaridan biri ularni doimo tezkor va aniq axborotlarga ega bo'lish imkoniyati yuqoriligidadir. Shunday ekan Respublikamizda zamonaviy axborot kommunikatsion texnologiyalarni turli sohalarda qo'llash zaruriyati mavzusi **dolzarbligidan** dalolat beradi.

Mavzuning dolzarbligi. Bitiruv malakaviy ish mavzusining dolzarbligi shundan iboratki, hozirgi vaqtda yoshlar ta'lim tarbiyasi, yoshlarni to'g'ri yo'lga boshlash, har tomonlama shakllangan komil inson qilib tarbiyalash har bir davlat yurt uchun muhim maqsadlardan biridir. Ushbu maqsadlarga erishish uchun zamonaviy axborot va kommunikatsiya texnologiyalaridan keng foydalanish, ta'lim jarayonida axborot texnologiyalarini qo'llash, ta'limda internet tarmog'i imkoniyatlaridan foydalanish katta ahamiyatga ega. Ta'lim muassasalarida ota-onalar bilan muloqot qilishni yo'lga qo'yilsa, bugungi kunda yoshlar tarbiyasidagi dolzarb muammolardan biri yechilgan hisoblanadi.

Bitiruv malakaviy ishining obykti. Ma'lumki biror muassasaning axborot tizimi tashrif buyuruvchilarga ushbu muassasa, uning faoliyati, umuman olganda muassasa to'g'risidagi barcha malumotlarni aks ettirib berishi lozim. Shuning uchun bitiruv malakaviy ishida birorta ta'lim muassasasi ta'lim tarbiya jarayonini tashkil etilishi, madaniy – marifiy tadbirlar, yangiliklar, qabul jarayoni, va boshqa faoliyati obyekt sifatida qaraldi.

Bitiruv malakaviy ishining predmeti. Dasturiy ta'minot ishlab chiqishda

o'rganilgan dasturlash tillari va algoritmlari ishning predmeti sifatida qaraldi. **Bitiruv malakaviy ishining maqsadi va vazifalari.** Bitiruv malakavish ishning maqsadi – ta'lim muassasalarida talabalar yoki o'quvchilarning ota-onalari bilan muloqotni axborot texnologiyalaridan foydalangan holda onlayn tashkil etishdan iborat.

Axborot tizimi orqali ta'lim muassasasi haqida ma'lumot, guruh rahbarlari haqida ma'lumot va ota-onalar bilan muloqotni tashkil etish bitiruv malakaviy ishining vazifalariga kiradi.

Bitiruv malakaviy ishi tuzilmasi tasnifi. Bitiruv malakaviy ishining tuzilmasi kirish, uch bob, xulosa, foydalanilgan adabiyotlar ro'yhati va ilovalardan iborat. Masalaning qo'yilishi

I BOB I BOB TA'LIM MUASSASALARINING FAOLIYATINI TIZIMLI TAHLILI VA MASALANING QO'YILISHI

1.1-§. Ta'lim muassasalari faoliyatini tahlil qilish

1989-1993 – yillarda Urganch shahridagi fizika, matematika va kimyo fanlariga ixtisoslashgan maktab bo'lib, X.Siddiqov nomi berilgan.

1993-1998 – yillarda Xalq ta'limi vazirligi hamda oliy va o'rta maxsus ta'lim vazirligining 01.12.1994 yildagi №258 va 236-sonli qo'shma buyrug'iga asosan mazkur 20-maktab UrDU qoshidagi litsey-internatga aylantirildi.

Litsey – internatga O'zbekiston Respublikasi oliy va o'rta maxsus, kasb-hunar ta'limi Markazining 16.07.1998 yildagi 192-sonli buyrug'iga asosan UrDU qoshidagi akademik litsey nomi berildi.

Xorazm viloyati hokimiyati va O'zbekiston Respublikasi oliy va o'rta maxsus ta'lim vazirligining 2006-yil 15-fevralda chiqargan 6-340/39-sonli qo'shma buyrug'iga asosan UrDU qoshidagi akademik litsey 01.02.2006 yildan TATU Urganch filiali qoshidagi 1-son akademik litsey qilib o'zgartirildi.

Akademik litseyda ta'lim tarbiya-jarayoni DTS lariga mos ravishda olib boriladi. Har bir o'quv mashg'ulotlari yangi pedagogik texnologiyalar va zamonaviy texnik vositalar, ko'rgazmali qurollardan foydalanilgan holda o'tkaziladi.

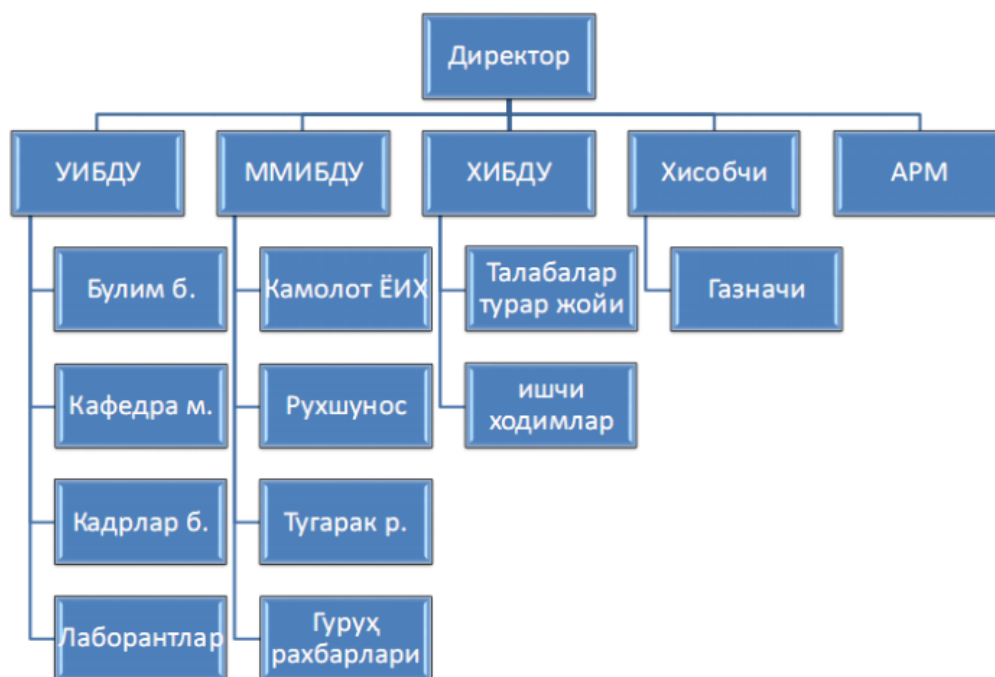
Litseyda 2 ta kompyuter, 2 ta lingafon, 2 ta fizika laboratoriya, 1 ta kimyo laboratoriya va 1 ta biologiya laboratoriya xonalari mavjud.

Akademik litseyda 200 o'rinli talabalar turar joyi mavjud bo'lib, unda talabalar istiqomat qilishlari uchun barcha shart-sharoitlar yaratilgan.

2014-2015 o'quv yilida 12 nafar talaba viloyat fan olimpiadasida g'olib bo'ldi. Litseyda DTM bazasi asosida muntazam test sinovlari olib boriladi.

2013-2015 o'quv yillarida litsey bitiruvchilarining 60% dan ortig'i xalqaro va respublika oliy o'quv yurtlari talabalari bo'lishdi.

Litsey tuzilmasi



1.1-rasm. Litsey tuzilmasi

Ta'lim yo'nalishlari. TATU Urganch filiali qoshidagi 1-son akademik litseyga 2016-2017 o'quv yili uchun quyidagi yo'nalishlar bo'yicha talabalar qabul qiladi.

1. a) Aniq (fizika-matematika) fanlar yo'nalishi

b) Aniq (iqtisod) fanlar yo'nalishi

2. Tabiiy fanlar yo'nalishi

3. Xorijiy filologiya yo'nalishi.

O'qish muddati 3 yil. Xalqaro, respublika, viloyat fan olimpiadalari g'oliblari yo'nalishlariga mos ravishda qabul qilinadi.

Akademik litseyni tugatgan talabalar yo'nalishiga qarab qo'shimcha EHM operatori, laborant, tarjimon kasblariga ega bo'ladilar

Litseyda o'z fanining mohir ustalari bo'lgan pedagoglar ta'lim berishadi.

Yotoqxonaga muhtojlar yotoqxonada bilan ta'minlanadi. Yotoqxonada har ikkala talabaga 1 xona ajratilgan bo'lib, ular uchun barcha shart-sharoitlar yaratilgan.

Litseyga qabul qilishda qabul komissiyasiga topshiriladigan xujjatlar ro'yxati:

1. Direktor nomiga ariza

2. 9-sinfni bitirganligi to'g'risidagi shahodatnoma
3. 086-Y shakldagi tibbiy ma'lumotnoma
4. Turar joyidan ma'lumotnoma
5. 3x4 – 4x6 o'lchamdagi fotosurat (6 dona)
6. Tug'ilganlik haqidagi guvohnoma yoki pasportining nusxasi
7. Xalqaro, respublika, shahar bosqichidagi fan olimpiadalari, tanlovlari va musobaqalarida golibligi to'g'risidagi hujjat
8. Pedagogik-psixologik tashxis markazining tavsiyanomasi

9. Imkoniyatlari cheklangan bitiruvchilar uchun Qoraqalpogiston Respublikasi Xalq ta'limi vazirligi, viloyatlar va Toshkent shahar xalq ta'limi boshqarmalari qoshidagi tibbiy-psixologik-pedagogik komissiyaning xulosasi va tavsiyasi yoki mehnat va aholini ijtimoiy muhofaza qilish boshqarmalarining, tuman(shahar) ijtimoiy ta'minot bo'limlarining yo'llanmalari. Raxbariyat. Akademik litseyning tuzilmasidan kelib chiqqan xolda litsey raxbariyat tarkibi o'rganildi. Raxbariyat haqidagi ma'lumotlar davlat organlarining veb saytlariga qo'yilgan talablarga ko'ra rasmiy veb saytda joylashtirilishi lozim. Shu sababli quyida keltirilgan litsey rahbarlari to'g'risidagi ma'lumotlar veb saytga joylashtiriladi:

- TATU UF qoshidagi 1-son akademik litsey direktori: Boltayev Jamol Ismailovich;
- O'quv ishlar bo'yicha direktor o'rinbosari Axmedov G'ulomjon Jumanazarovich;
- Ma'naviyat va ma'rifat ishlar bo'yicha direktor o'rinbosari Xudoyberganov Arslonbek Qalandarovich;
- Xo'jalik ishlar bo'yicha direktor o'rinbosari Raximov Rahmat Davlatovich

Ushbu ma'lumotlardan tashqari saytda raxbarlarning ish telefon raqamlari, elektron pochta manzillari, jismoniy va yuridik shaxslarni qabul kunlari va vaqtlari to'g'risidagi ma'lumotlar joylashtiriladi.

Kafedralar. Litseyda 5 ta kafedra mavjud bo'lib, saytga kafedra raxbarlari, kafedralar tashkil etilishi va kafedra bilan bog'liq boshqa ma'lumotlar joylanishi rejalashtirildi.

- Aniq fanlar kafedra mudiri Raximova Feruza Atabekovna;

- Tabiiy fanlar kafedra mudiri Atajanova Shirin;
- Xorijiy filologiya fanlari kafedra mudiri Vaisova Gulchehra Axmedovna;
- Ijtimoiy gumanitar fanlar kafedra mudiri Nurmetov Ismoil Alimbaevich;
- O‘zbek filologiyasi kafedra mudiri Sultanova So‘najan Matyaqubovna.

1.2-§. O‘quv muassasalari axborot tizimlariga bo‘lgan talablarni o‘rganish

O‘rta maxsus ta’lim muassasalari veb sayti – yoshlar, murabbiylar, ota – onalar, shuningdek barcha foydalanuvchilar uchun muassasa faoliyati, ta’lim yo‘nalishlari haqidagi kerakli axborotlarni jamlash bilan bir qatorda, ularga axborot texnologiyalari soxasidagi kerakli ma’lumotlarni berish, o‘zaro muloqot qilish va tajriba almashinishlari uchun zarur imkoniyatlarni yaratib berish vazifasini bajarishi lozim.

Xozirgi vaqtda O‘zbekiston respublikasida davlat organlari saytlariga qo‘yilgan talablar mavjud.

Rasmiy veb-saytni qonunchilik talablariga muvofiqligi Vazirlar Maxkamasining 31.12.2013 yildagi 355 - sonli qarori asosida amalga oshiriladi. Rasmiy veb saytni yuritishga qo‘yilgan talablardan bir nechtasini keltirib o‘tamiz:

1. Rasmiy veb-saytda joylashtirilgan asosiy axborotlardan foydalanish erkin va bepul bo‘lishi kerak.
2. Bazaviy axborot xizmatlaridan tashqari, davlat va xo‘jalik boshqaruvi organi, mahalliy davlat hokimiyati organi rasmiy veb-sayt orqali qonun hujjatlarida belgilangan tartibda pulli asosda qo‘shimcha xizmatlar ko‘rsatishi mumkin.
3. Rasmiy veb-saytga joylashtirilayotgan axborotlarning sanasi ko‘rsatilishi va ular muntazam yangilab borilishi kerak. Axborotlarni yangilab borish muddati davlat va xo‘jalik boshqaruvi organi, mahalliy davlat hokimiyati organi tomonidan belgilanadi, biroq axborotlar bir haftada kamida bir marta yangilanib borishi kerak.
4. Davlat boshqaruvi organlari, mahalliy davlat hokimiyati organlari tomonidan normativ-huquqiy hujjatlar rasmiy veb-saytga normativ-huquqiy hujjat rasman

e'lon qilingandan keyin ikki kundan kechikmay joylashtirilishi kerak.

5. Davlat va xo'jalik boshqaruvi organi, mahalliy davlat hokimiyati organi rahbarining buyrug'i bilan rasmiy veb-saytdagi axborotlarni shakllantirish, joylashtirish, shuningdek yangilab borish uchun mas'ul bo'lgan shaxslarni (bo'linmani) belgilaydi.
6. Davlat va xo'jalik boshqaruvi organi, mahalliy davlat hokimiyati organining rasmiy veb-sayti Internet tarmog'idagi «UZ» domen zonasida va O'zbekiston Respublikasining Hukumat portalida belgilangan tartibda ro'yxatdan o'tkazilishi, shuningdek O'zbekiston Respublikasi hududida serverda jismoniy joylashtirilishi kerak (xosting).
7. Rasmiy veb-saytdagi axborotlar davlat tilida taqdim etilishi kerak. Rasmiy veb-sayt axborotlari boshqa tillardagi versiyaga ega bo'lishi mumkin.
8. Davlat va xo'jalik boshqaruvi organi, mahalliy davlat hokimiyati organi o'zining rasmiy veb-saytida joylashtirilgan axborotlarning yo'q qilinishi, to'sib qo'yilishi, buzib talqin qilinishi, qalbakilashtirilishining hamda boshqa shakllardagi ruxsatsiz aralashuvlarning oldini olish bo'yicha tegishli muhofaza chora-tadbirlarini ko'rishi kerak.

1.3-§. Axborot tizimi yaratishning asosiy qoidalari, dizayn bosqichlari va talablari

Kundalik hayotimizning har bir sohalarida axborot-kommunikatsion texnologiyalari muxitida Internet jadal suratlar bilan rivojlanib kelmoqda. Xususan O'zbekistonda ta'lim sohasida olib borilayotgan tinimsiz izlanishlarni keng ommaga targ'ib qilish, soxa yangiliklariga o'z vaqtida ega bo'lish, "Open access" tizimida ma'lumotlar bazasini yaratish ya'ni, axborot ta'lim resurslaridan ochiq va keng foydalanish, ustoz – shogird o'qitish tizimini on-lineda olib borish, masofaviy ta'lim olish va boshqa ko'pgina imkoniyatlar ochilmoqda. Bunday yutuqlar keng ko'lamli Butun jahon o'rgimchak to'ri, Internet tarmogi va unga joylashgan Veb – saytlarning doimo aktual turishi va taxrir qilinishi evaziga erishiladi. So'ngi statistik ma'lumotlarga ko'ra, 2007 yil uchun O'zbekistonda: Internet tarmogida

faoliyat olib borish uchun fundamental masalalaridan biri Veb – saytni ishlab chiqish jarayonidir. Bu borada O‘zbekiston Respublikasi Oliy va o‘rta maxsus ta’lim Vazirligining 2006 yil 29 dekabrda “Axborot ta’lim resurslarining yaratilishi, ro‘yxatdan o‘tkazilishi va joylashtirilishini tartibga solish to‘g‘risida”gi 296-sonli buyrug‘iga muvofiq, saytni loyixalashtirishda quyidagilarni aniqlash maqsadga muvofiqdir: Birinchidan, saytni loyixalashtirish bosqichlariga quyidagilar kiradi:

WWW xizmati yaratilganiga hozirgi kunda 20 yildan oshgan bo‘lsa, shu vaqt davomida veb sahifalar statik ma’lumotlarni tasvirlashdan hozirgi kundagi dinamik shakliga bo‘lgan rivojlanish bosqichlarini bosib o‘tdi. Dastlabki davrda veb sayt va veb sahifalarni yaratish uchun faqat HTML razmetkalar tili ishlatilgan bo‘lsa, hozirgi kunda esa veb saytlarni yaratish uchun maxsus texnologiyalar, veb uchun mo‘ljallangan dasturlash tillari, script yaratish tillari, mijoz – server texnologiyasi, hamda shular bilan birgalikda veb sayt va portallarni yaratishning metod va talablari, uning mazmuni va mo‘ljallangan auditoriyasiga moslash talablari vujudga keldi. Quyida veb saytlarni yaratish bo‘yicha ba’zi talablarni keltirib o‘tamiz.

Birinchi navbatda saytning maqsad va vazifalari aniq belgilab olinishi kerak. Ushbu maqsadlar asosida saytni yaratish talablari, yani uning mazmuni, tarkibi, strukturasu belgilab olinadi.

Ikkinchi navbatda sayt kimlar uchun mo‘ljallangan, kimlar saytni foydalanuvchilari ekanligini aniqlab olish zarur.

Saytga maqsadi, vazifalari va tematikasiga mos materiallar tarkibi, turi tahlil qilinib saytning axborot resurslari tuzilmasi katigo‘riyalar bo‘yicha klassifikasiyalandi;

Agar sayt davlat boshqari yoki xo‘jalik boshqaruvi tashkilotiga tegishli sayt yoki portal bo‘lsa unda sohaga oid yangiliklar kiritilib borilish va bu yangiliklarni kiritish uchun imkoniyatlar yaratilgan bo‘lishi zarur. Shuningdek hukumat qarorlari - yo‘nalishga oid xukumat qarorlari va Prezident farmonlari keltirilgan bo‘lishi lozim;

Portalda uning tashkil etuvchilarixaqidagi ma'lumotlar, yani: manzili, telefon raqami, electron pochta manzili, faks raqami keltirilishi lozim.

Portalda sohaga va joriy tashkilotda bo'layotgan e'lonlar, tadbirlar xaqidagi ma'lumotlar bo'lishi lozim;

Foydali web-saytlari – yo'nalishga tegishli qiziqarli va mashhur Veb – saytlarning ro'yxati keltiriladi

Veb sayt yaratishda dizayn va rang tanlash.

Har qanday dizaynning asosini rang tashkil etadi. Dizaynning har qanday jodiy ishi quyidagilardan iborat bo'ladi: 1) Rasm. 2) Tekst 3) Rang

Bularning ichida dizaynning ko'rkam bo'lishini hal qiluvchi narsa bu aynan rang hisoblanadi. Duch kelgan rangni tanlash ko'rimsiz dizaynga sabab bo'ladi. Rang tanlashda juda ehtiyot bo'lish kerak. Aslida ranglar bilan ishlash haqida juda ko'p kitoblar yozilgan. Agar internetning qidiruv saytlaridan "Color theory" yoki "Color scheme" – deb qidirsangiz bu haqda juda ko'plab maqola va kitoblar borligini ko'rasiz. Ammo bularni chuqur o'rganish sizdan ko'p vaqt talab etadi. Qolaversa ularni hammasini siz amalda qo'llamaysiz. Shuning uchun sizga eng kerakli malumotlarni qisqacha keltirishga qaror qildik.

Tabiatga diqqat bilan etibor qilsangiz, undagi har bir rang nimadandir darak beradi. Tabiatni yaratgan Xudo undagi ranglarni juda munosib yaratgan. Masalan yashil rang insonni ruhan tinchlantiradi. Shuning uchun tabiatda asosan yashil rang ko'p takrorlanadi. Qizil rang esa insonni hushyorlikka chaqiradi Yoki ogohlantiradi. Shuning uchun jonzotlarning qoni qizil rangda. Tikonli atirgullarning rangi ham qizil bo'lishi bejiz emas.

Ranglarni insonga ta'sirini o'rganish bilan nafaqat olimlar yoki rassomlar shug'ullangan. Balki psixologlar ham ranglarning insonga ta'sirini o'rganishgan.

Har bir rang qanday ma'no berishini birma bir sanaymiz:

QIZIL – Ogohlik, Jasurlik, issiqlik, G'azab;

KO'K – bilim, donolik, ulug'vorlik, yuksaklik, sovuqlik, muloqot;

SARIQ - xursandlik, do'stlik, ishonch, yordam;

YASHIL - tinchlik, xotirjamlik, o'sish, atrof muhit, hayot;

JIGARRANG YOKI TO‘Q SARIQ – issiqlik, sahovat, taom-ovqat, xavfsizlik, xursandchilik;

BINAFSHA - Boylik, mo‘jiza, cheksizlik, sifat, ijod, go‘zallik;

QORA – Maxfiylik, og‘irlik, kuch-quvvat

OQ – tozalik, oddiylik, yengillik, soflik, ochiqko‘ngillik.

Demak ranglarning ma’nosini anglab oldik. Biror mavzuda ijod qilishda avvalo shu mavzuga mos rang tanlash kerak. Masalan: Meditsina mavzusida biror banner dizayni yoki sayt dizayni tayyorlamoqchi bo‘lsangiz, u holda bunday dizaynga ko‘proq ko‘k va oq rang mos keladi. Sport mavzusiga esa ko‘proq qizil va qora ranglar kombinatsiyasi mos keladi. Chunki qizil rang bu qiziqqonlikni bildiradi, qora esa kuch quvvatni anglatadi. Lekin bu ranglarni ishlatishning o‘z qoidasi bor. Ya’ni fon rangi matn rangi rasmlar rangi va boshqalarni tanlashda ehtiyot bo‘lish kerak.

Veb Sayt turlari

- **Tashrif sayti.** Jismoniy yoki yuridik shaxs, ko‘rxona, xaqida qisqacha ma’lumot taqdim etuvchi internet sahifa.
- **Tematik sayt.** Bir mavzuni yoritib beruvchi veb sahifa. Barcha maqolalar bir mavzuga yoki yonalishag yonaltirilgan.
- **Qidiruv tizimi.** Internet foydalanuvchiga kerak bo‘lgan ma’lumotni izlab topishga ko‘mak beradi (yandex, google, bing va boshqalar).
- **Pochta serveri.** Bu sayt foydalanuvchiga elektron pochta yashigini tashkil qilishga va elektron xat jonatish va qabul qilishga imkon beradi (mail.ru)
- **Fail saqlovchi sayt.** Foydalanuvchiga shaxsiy fayllarni saqlash uchun server diskdan joy ajratib beradi.
- **Blog.** Shaxsiy veb sahifa bo‘lib, qaysidur shaxsga tegishlidir, u o‘zi xaqida yoki o‘z iishi xaqida ma’lumot beradi.
- **Chat.** Bu sayt sahifasida onlayn rejimida bir nechta foydalanuvchi yozma ravishda muloqat qiladilar.

- **Forum.** Bu sahifada bir mavzu muxokama qilinadi, muxokama bir necha soat yoki bir necha yil xam davom etishi mumkin, mavzuni dolbzarligiga va qiziqarliligiga bog'liq.
- **Portal.** Ko'plab xizmatlarni o'z ichida jamlagan veb resurs portal deb ataladi. Ziyonet.uz, yandex.ru, google.com, mail.ru, yahoo.com va boshqa shunga oxshash saytlar ko'plab imkoniyatlarga ega. Bu saytlar qidiruv sistemaga ega, pochta server, tematik sahifalar va yangiliklar, o'z ichiga forum va charlarni kiritgan.

Veb Sayt. Veb Sayt bu hujatlar eg'indisi bo'lib barcha ma'lumot HTML kodi yordamida kiritiladi, hujatlar esa asosan *.html, *.css, *.php, *.js formatida papkalarga joylashgan bo'ladi. Xar bir sayt domen va xostga ega.

Domen veb saytning adresi (mail.ru, yandex.ru, ziyonet.uz). Xosting veb sayt joylashgan hudud. Provyayder kompaniyalari kelishuv asosida sizning saytingiz uchun wrb serverdan joy ajratadi.

Veb sayt yaratishning ba'zi umumiy qoidalari va bosqichlari. Brauzerlarni tanlash va yangi texnologiyalardan foydalanish to'g'risida to'g'ri qaror qabul qilish ko'p hollarda foydalanuvchilarning tanlangan auditoriyasiga bog'liq bo'ladi. Yangi sayt ishlab chiqishdan avval tadqiqotlar o'tkazib, qanday platformalar, brauzerlar, texnikaviy yangiliklar afzal ko'rilayapti, foydalanuvchilar ulanishning qanday tezliklaridan foydalanayotganligini aniqlash zarur. Agar mavjud sayt qayta ishlanayotgan bo'lsa, oldin serverda ro'yxatga olingan yozuvlar, saytdan foydalanish to'g'risida saqlanayotgan axborot tahlil qilinadi.

ilmiy yoki akademik mazmundagi sayt ishlab chiqilayotgan bo'lsa, ushbu sayt brauzerlarda qanday ishlashiga alohida e'tibor qaratish zarur;

sayt iste'molchilar uchun mo'ljallangan bo'lsa, brauzerlar uchun saytning yuklanish tezligi va tashqi ko'rinishiga e'tibor beriladi;

Ko'p maqsadli saytlarning aksariyati uchun «farqlarni ajratish» yondashuvidan foydalanish ma'quldir, agar resurslar imkon bersa, bir nechta versiyalarni yarating va ularga tegishlicha xizmat ko'rsatiladi.

Veb-sahifaning aniq o'lchamini aniqlashdagi birinchi qadam display ta'minlab beradigan eng katta makonni aniqlash bo'lishi zarur. Kompyuter

displeylari turli standart o'lchamlarga ega bo'ladi va, odatda, dyuumlarda o'lchanadi. Moslashuvchan sahifalarni ishlab chiqish. Veb-sahifalar sukut bo'yicha moslashuvchan qilib ishlab chiqiladi. HTML-fayl matni va elementlari brauzer oynasiga tushadi, bunda ular, displeyning o'lchamidan qat'i nazar, butun makonni to'ldiradi. Agar brauzer oynasining o'lchami o'zgarsa, elementlar yangi o'lchamlarga sozlash uchun boshqatdan kiritiladi. Veb ning mohiyati mana shunda namoyon bo'ladi. Ko'plab dizaynerlar ongli ravishda sahifalarni ular Veb-oyna kengayishlari va zichlanishlariga bardosh berishi uchun shunday qilib ishlab chiqadilar. Bu yondashuv o'zining afzalliklari va kamchiliklariga ega.

Afzalliklari:

- Veb-sahifalar ajrata olishi turlicha bo'lgan displeylarda aks ettiriladi, moslashuvchan sahifani har qanday displeyga chiqarish uchun sozlash mumkin;

Kamchiliklari:

- katta displeylarda qator uzunligi haddan ortiq bo'lishi mumkin, bunda matn brauzer oynasining butun enini to'ldiradi. Uzun qatorlar ekrandan o'qish uchun, ayniqsa, noqulaydir, shuning uchun oyna yoki freymning butun eni matn bilan to'ldirilganda, ko'p foydalanuvchilar uchun o'qish sharoitlari ancha yomonlashadi;

- katta displeylarda elementlar ekranda yetarli darajada uyg'un joylashgan bo'ladi, kichik displeylarda ular zich bo'ladi;

- moslashuvchan dizayn natijalarini oldindan aytish qiyin va sahifalar turli foydalanuvchilarda turlicha bo'ladi.

O'lchami qaydlangan sahifalarni ishlab chiqish . Kimda-kim sahifalarni belgilash ustidan ko'proq nazorat o'rnatishni istasa, eni qaydlangan sahifalar ishlab chiqish kerak bo'ladi, displey o'lchami yoki oyna o'lchamlari o'zgarishlaridan qat'i nazar u, barcha foydalanuvchilar uchun doimiy bo'ladi. Bu yondashuv noshirlik ishida sahifalar yaratish printsiplariga asoslangan bo'lib, bu printsiplar doimiy to'rni, sahifada joylashgan elementlarning nisbatini va qatorlarning qulay uzunligini saqlab turish kabilardan iboratdir.

Afzalliklari:

- displey o'lchamlaridan qat'i nazar sahifa bir xil ko'rinadi. Bu, ayniqsa, barcha tashrif buyuruvchilarga o'zining imidjini bir xil qilib ko'rsatishga intiladigan kompaniyalar uchun muhimdir;

- kengligi qaydlangan sahifa va ustunlar uzun qatorlarni yaxshiroq boshqarishni ta'minlaydi. Katta displeylarda ko'rishda qatorlar juda ham uzun bo'lib ketmasligi uchun jadvallardan foydalanish mumkin.

Kamchiliklari:

- agar brauzerning ushbu oynasining o'lchami sahifa to'ridan kam bo'lsa, sahifaning ba'zi qismlari ko'rinmay qoladi va bo'yamasiga aylantirish talab etilishi mumkin. U deyarli har doim ishni qiyinlashtiruvchi to'siq sifatida qabul qilinadi, shuning uchun ham aksariyat hollarda undan qochish zarur. Yechimlardan biri - ko'pchilikka to'g'ri keladigan sahifa o'lchamini tanlash;

- ilgariydek brauzerlarda chiqariladigan simvollar o'lchamini nazorat qilish murakkab ish, elementlar ishlab chiqish vaqtida foydalanilgan o'lchamga nisbatan kattaroq yoki kichikroq o'lchamdan foydalanish natijasida oldindan aytib bo'lmaydigan darajada surilib ketishi mumkin;

- sahifa tasvirini to'laligicha nazorat qilishga intilish bu muhitga qarshi chiqish, demakdir. Veb - bu bosma nashr emas, uning o'z kuchli tomonlari va xususiyatlari bor. Moslashuvchan dizayn strategiyasi himoyachilari Veb da qaydlangan dizaynga o'rin yo'q, deb aytadilar.

1.4 Masalaning qo'yilishi. Ushbu bitiruv malakaviy ishida o'quv muassasalarida talabalarning ota-onalari bilan guruh murabbiylarining muloqotini online tashkil etish masala qilib olindi.

II BOB. OTA-ONALAR BILAN MULOQOT DASTURI ALGORITMLARI VA UNI ISHLAB CHIQUISH

2.1-§. Masalani yechilishining instrumental vositalarini tanlash

Python – bu o'rganishga oson va shu bilan birga imkoniyatlari yuqori bo'lgan oz sonlik zamonaviy dasturlash tillari qatoriga kiradi. Python yuqori darajadagi ma'lumotlar strukturasi va oddiy lekin samarador obyektga yo'naltirilgan dasturlash uslublarini taqdim etadi.

Pythonning o'ziga xosligi

- Oddiy, o'rganishga oson, sodda sintaksisga ega, dasturlashni boshlash uchun qulay, erkin va ochiq kodlik dasturiy ta'minot.
- Dasturni yozish davomida quyi darajadagi detallarni, misol uchun xotirani boshqarishni hisobga olish shart emas.
- Ko'plab platformalarda hech qanday o'zgartirishlarsiz ishlay oladi.
- Interpretatsiya(Интерпретируемый) qilinadigan til.
- Kengayishga (Расширяемый) moyil til. Agar dasturni biror joyini tezroq ishlashini xoxlasak shu qismni C yoki C++ dasturlash tillarida yozib keyin shu qismni python kodi orqali ishga tushirsa(chaqirsa) bo'ladi.
- Juda ham ko'p xilma-xil kutubxonalarga ega.
- xml/html fayllar bilan ishlash
- http so`rovlari bilan ishlash
- GUI(grafik interfeys)
- Web ssenariy tuzish
- FTP bilan ishlash
- Rasmi audio video fayllar bilan ishlash
- Robot texnikada
- Matematik va ilmiy hisoblashlarni programmalash

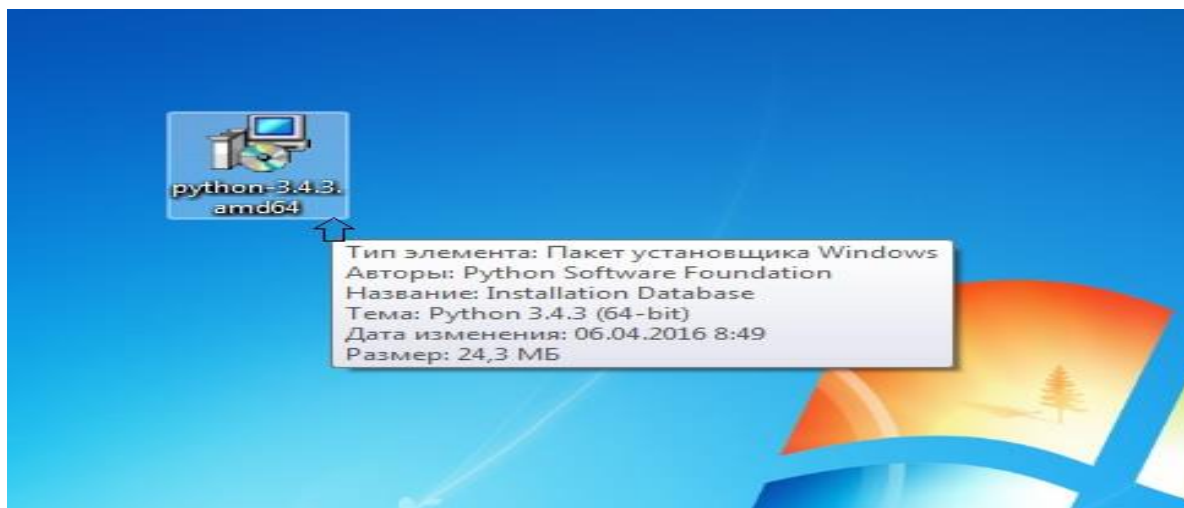
Pythonni katta proyektlarda ishlatish mumkin. Chunki, uni chegarasi yo`q, imkoniyati yuqori. Shuningdek, u sodda va universalligi bilan programmalash tillari orasida eng yaxshisidir.

Python dasturlash tilini o`rnatish.

Agar siz biror GNU/Linux distributivini ishlatayotgan bo'lsangiz ko'p xollarda sizning tizimingizda **python** o'rnatilgan bo'ladi. Buni tekshirib ko'rish uchun terminalingizdan quyidagi buyruqni ishga tushirib ko'ring. **python -V**

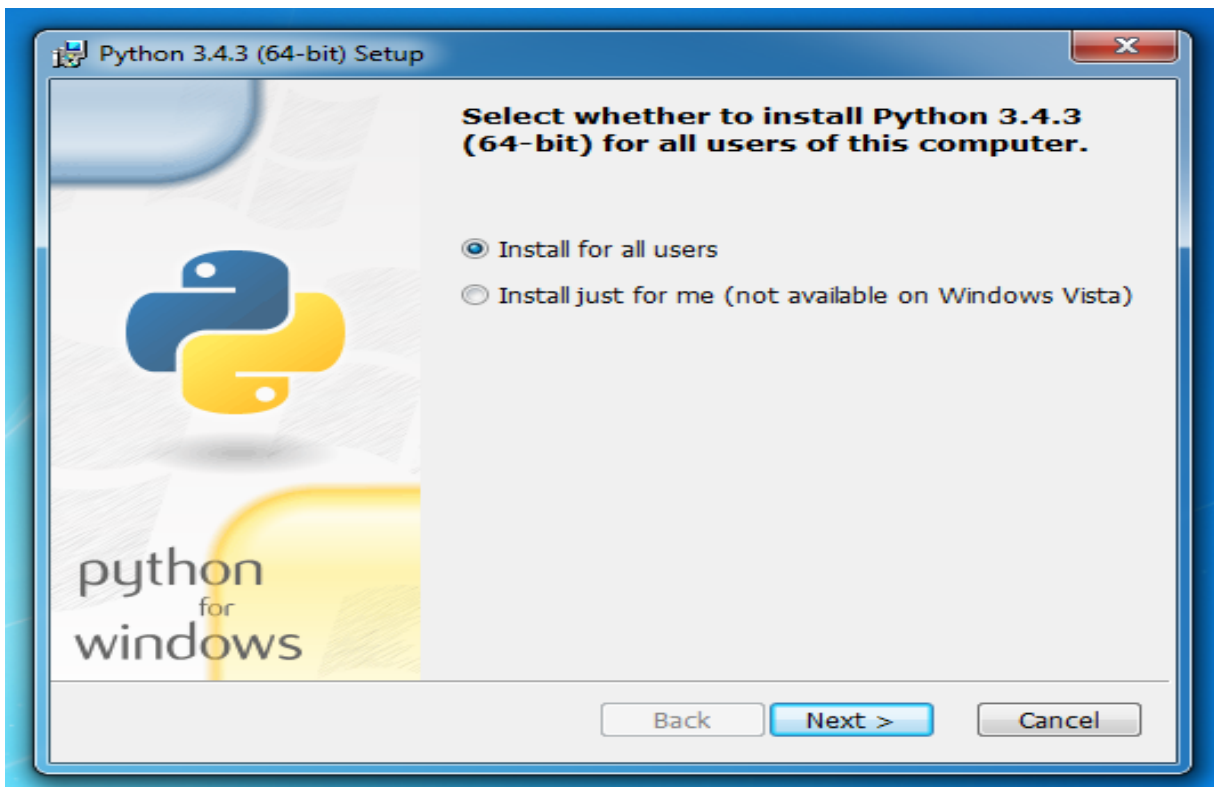
Agar sizda **Python 3.4.3** yozuvi yoki shunga o'xshash yozuv hosil bo'lsa unda hammasi joyida.

Windows operatsion tizimiga o'rnatish uchun www.python.org/downloads web sahifasiga o'tamiz va u yerdan oxirgi python versiyasini yuklab olamiz. Pythonni o'rnatish odatiy dasturlarni o'rnatish kabi kechadi. Hech qanday qiyin joyi yo'q.



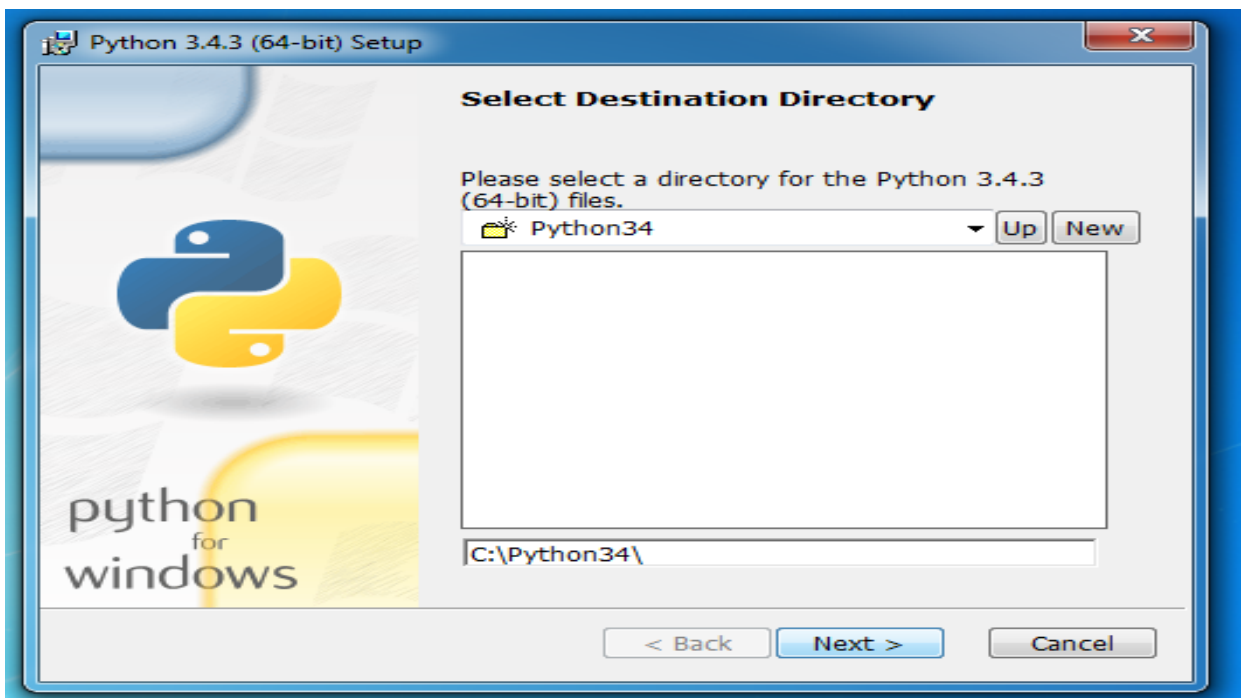
2.1-chizma. Python dasturining o`rnatiluvchi fayli.

Python dasturlash tilining o`rnatuvchi paketini ustiga sichqoncha ko`ratkichini 2 marta bosamiz va bizga quyidagi oyna hosil bo`ladi.



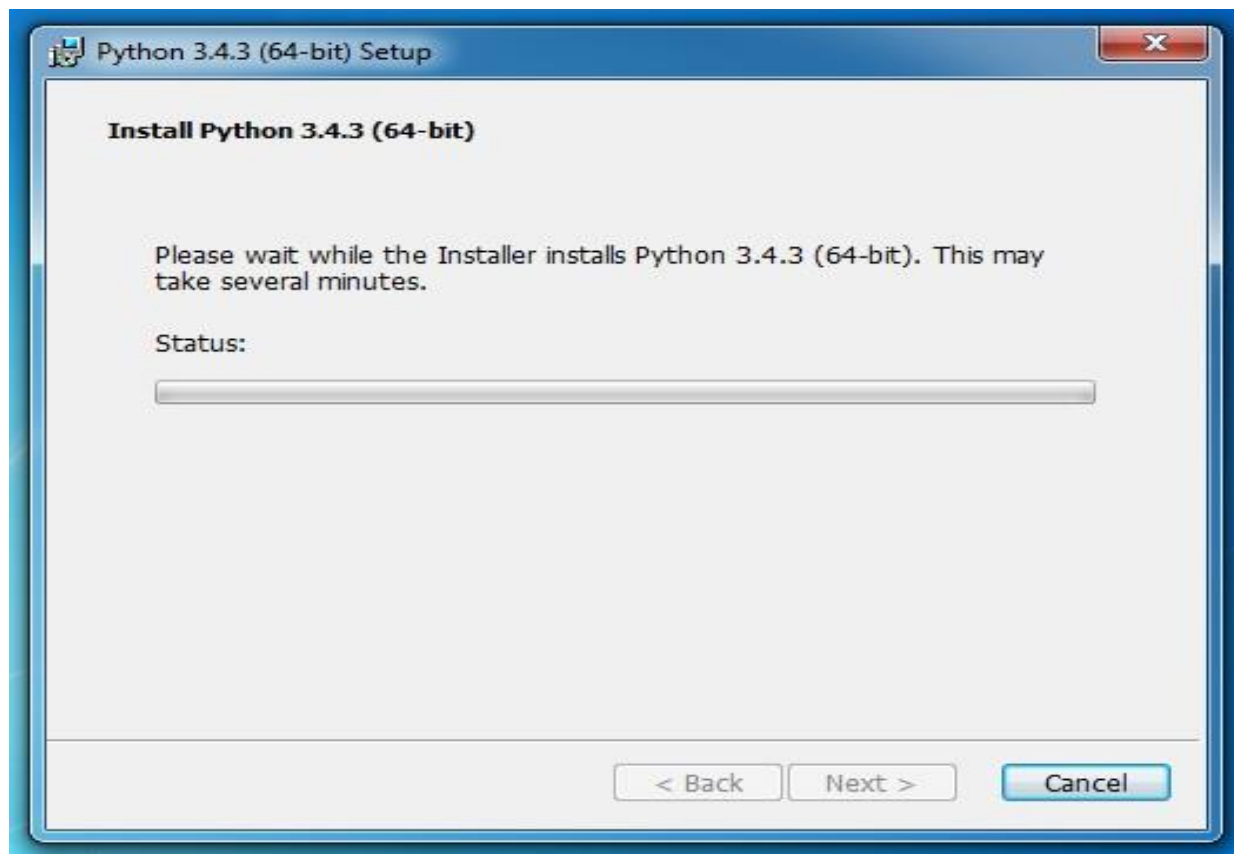
1.1.2-chizma. Python dasturini o`rnatishni boshlashni ko`rsatuvchi oyna.

Bu yerda **Install for all users**-barcha foydalanuvchilar uchun. **Install just for me**-faqat siz uchun, agar buni tanlab istalyatsiya qilsak ya'ni o`rnatsak Windows Vista operatsion sistemasida xatolik yuz beradi va dastur ishlamaydi. Shuning uchun **Install for all users** ni tanlaganimiz maqul. Keyin next tugmasi bosamiz.



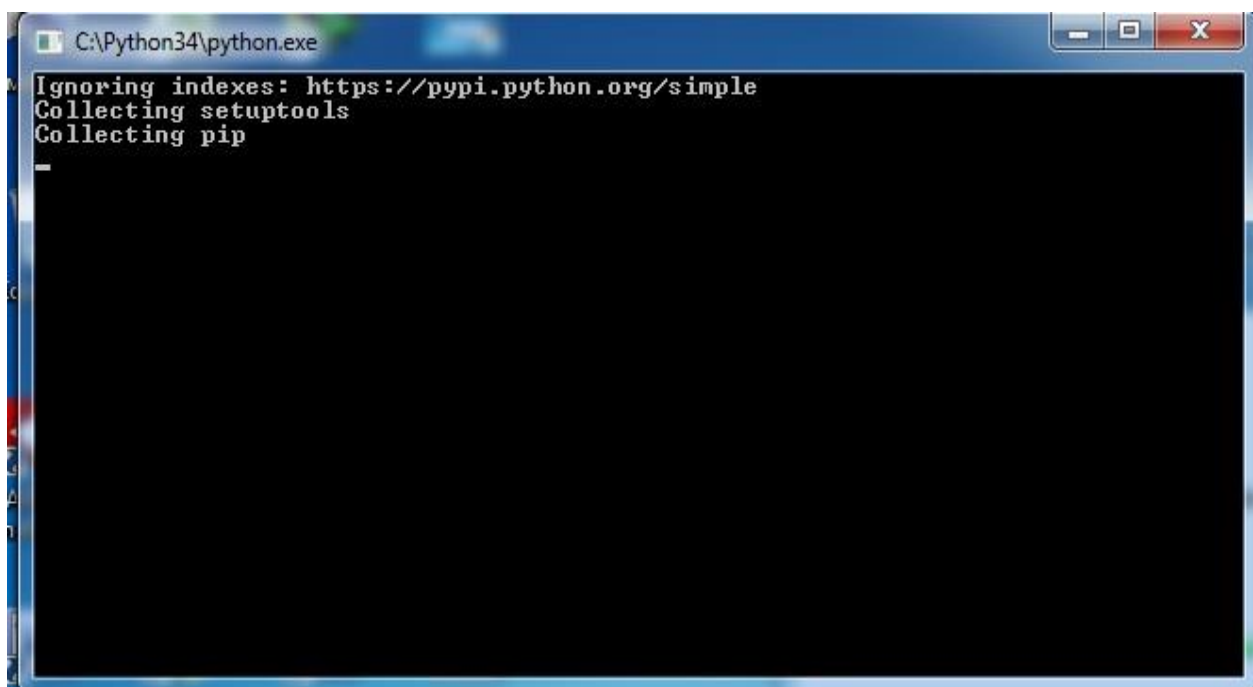
2.2-chizma. Python dasturini o`rnatilish joyini ko`rsatish oynasi.

Bu yerda esa Python dasturlash tilini qayerda o`rnatilishi ko`rsatilayapti.



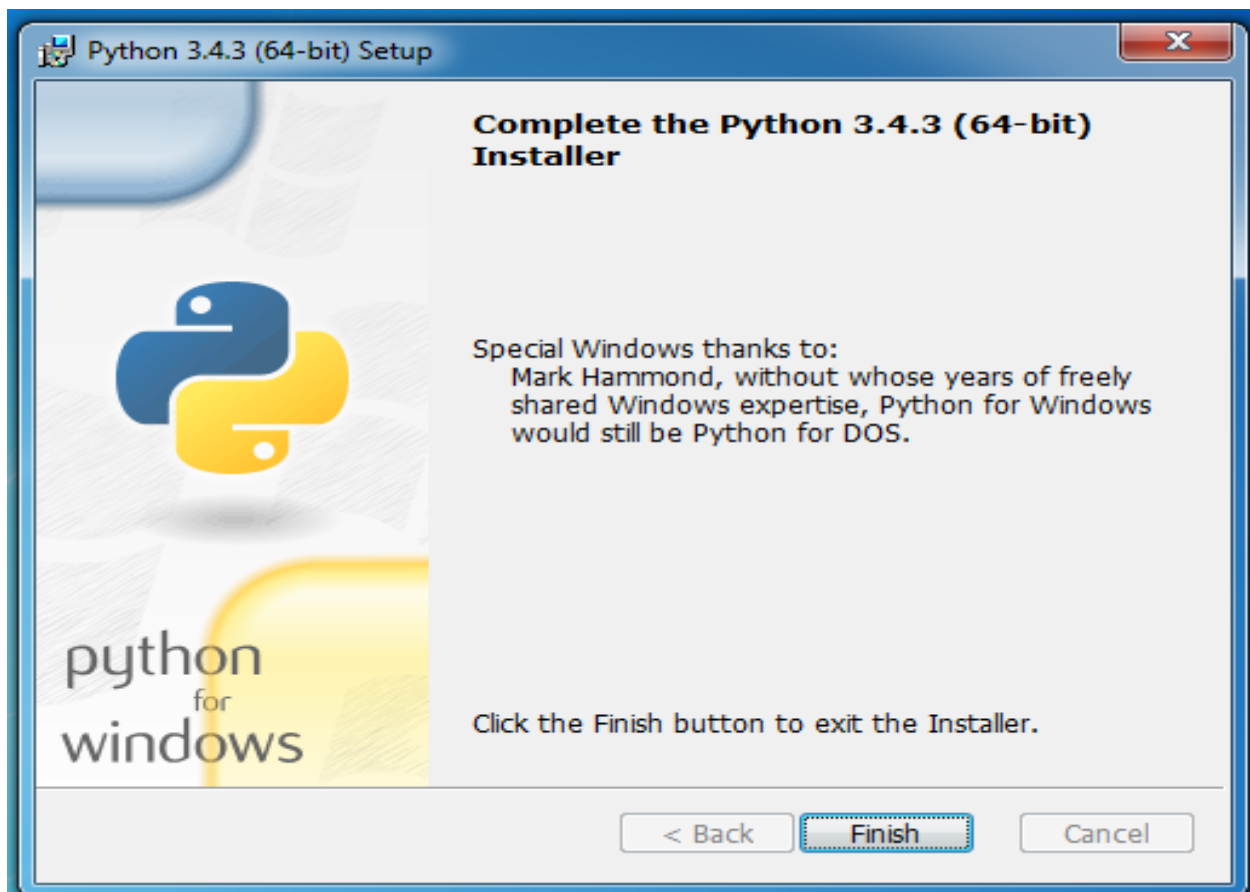
2.3-chizma. Python dasturini o`rnatilish jarayoni.

Python o`rnatilyapti va bir necha sekunddan so`ng quyidagi oyna namoyon bo`ladi:



2.4-chizma. PIP kutubxonasini qo`shish jarayonida hosil boladigan oyna.

Bunda Console rejimida dastur ishga tushib **pip** kutubxonasini qo`shadi.



2.5-chizma. Python dasturini o`rnatish tugallanganligi haqidagi muloqot oynasi.

Va dasturni o`rnatish muvaffaqiyatli tugallandi.

Python tili sintaksisi, asosiy operatorlari.

Python tili sintaksisi o`zi kabi sodda

- Satr oxiri instruksiyaning oxiri hisoblanadi (nuqta vergul shart emas)
- Har bir qator boshidagi bo`sh joy(отступ) muhim ahamiyatga ega. Kiritilgan amallar bo`sh joylarning kattaligiga qarab **bloklarga** birlashadi. Bo`sh joy istalgancha bo`lishi mumkin asosiysi bitta kiritilgan blok chegarasida bo`sh joy bir xil bo`lishi kerak. Noto`g`ri qo`yilgan bo`sh joylar xatolik yuz berishiga olib kelishi mumkin. Bitta probel bilan bo`sh joy hosil qilish yaxshi qaror emas uni o`rniga to`rtta probel yoki Tab belgisini ishlatish kerak.
- Pythonga kiritilgan amallar bir xil shablonda yoziladi. Bunda asosiy amal ikki nuqta bilan tugatiladi va uning orqasidan kiritilgan blok kodi ham joylashadi. Odatda, asosiy amalning ostidagi satr bo`sh joy bilan ajratiladi.

Bir nechta maxsus holatlar

- Bazan bir nechta amalni bitta satrga nuqtali vergul bilan ajratgan holda yozish mumkin.

```
a = 1; b = 2; print(a, b)
```

Buni ko`p ham qo`llamang! Yaxshisi bunday qilmang, o`qishga noqulay.

- Bitta amalni bir nechta satrga yozish mumkin faqat aylana, to`rtburchak va figurali qavslardan foydanish kerak.

```
if (a == 1 and b == 2 and  
c == 3 and d == 4):  
print('spam'*3)
```

Kalit so`zlar

False – yolg`on.

True - rost.

None - “bo`sh” obyekt.

and – mantiqiy VA amali.

with / as – konteks menejeri.

break –tsikldan chiqish.

class – metod va atributlarda iborat.

continue – tsikldan keyingi iteratsiyaga o`tish.

def – funksiyani aniqlash.

del – obyektни yo`qotish.

elif – aks holda, agar.

else – for/else yoki if/elsega qarang.

for – for tsikli.

from – moduldan bir nechta funksiyani import qilish.

if - agar.

import – moduldan import.

is –xotirani bitta joyida 2 ta obyektни jo`natsa bo`ladimi.

lambda –yashirin funksiyani aniqlash.

not –mantiqiy inkor amali.

or –mantiqiy Yoki amali.

while – while tsikli.

Komentariy

Kommentariy. *Kommentariy* # simvolidan keyin yoziladi va dastur kodini o'qiyotgan dasturchi uchun eslatma bo'lib xizmat qiladi. Misol uchun:

```
print('salom dunyo!') # print — bu funksiya
```

yoki:

```
# print — bu funksiya
```

```
print('salom dunyo!')
```

Komentariy dastur kodini o'qiyotganlar uchun foydali bo'ladi va dastur nima qilishini oson tushunishga yordam beradi. Unga yechimdagi muhim joylarni, muhim bo'lgan qismlarni yozish mumkin.

O`zgaruvchilar

Biror ma'lumotni saqlash va uning ustida turli amallarni bajarish uchun bizga o'zgaruvchilar yordam beradi. O'zgaruvchining qiymati, o'z nomi bilan aytib turibdiki, o'zgarishi mumkin. Unda xohlagan qiymatni saqlash mumkin. O'zgaruvchilar kompyuter xotirasidagi joy bo'lib, u yerda siz biror ma'lumotni saqlaysiz. O'zgaruvchining konstantadan farqi, o'zgaruvchiga dastur ishlashi davomida (run time) murojaat qilib, uning qiymatini o'zgartira olamiz. Konstantaga esa oldindan ma'lum bir qiymat beriladi va bu qiymatni o'zgartirib bo'lmaydi.

O'zgaruvchilarni nomlashda quyidagi qoidalarga amal qilish kerak:

- O'zgaruvchining birinchi belgisi alifbo harfi (ASCII simvollar katta va kichik registrda) yoki “_” (ostki chiziq) simvoli bo'lishi mumkin.
- O'zgaruvchilarning qolgan qismi harflardan (ASCII simvollar katta va kichik registrda), “_” (ostki chiziq) simvoli va raqamlardan(0-9) tashkil topishi mumkin.
- O'zgaruvchilar nomlashda katta va kichik registrlar farqlanadi. Masalan, *myname* va *myName* – bular boshqa-boshqa o'zgaruvchi hisoblanadi.
- O'zgaruvchilarni to'g'ri nomlashga misollar: *i*, *_my_name*, *name_23*, *a1b2_c3*
- O'zgaruvchilarni noto'g'ri nomlashga misollar: *2things*, *'*, *my-name*, *>a1b2_c3* va “o'zgaruvchi qo'shtirnoqda”

O'zgaruvchi va konstantalarni qo'llanishiga misol:

```
i = 5
print(i)
i = i + 1
print(i)
s = "Bu ko'p qatorlik satr.
Bu uning ikkinchi qatori."
print(s)
Natija:
5
6
Bu ko'p qatorlik satr.
Bu uning ikkinchi qatori.
```

Yuqoridagi misolda dastlab biz 5 konstanta qiymatini '=' operatori yordamida *i* o'zgaruvchiga o'zlashtirib olamiz.

```
i = 5
```

so'ng *i* o'zgaruvchi qiymatini print funksiyasi orqali ekranga chop etamiz.

```
print(i)
```

i o'zgaruvchining qiymatiga 1 qo'shamiz va o'zgaruvchining o'ziga saqlaymiz.

So'ng *i* o'zgaruvchining qiymatini chop etamiz.

```
i = i + 1
```

```
print(i)
```

Yuqoridagi kabi satr konstanta qiymatini *s* o'zgaruvchiga biriktiramiz va shundan so'ng uni chop etamiz.

```
s = "Bu ko'p qatorlik satr.
```

```
Bu uning ikkinchi qatori."
```

```
print(s)
```

Eslatma: O'zgaruvchilar oddiy qiymat biriktirish bilan ishlatiladi. Hech qanday oldindan e'lon qilib qo'yish talab etilmaydi.

2.2-§. Dastur ishlab chiqish uchun python dasturlash tili imkoniyatlari

Ro`yxat- Pythonda erkin turdagi obyektning o`zgaruvchan qatorlashgan kolleksiyasi hisoblanadi (massivga o`xshash, lekin tiplar har xil bo`lishi mumkin). Ro`yxatlardan foydalanish uchun ularni tuzish kerak. Ro`yxatni har xil yondashuvlar yordamida yaratish mumkin. Masalan har bir iteratsiya qilinadigan obyekt (masalan satrni) Pythonni o`ziga kiritilgan **list** funksiyasi yordamida kiritish mumkin.

```
>>> list('ro`yxat')
['r', 'o', '', 'y', 'x', 'a', 't']
```

Ro`yxatni yana literallar yordamida tuzish mumkin.

```
>>> s=[] #bo`sh ro`yxat
>>> l=['r','o','', 'y',['xat'],2]
>>> s
[]
>>> l
['r', 'o', '', 'y', ['xat'], 2]
```

Misoldan ko`rinadiki ro`yxat istalgancha obyekt dan yoki hech narsadan (bo`sh) tashkil topishi mumkin.

Ro`yxat yaratishning yana bir usuli- ro`yxatlarning generatorlari. Ro`yxat generatori bu- ketma-ketlikni har bir elementiga arifmetik amalni qo`llab yangi ro`yxat tuzish usuli. Generatorlar for sikliga juda o`xshash bo`ladi.

```
>>> list('ro`yxat')
['r', 'o', '', 'y', 'x', 'a', 't']
```

Ro`yxatlar generatorining juda murakkab konstruksiyalari bor.

```
>>> c = [c * 3 for c in 'list' if c != 'i']
>>> c
['lll', 'sss', 'ttt']
>>> c = [c + d for c in 'list' if c != 'i' for d in 'spam' if d != 'a']
>>> c
['ls', 'lp', 'lm', 'ss', 'sp', 'sm', 'ts', 'tp', 'tm']
```

Ro`yxatning funksiya va metodlari

Ro`yxatni yaratgandan so`ng uning ustida turli amallarni bajarish kerak bo`ladi, albatta, buning uchun esa Pythonni o`ziga kiritilgan bir qancha funksiya va metodlar bor.

Metod	Vazifasi
List.append(x)	Ro`yxat oxiridan element qo`shish
List.extend(L)	Oxiriga hamma elementlarni qo`shib list ro`yxatini kengaytiradi.
List.insert(i,x)	i-elementga x qiymatini kiritadi
List.remove(x)	Ro`yxatdan x qiymatga ega elementni o`chiradi
List.pop([i])	Ro`yxatning i-elementini o`chiradi va qaytaradi. Agarda indeks ko`rsatilmagan bo`lsa oxirgi element o`chiriladi
List.index(x,[start],[end])	X qiymatga teng start dan end gacha birinchi elementni qaytaradi
List.count(x)	X qiymatga teng elementlar sonini qaytaradi
List.sort([key=funksiya])	Funksiya asosida ro`yxatni saralaydi
List.reverse()	Ro`yxatni ochadi
List.copy()	Ro`yxatning nusxasi
List.clear()	Ro`yxatni tozalaydi

2.1.1-chizma. Ro`yxat metodlari tasnifi

Ro`yxatda metodlarni qo`llanilishini misollar yordamida ko`rib chiqsak

```

>>> a=[] # bo`sh ro`yxat
>>> a
[]
>>> a.append(3)# ro`yxatga element qo`shish
>>> a
[3]
>>> a=a+[2]
>>> a
[3, 2]
>>> a+= [4]
>>> a
[3, 2, 4]
>>> b=[1,6]
>>> b
[1, 6]
>>> a.extend(b)#ikkita ro`yxatni qo`shish
>>> a
[3, 2, 4, 1, 6]
>>> a+=b
>>> a
[3, 2, 4, 1, 6, 1, 6]
>>> k=['u','f','d','f']
>>> k.count('f')# ro`yxatda f elementi soni
2
>>> k.index('d')# ro`yxatdagi d elementining indeksi
2
>>> k.insert(2,'g')#ro`yxatni 2-indeksiga g elementini qo`shish
>>> k
['u', 'f', 'g', 'd', 'f']
>>> k.pop(2)#ro`yxatdan 2-indeksdagi elementni o`chirib qaytarish
'g'
>>> k
['u', 'f', 'd', 'f']
>>> k.remove('d')#ro`yxatdan d qiymatli elementni o`chirish
>>> k
['u', 'f', 'f']
>>> k.clear()#ro`yxatni tozalash
>>> k
[]

```

Ro`yxatning ishlatilishiga misol: bunda ro`yxatning maksimal elementini topish 2 xil usulda max() standart funksiyasi yordamida va har bir elementni birma-bir solishtirib chiqish orqali yechilgan.

```

""" massivdan maxsimal elementni topish
kiritish:
  1 2 8 4 5
natija:
  maxsimum: A[3]=8
  |maxsimum: A[3]=8
"""

print ( "massiv elementlarini kiriting:" )
A = list( map(int, input().split()) )
N = len(A)

nMax = 0
for i in range(1,N):
  if A[i] > A[nMax]:
    nMax = i
print ( "Maxsimum: A[" , nMax, "]=", A[nMax], sep="" )

M = max(A)
nMax = A.index(M)
print ( "Maxsimum: A[" , nMax, "]=", M, sep = "" );

```

Natija:

```

massiv elementlarini kiriting:
45 16 12 8 9 63
Maxsimum: A[5]=63
Maxsimum: A[5]=63

```

Kortejlar(tuple)

Kortejlar bir nechta ob'ektlarni birgalikda saqlashga xizmat qiladi. Ularni ro'yxatlarga o'xshatish mumkin. Lekin ular ro'yxatlar kabi boy funkcionallikka ega emas. Ularning asosiy jihati qatorlarga o'xshab o'zgarmasliklaridir. Kortej-elementlar orasini vergul bilan ajratish orqali hosil qilinadi. Kortejga ma'no jihatdan o'zgarmas ro'yxat deb ta'rif berdik. Shu o'rinda savol tug'iladi. Ro'yxat bo'lsa kortej nimaga kerak:

1. Turli holatlardan himoyalanih. Bu degani kortej o'zgartirishlardan himoyalangan bo'ladi, rejali (bu yomon) va tasodifiy (bu yaxshi) o'zgarishlardan xalos bo'ladi.
2. Kichik hajm. So'zlar bilan ifodalamasdan.

```

>>> a = (1, 2, 3, 4, 5, 6)
>>> b = [1, 2, 3, 4, 5, 6]
>>> a.__sizeof__()
72
>>>
>>> b.__sizeof__()
88

```

3. Kortejdan lug`at kaliti sifatida foydalanish mumkin:

```

>>> d = {(1, 1, 1) : 1}
>>> d
{(1, 1, 1): 1}
>>> d = {[1, 1, 1] : 1}
Traceback (most recent call last):
  File "<pyshell#7>", line 1, in <module>
    d = {[1, 1, 1] : 1}
TypeError: unhashable type: 'list'

```

Kortej afzalliklari haqida bilib oldik. Endi kortej bilan qanday ishlashni ko`ramiz. Bu xuddi ro`yxatlar bilan ishlashga o`xshaydi. Bo`sh kortejni yaratamiz:

```

>>> a=tuple()#Pythonning standart Tuple funksiyasi yordamida
>>> a
()
>>> a=()# kortej literal yordamida
>>> a
()

```

Bir elementli kortejni yaratamiz:

```

>>> a=('s')
>>> a
's'

```

Stop. Satr paydo bo`ldi. Bizga esa kortej kerak. Qanday qilib kortejga ega bo`lamiz?

```

>>> a=('s',)
>>> a
('s',)

```

Ura! Ishladi! Hamma gap vergulda. Qavuslar bu yerda hech nimani anglatmaydi. Kortejni yana quyidagicha yatish mumkin.

```

>>> a='s',
>>> a
('s',)

```

Lekin qavus qo`yish lozim, ba`zi bir holatlar borki qavuslar juda kerak. Tuple()

funksiyasi yordamida iteratsiyalangan obyekt dan kortej yaratish mumkin:

```
>>> a=tuple('salom, dunyo')
>>> a
('s', 'a', 'l', 'o', 'm', ',', ' ', 'd', 'u', 'n', 'y', 'o')
```

Kortejning elementlariga ham ro`yxatlardagi kabi indeks bo`yicha murojat qilish mumkin (masalan: element nechanchi indeksdaligini bilish uchun). Bunda ham indekslash 0 (nol) dan boshlanadi ya'ni 6 ta elementdan iborat kortejning indeksi 0..5 gacha bo`ladi. Kortej indeksi esa [] operatori orasiga beriladi. Agar bizga kortejning elementlarini teskaridan chiqarish kerak bo`lsa uning indeksini manfiy (masalan: kortej[-5]) beramiz. Quyidagi misolda shularni ko`rib chiqamiz.

```
k=tuple('Salom')
print(k)
print('kortejni birinchi elementi:', k[0])
print('kortejni oxirgi elementi:', k[4])
print()
#kortej elementiga teskari tartibda murojaat

print('kortejni birinchi elementi:', k[-5])
print('kortejni oxirgi elementi:', k[-1])
```

Natija:

```
('S', 'a', 'l', 'o', 'm')
kortejni birinchi elementi: S
kortejni oxirgi elementi: m

kortejni birinchi elementi: S
kortejni oxirgi elementi: m
```

Kortej kesmasi

Kortejning biror qismini bo`lagini ya'ni kesmasini ajratish uchun(masalan: 2 elementidan to 5-chi elementigacha) uning indeksleri orasiga-ikki nuqta(:) qo'yamiz.

```

kortej = ('p','r','o','g','r','a','m','i','s','t')

# 2-chidan 4-chi elemngacha chiqarish:('r', 'o', 'g')
print(kortej[1:4])

# boshidan 2 ta elementni chiqarish:('p', 'r')
print(kortej[:2])

# 8-chi elementdan to oxirigacha chiqarish:('i', 's', 't')
print(kortej[7:])

# boshidan oxirigacha barcha elementlarni
# chiqarish: ('p', 'r', 'o', 'g', 'r', 'a', 'm', 'i', 's', 't')
print(kortej[:])

```

Natija:

```

('r', 'o', 'g')
('p', 'r')
('i', 's', 't')
('p', 'r', 'o', 'g', 'r', 'a', 'm', 'i', 's', 't')
|

```

Kortejni o`zgartirish

Kortejlar ustida amal bajarayotganda “+” operatori orqali ikkita kortejni qo`shishimiz mumkin, va bu konkatenatsiya deb ham ataladi.

Kortej elementlarini takrorlash uchun “*” operatoridan foydalanamiz. “+” va “*” operatorlarini kortejlarda qo`llaganimizdan so`ng yangi ko`rinishdagi kortejga ega bo`lamiz

```

# konkatenatsiyani chiqarish:(1, 2, 3, 4, 5, 6)
print((1, 2, 3) + (4, 5, 6))

# takrorlash:('takror', 'takror', 'takror')
print(("takror",) * 3)

```

Kortejning funksiya va metodlari

Count(x)-kortejdagi x elementi sonini qaytaradi.

Index(x)-kortejdagi x elementining indeksini qaytaradi.

Any()-agar kortej elementi mavjud bo`lsa True qiymat qaytaradi, aks holda (kortej bo`sh bo`lsa) False qiymat qaytaradi.

Max()-kortejning maksimal elementini qaytaradi.

Min()- kortejning minimal elementini qaytaradi.

Len()-kortejning uzunligini qaytaradi.

Sorted()-kortej elementlaridan iborat yangi tartiblangan ro`yxatni qaytaradi.

Sum()-kortej elementlari yig`indisini qaytaradi.

```
k=('2', '12', '9', '78', '15', '12',)
j=tuple()
print(k)
print('kortejdagi x element soni',k.count('12'))
print('kortejdagi x element indeksi',k.index('78'))
print('kortejni tekshirish',any(k))
print('kortejni tekshirish',any(j))
print('max element:',max(k))
print('min element:',min(k))
print('kortej uzunligi:',len(k))

('2', '12', '9', '78', '15', '12')
kortejdagi x element soni 2
kortejdagi x element indeksi 3
kortejni tekshirish True
kortejni tekshirish False
max element: 9
min element: 12
kortej uzunligi: 6
['12', '12', '15', '2', '78', '9']
```

Elementni kortejga tegishli ekanligini tekshirish uchun *in* kalit so`zidan foydalaniladi:

```
kortej=('s','a','l','o','m',)
print('o' in kortej)
```

Natija:

True

Pythondagi lug`atlar kalit bo`yicha kirishga ruxsat etuvchi erkin obyektning tartiblangan jamlanmasi. Ularni yana assotsiativli massivlar yoki hesh jadvallar deb nomlaydilar. Soddaroq qilib aytadigan bo`lsak lug`at xuddi manzillar kitobiga o`xshaydi, ya`ni biror insonning ismini bilgan holda uning manzili yoki u bilan bo`g`lanish ma`lumotlarini olish mumkin. Lug`atlar bilan ishlash uchun ham albatta uni yaratish kerak. Lug`atni yaratishni esa bir necha usullari bor. **Birinchi**dan literal yordamida:

```

>>> d={}
>>> d
{}
>>> d={'dict':1, 'lug`at':2}
>>> d
{'dict': 1, 'lug`at': 2}

```

Ikkinchidan dict() funksiyasi yordamida:

```

>>> d=dict(qisqa='dict', uzun='dictionary')
>>> d
{'uzun': 'dictionary', 'qisqa': 'dict'}
>>> d=dict([(1,1), (2,4)])
>>> d
{1: 1, 2: 4}

```

Uchinchidan fromkeys metodi orqali:

```

>>> d=dict.fromkeys(['a', 'b'])
>>> d
{'a': None, 'b': None}
>>> d=dict.fromkeys(['a', 'b'], 100)
>>> d
{'a': 100, 'b': 100}

```

To`rtinchidan lug`at generatori yoradamida ular ro`yxat generatoriga juda o`xshash:

```

>>> d={a: a**2 for a in range(7)}
>>> d
{0: 0, 1: 1, 2: 4, 3: 9, 4: 16, 5: 25, 6: 36}

```

Lug`at metodlari

Dict.clear()- lug`atni tozalaydi.

Dict.copy()-lug`at nusxasini qaytaradi.

Classmethod **dict.fromkeys(seq[, value])**- Seq dan kalitni va Value qiymatlariga ega bo`lgan lug`atni yaratadi.

Dict.get(key[, default])-kalit qiymatini qaytaradi, lekin u bo`lmasa xatolik beradi, default (jimlikda None) qaytaradi.

Dict.items()-juftliklarni qaytaradi(kalit, qiymat)

Dict.keys()- lug`atdagi kalitlarni qaytaradi

Dict.pop(key[default])-kalitni yo`qotib qiymatni qaytaradi. Agarda kalit bo`lmasa defaultni qaytaradi.

Dict.popitem()- juftlikni o`chirib qaytaradi (kalit, qiymat). Agarda lug`at bo`sh bo`lsa KeyError istisnoli chaqiradi. Esingizda tursin lug`atlar tartibli emas.

Dict.setdefault(key [, default])-kalit qiymatni qaytaradi, lekin u bo`lmasa xatolik bermaydi, default qiymatga ega kalitni yaratadi (jimlikda None).

Dict.update([other])- other dan juftliklarni (kalit, qiymat) kiritib lug`atni to`ldiradi. Mavjud bo`lgan kalitlar qaytadan yoziladilar. None (eski lug`at) qaytaradi.

Dict.values()-lug`atdagi qiymatni qaytaradi.

```
d=dict(ismi='Gulnoza', yoshi='8', maktabi='1')
print()
print('lug`atning qiymati:',dict.values(d))
print()
print('lugatdagi juftliklar yani kalit va uning qiymatlari:',dict.items(d))
print()
print('lugatning kalitlari:',dict.keys(d))
print()
print('lugatning nusxasi:',dict.copy(d))
```

natija:

lug`atning qiymati: dict_values(['Gulnoza', '8', '1'])

lugatdagi juftliklar yani kalit va uning qiymatlari: dict_items([('ismi', 'Gulnoza'), ('yoshi', '8'), ('maktabi', '1')])

lugatning kalitlari: dict_keys(['ismi', 'yoshi', 'maktabi'])

lugatning nusxasi: {'ismi': 'Gulnoza', 'yoshi': '8', 'maktabi': '1'}

To`plam (set & frozenset)

Pythondagi to`plam- tasodifiy tartibda va takrorlanmaydigan elementlardan tashkil topgan “konteyner” deyiladi. To`plamni yaratamiz:

```

>>> a=set()
>>> a
set()
>>> a=set('hello')
>>> a
{'e', 'l', 'h', 'o'}
>>> a={'a', 'b', 'c', 'd'}
>>> a
{'c', 'a', 'b', 'd'}
>>> a={i**2 for i in range(10)} # ko`plik generatori
>>> a
{0, 1, 64, 4, 36, 9, 16, 49, 81, 25}
>>> a={}# bunday qilmang!
>>> type(a)
<class 'dict'>

```

Misoldan ko`rinadiki to`plam lug`atdagi literalga o`xshash literalga egalik qiladi, lekin leteral yordamida bo`sh to`plamni yaratib bo`lmaydi.

To`plamdan takrorlanadigan elementlardan qutulish uchun foydalanish qulay bo`ladi:

```

>>> words=['salom', 'aka', 'salom', 'uka']
>>> set(words)
{'aka', 'uka', 'salom'}

```

To`plamlar bilan bir qancha operatsiyalarni bajarish mumkin: birlashmalarni topish, kesimlarni topish...

Len(s)- to`plamdagi elementlar soni(to`plam hajmi).

X in s- 'x' 's' to`plamga tegishli bo`ladimi yo`qmi shuni tekshiradi

Set.isdisjoint(other)-agarda set va other umumiy elementlarga ega bo`lmasalar rost qiymat qaytaradi.

Set==other- set ning hamma elementlari otherga tegishli bo`ladilar otherni hamma elementlari setga tegishli bo`ladilar.

Set.issubset(other) yoki **set<=other**-set ning hamma elementlari other ga tegishli bo`ladilar.

Set.issuperset(other) yoki **set>=other**-analogik holat.

Set.union(other, ...) yoki **|other|...**-bir qancha to`plamlar birlashmasi.

Set.intersection(other, ...) yoki **&other&...**- kesib olish.

Set.difference(other, ...) yoki **-other-...-other** ga tegishli bo'lmagan set ning hamma elementlar to'plami.

Set.symmetric_difference(other); set^other- birinchi to'plamda uchraydigan, lekin ularning ikkala to'planning kesishmasida uchramaydigan elementlar.

Set.copy-to'plam nusxasi

To'plamni to'g'ridan-to'g'ri o'zgartiradigan operatsiyalar

Set.update(other, ...); set|=other| ...- to'plam birlashmasi

Set.intersection_update(other, ...); set&=other&...- to'plam kesishmasi

Set.difference_update(other, ...); set -= other | ...-to'plam ayirmasi

Set.symmetric_difference_update(other); set ^= other- birinchi to'plamda uchraydigan, lekin ularning ikkala to'planning kesishmasida uchramaydigan elementlar tashkil topgan to'plam.

Set.add(elem)- to'plamga element qo'shadi.

Set.remove(elem)- to'plamdagi elementni o'chiradi. Agarda ko'rsatilgan element to'plamda mavjud bo'lmasa `KeyError` ni qaytaradi.

Set.discard(elem)- gar to'plamda ko'rsatilgan element bo'lsa uni o'chiradi.

Set.pop()- to'plamdagi birinchi elementni o'chiradi, lekin to'plam elementlari tartib bilan joylashmagani uchun birinchi element qaysiligini aniq ko'rsatib bo'lmaydi.

Set.clear()- to'plamni tozaydi.

Modul tushunchasi, standart kutubxonalar, sys va copy moduli.

Standart kutubxona modullarini o'rganishdan oldin Pythonda modul tushunchasiga aniqlik kiritib olish lozim.

Python tilida bir xil vazifani bajaruvchi modullar yig'indisini bitta paketga joylashtirish mumkin. Shunday paketlardan biri sifatida XML paketini misol qilib keltirish mumkin. Ushbu paket XML ning har xil aspektlarini qayta ishlashga mo'ljallangan modullardan tashkil topgan.

Python tilda dastur tuzishda modul atributlari modulda aniqlangan nomlar bo'lgan obyekt modul sifatida taqdim etiladi.

```
>>> import datetime
>>> d1=datetime.date(2017,11,20)
..
```

Bu misolda datetime moduli import qilinayapti. Import operatorining ishi natijasida mazkur nomlar kengligida datetime nomi bilan obyekt paydo bo`lyapti.

Python tilida modullar oddiy (Pythonda yozilgan) va kengaytiriladigan ya'ni boshqa tilda yoziladigan masalan Python interpretatori yozilgan C dasturlash tilida yozilgan modullarga bo`linadi. Foydalanuvchi nuqtai nazarida ular ishlash tezligi bilan farq qiladi. Standart kutubxonada modul 2 xil variantda bo`ladi: Pytonda yozilgan yoki C da. Bunga misol sifatida *pickle* va *cpickle* modullarini keltirish mumkin. Odatda Pythonda oddiy modullar kengaytirilgan modullarga nisbatan ishlatishda qulay hisoblanadi.

Modul funksiyalaridan foydalanish uchun uni boshqa dasturdan yuklash (импортировать) mumkin. Dastlab standart kutubxonalar modullarini qanday ishlatishni ko`raylik.

```
1 import sys
2
3 print("Buyruqlar qatori argumentlari:")
4
5 for i in sys.argv:
6
7     print(i)
8
9 print("\n\n O`zgaruvchi PYTHONPATH qiymati", sys.path, "\n")
```

Natija:

```
1. $ Python3 using_sys.py biz argumentlarmiz
2. Buyruqlar qatori argumentlari:
3. using_sys.py
4. biz
5. argumentlarmiz
6. O`zgaruvchi PYTHONPATH qiymati ['/home/user/python darslari/7-
dars', '/usr/lib/python3.4', '/usr/lib/python3.4/plat-x86_64-
linux-gnu', '/usr/lib/python3.4/lib-dynload',
'/usr/local/lib/python3.4/dist-packages', '/usr/lib/python3/dist-
packages']
```

Bu misolda dastlab sys moduli import buyrug`i yordamida yuklanyapti. sys moduli Python interpretatoriga va uning muxitiga ya'ni tizimiga (system)

tegishli funksiyalardan tashkil topgan. Python **import sys** buyrug`ini bajarayotganda sys modulini qidiradi. Bu holatda sys standart modullardan biri bo`lganligi uchun, Python uni qayerdan izlash kerakligini biladi. Agar bu oddiy modul, ya`ni Pythonda yozilgan modul bo`lganida edi, u holda Python uni **sys.path** ko`rsatilgan kataloglardan izlagan bo`lar edi. Agar modul topilsa, undagi buyruqlar bajariladi va bu modul foydalanishga (доступным) shay holatga keladi. sys modulidagi argv o`zgaruvchisiga murojat qilish nuqta orqali amalga oshiriladi ya`ni sys.argv. Bunday ifodalashning afzalligi dasturda ishlatilishi mumkin bo`lgan argv o`zgaruvchisi bilan xatoliklar yuz bermaydi. sys.argv qatorlar ro`yxati hisoblanadi. U buyruqlar qatori argumentlaridan ya`ni buyruqlar qatoridan dasturga uzatilgan argumentlardan tashkil topgan.

Python dasturida modullarni ulash *import* operatori orqali amalga oshirilishi yuqoridagi misolda ko`rdik. Modullarni ulashini ham 2 xil shakli mavjud: birinchisi *import* operatori orqali bo`lsa, ikkinchisi *from-import* operatori orqalidir. **From... import ... operatori-** argv o`zgaruvchisini dasturga to`g`ridan-to`g`ri yuklash uchun hamda har doim sys.argv deb yozmaslik uchun, from sys import argv ifodasidan foydalanish mumkin. sys modulida ishlatiladigan hamma nomlarni yuklash uchun “from sys import *” buyrug`ini bajarish mumkin.

```
1. from math import *
2.
3. n = int(input('Biror son kiriting: '))
4.
5. print(sqrt(n))
```

Standart kutubxonalar

Python tili standart kutubxonasining modullarini shartli ravishda mavzular bo`yicha quyidagi guruhlariga ajratish mumkin:

1. Bajarish davri servislari. Modular: sys, atexit, copy, traceback, math, cmath, random, time, calendar, datetime, sets, array, struct, itertools, locale, gettext.
2. Siklni qayta ishlashni qo`llab-quvvatlovchi. Modular: pdb, hotshot, profile, unittest, pydoc. Paketlar: docutils, distutils.

3. OS (fayllar, protseslar) bilan ishlash. Modullar: os, os.path, getopt, glob, popen2, shutil, select, signal, stat, tempfile.
4. Matnlarni qayta ishlovchi. Modullar: string, re, StringIO, codecs, difflib, mmap, sgmlib, htmlib, htmlentitydefs. Paket: xml.
5. Ko`p oqimli hisoblashlar. Modullar: threading, thread, Queue.
6. Ma'lumotlarni saqlash. Arxivlash. Modullar: pickle, shelve, anydbm, gdbm, gzip, zlib, zipfile, bz2, csv, tarfile.
7. Platformaga tobe modullar. UNIX uchun: commands, pwd, grp, fcntl, resource, termios, readline, rlcompleter. Windows uchun: msvcrt, _winreg, winsound.
8. Tarmoqni qo`llab-quvvatlash. Internet protokollari. Modullar: cgi, Cookie, urllib, urlparse, httplib, smtplib, poplib, telnetlib, socket, asyncore. Serverlarga misollar: SocketServer, BaseHTTPServer, xmlrpclib, asynchat.
9. Internetni qo`llab-quvvatlash. Ma'lumotlar formatlari. Modullar: quopri, uu, base64, binhex, binascii, rfc822, mimetools, MimeWriter, multifile, mailbox. Paket: email.
10. Python uchun. Modullar: parser, symbol, token, keyword, inspect, tokenize, pycbr, py_compile, compileall, dis, compiler.
11. Grafik interfeys. Modul: Tkinter.

Ko`pincha modullar o`zida bir yoki bir nechta sinflarni saqlaydilar. Bu sinflar yordamida kerakli tipdagi obyekt yaratiladi, lekin gap moduldagi nomlar haqida emas, aksincha shu obtekt atributi haqida boradi. Bir nechta modullar faqat erkin obyektlar ustida ishlash uchun umumiy bo`lgan funksiyalardan iborat bo`ladilar.

Sys moduli

Sys moduli Python interpretatorida dasturni bajaruvchi muhitdir. Quyida bu modulni eng ko`p qo`llaniladigan obyektlari keltirilgan:

Exit([c])- dasturdan chiqish. Tugatishning raqamli kodini yuborish mumkin: agarda dasturni tugatish muvafaqiyatli amalga oshsa 0 ni yuboradi, aksincha bo`lsa ya'ni xatolik yuz bersa boshqa raqamlarni yuboradi.

Argv- buyruqlar qatori argumentlari ro`yxati. Oddiy holatda sys.argv[0] buyruqlar qatoriga ishga tushirilgan dastur nomini va boshqa parametrlar yuboriladi.

Platform- interpretator ishlaydigan platforma.

Stdin, stdout, stderr- standart kiritish, chiqarish, xalolarni chiqarish. Ochiq faylli obyektlar.

Version- interpretator versiyasi.

Sereursionlimit(limit)- rekursiv chaqirishlarni maksimal kiritish darajasini o`rnatadi.

Exc_info()-kiritish-chiqarish istisnosi haqida ma'lumot.

Copy moduli

Bu modul obyektlarni nusxalashga mo`ljallangan funksiyalarga ega. Boshida Pyhtonda sal sarosimaga solish uchun “paradoks” ni ko`rib chiqish tavsiya etiladi.

```
lst1 = [0, 0, 0]
lst = [lst1] * 3
print(lst)
lst[0][1] = 1
print (lst)
```

Va biz kutmagan natija paydo bo`ladi:

```
[[0, 0, 0], [0, 0, 0], [0, 0, 0]]
[[0, 1, 0], [0, 1, 0], [0, 1, 0]]
```

Gap shundaki bu yerda lst ro`yxati shu ro`yxatning izohiga ega. Agarda rostdan ham ro`yxatni ko`paytirmoqchi bo`lsak, copy modulidagi copy() funksiyasini qo`llash kerak.

```
from copy import copy
lst1 = [0, 0, 0]
lst = [copy(lst1) for i in range(3)]
print (lst)
lst[0][1] = 1
print (lst)
```

Endi kutilgan natija paydo bo`ladi:

```
[[0, 0, 0], [0, 0, 0], [0, 0, 0]]
```

[[0, 1, 0], [0, 0, 0], [0, 0, 0]]

Copy modulida yuqori aniqalikda nusxalash uchun `deepcopy()` funksiyasi bor bu funksiya yordamida obektlar butun imkoniyati bilan rekursiv nusxalanadi.

Math, cmath, random va os moduli.

Math va cmath modullarida haqiqiy va kompleksli argumentlar uchun matematik funksiyalar to'plangan. Bu C tilida foydalaniladigan funksiyalar. Quyida math modulining funksiyalari keltirilgan. Qayerda z harfi bilan argumentga belgilash kiritilgan bo'lsa, u cmath modulidagi analogik funksiya ham shunday belgilanishini bildiradi.

Acos(z)- arkkosinus z.

Asin(z)- arksinus z.

Atan(z)- arktangens z.

Atan2(y, x)- $\text{atan}(y/x)$.

Ceil(x)- x ga teng yoki katta eng kichik butun son.

Cos(z)- kosinus z.

Cosh(x)- giperbolik x kosinusi.

e- e konstantasi.

Exp(z)- eksponenta (bu degani e^{**z})

Fabs(x)- x absolute raqami.

Floor(x)- xga teng yoki kichik eng katta butun son

Fmod(x,y)- x ni y ga bo'lgandagi qoldiq qism.

Frexp(x)- mantisa va tartibni (m, i) juftligi kabi qaytaradi, m- o'zgaruvchan nuqtali son, i esa- $x=m*2^{**i}$ ga teng butun son bo'ladi. Agarda 0-(0,0) qaytarsa boshqa paytda $0.5 \leq \text{abs}(m) < 1.0$ bo'ladi.

Factorial(x)- x ning faktoriali. $N!=1*2*3*...*n$

Hypot(x,y)- $\text{sqrt}(x*x+y*y)$

Ldexp(m,i)- $m*(2^{**i})$.

Log(z)- natural logarifm z.

Log10(z)- o'nlik logarifm z.

Log2(z)-logarifm ikki asosga ko'ra z.

Modf(x)- (y,q) juftlikda x ning butun va kasr qismini qaytaradi.

p-pi konstantasi.

Pow(x,y)- $x^{**}y$.

Sin(z)- z ning sinusi.

Sinh(z)- z ning giperbolik sinusi.

Sqrt(z)- z ning kvadrat ildizi.

Tan(z)- z ning tangensi.

Tanh(z)- z ning giperbolik tangensi.

Trunc(x)- x haqiqiy sonning butun qismini qaytaradi.

degrees(x)-x ni radiandan gradusga o`tkazish.

radians(x)- x ni gradusdan radianga o`tkazish.

```
>>> import math
>>> math.factorial(10)
3628800
>>> math.hypot(3,4)
5.0
>>> math.ceil(10.8)
11
>>> math.ceil(10.2)
11
>>> math.log10(100)
2.0
>>> math.log2(64)
6.0
>>> math.trunc(10.6)
10
```

Random moduli

Bu modul har xil taqsimotlar uchun tasodifiy raqamlarni generatsiya qiladi.

Eng ko`p qo`llaniladigan funksiyalari:

Random()-[0.0, 1.0) yarim ochiq diapozondagi tasodifiy sonlarni generatsiya qiladi.

Choice(s)- s ketma- ketlikdan tasodifiy elementni tanlab oladi.

Shuffle(s)- s o`zgaruvchan ketma-ketlik elementlarini joyiga joylashtiradi.

Randrange([start], stop, [step])- rene(start, stop, step) diapozondagi tasodifiy butun raqamni chiqaradi. Choice(range(start, stop, step)) ga analogik holatda.

Normalvariate(mu, sigma)- normal holatda taqsimlangan ketma-ketlikdan raqamni chiqaradi. Bu yerda mu- o`rtacha, sigma-o`rta kvadratli ($\sigma > 0$) sonlar.

Boshqa funksiyalar va uning parametrlarini hujjatlashdan aniqlab olish mumkin. Modulda qandaydir holatga tasodifiy raqamlar generatorini joylashtirishga imkon beruvchi seed(n) funksiyasi ham mavjud. Masalan: agarda bitta tasodifiy raqamlar ketma-ketligidan ko`p marta foydalanishga ehtiyoj sezilsa.

Time va Sets moduli

Time moduli joriy vaqtni olish uchun va vaqt formatlarini o`zgartirish uchun fuksiyalarni taqdim etadi.

Sets moduli to`plamlar uchun ko`rsatgichlar tipini amalga oshiradi. Quyidagi misol bu moduldan qanday foydalanish mumkinligini ko`rsatadi. Su o`rinda bilishimiz kerakki, Python 2.4 va undan yuqori versiyalarda set tipi sets o`rniga kiritilgan.

```
import sets
A = sets.Set([1, 2, 3])
B = sets.Set([2, 3, 4])
print (A | B, A & B, A - B, A ^ B)
for i in A:
    if I in B: print (i,)
```

Natijada:

```
Set([1, 2, 3, 4]) Set([2, 3]) Set([1]) Set([1, 4])
2 3
```

Array va struct modullari

Bu modullar past darajali massiv va korsatgichlar tuzilmasini amalgam oshiradi. Ularning asosiy vazifasi- ko`rsatgichlarning ikkilamchi formatlarini ko`rib chiqish.

Os moduli

Os moduli-har xil operatsion sistemalarning o`ziga xos xususiyatlari bilan ishlovchi kategoriyadagi asosiy modul hisoblanadi. Bu modul funksiyalari ko`plab operatsion sistemalarda ishlaydilar. Kataloglarni bo`luvchi os moduli va u bilan bog`liq bo`lgan ifodalar konstanta ko`rinishida berilgan.

Konstanta	Vazifasi
Os.curdir	Joriy katalog

Os.pardir	Bosh katalog
Os.sep	Yo`lning elementlarini taqsimlovchi
Os.altsep	Boshqa yo`lning elementlarini taqsimlovchi
Os.pathsep	Yo`llar ro`yxatidagi yo`llarni taqsimlovchi
Os.defpath	Yashirin yo`llar ro`yxati
Os.linesep	Satrni yakunlovchi belgi

2.4.1-chizma. Kataloglarni bo`luvchi os moduli ifodalari konstanta ko`rinishida

Pythondagi dastur operatsion tizimda alohida jarayon ko`rinishida ishlaydi. Os modulining funksiyalari protsesda, muhitda bajariladigan turli xildagi ahamiyatga ega bo`lgan kirishlarga ruxsat etadilar. Os modulining eng muhim ruxsat etuvchi obyektlaridan biri deb environ o`rab oluvchi muhiti o`zgaruvchilarning lug`ati hisoblanadi. Masalan o`rab oluvchi muhit o`zgaruvchilar yordamida web server CGI-ssenariyga bir qancha parametrlarni o`tkazadi. Quyidagi misolda PATH o`rab oluvchi muhiti o`zgaruvchini olish mumkin:

```
import os
PATH=os.environ['PATH']
```

Funksiyalarning katta qismi fayllar va kataloglar bilan ishlashga mo`ljallangan. Quyida UNIX va Windows OT lar uchun ruxsat etilgan funksiyalar taqdim etilgan:

Access(path, flags)- path nomli fayl yoki catalog ruxsat etish(доступ) ni tekshiradi. Buyurma qilishga rucsatning tartibi flags raqami bilan belgilanadi. U esa yaratilgan kombinatsiyalar os.F_OK (fayl mavjud), os.R_OK (fayldan o`qish mumkin), os.W_OK (faylga yozish mumkin) va os.X_OK (fayllarni bajarishni, katalogni ko`rib chiqish mumkin) bayroqlari bilan belgilash mumkin.

Chdir(path)- path ni joriy ishchi katalog qiladi.

Getcwd()- joriy ishchi catalog.

Chmod(path, mode)- mode ga path bo`lgan ruxsat etish rejimini belgilaydi. Ruxsat etish tartibi bayroqlarni kombinatsiya qilib belgilashi mumkin. Bu ishda chmod() harakatda bo`lgan tartibni to`ldirmaydi, uni yangidan belgilamaydi, uni yangidan belgilaydi.

Listdir(dir)- dir katalogidagi fayllar ro`yxatini qaytaradi. Ro`yxatga maxsus belgilar “.” va “..” kirmaydi.

Mkdir(path [, mode])- path katalogini tuzadi. Jimlik holatida mode tartibi 0777 ga teng bo`ladi, bu degani S_IRWXU|S_IRWXG|S_IRWXO agarda stat moduli konstantalari bilan foydalansak.

```
import os
```

```
os.mkdir('C:\Users\Guljakhon\Desktop\Новая папка\katalog\dir2')
```

```
#ko`rsatilgan manzilda dir2 nomli yangi katalog yaratadi.
```

```
import os
```

```
os.mkdir('./dir2')
```

```
#joriy manzilda dir2 nomli yangi catalog yaratadi.
```

Makedirs(path [,mode])- hamma kataloglarni yaratuvchi, agarda ular mavjud bo`lmasalar mkdir() analogi oxirgi katalog mavjud bo`lgandan so`ng mustasnoni ishga tushiradi.

Remove(path), unlink(path)- path katalogini yo`qotadi. Kataloglarni yo`qotish uchun rmdir() va removedirs() dan foydalanadi.

Rmdir(path)- path nomli bo`sh katalogni yo`qotadi.

Removedirs(path)- birinchi bo`sh bo`lgan kataloggacha pathni yo`q qiladi. Agarda yo`lda eng oxirgi kiritilgan katalog osti bo`sh bo`lmasa OSError mustasnosini ishga tushiradi.

Rename(src, dst)- src fayli yoki katalogini dst deb qayta nomlaydi.

Renames(src, dst)- rename() analogi dst yo`li uchun kerakli kataloglarni yaratadi va src yo`lining bo`sh kataloglarini yo`qotadi.

Stat(path)- path haqidagi malumotni o`nta elementlik kortej shaklida qaytaradi. Kortej elementlariga kirish uchun stat moduli konstantalaridan foydalanish mumkin. Masalan stat.ST_MTIME (faylning oxirgi modifikatsiyasi vaqti).

Utime(path, times)- oxirgi modifikatsiya (mtime) va faylga kirishga ruxsat(atime) larini belgilaydi. Boshqa holatlarda times ikki elementli kortej (atime, mtime) sifatida ko`rib chiqiladi. Qaysidir faylni atime va mtime ni olish uchun stat() va stat modulining konstantalarini barobar ishga tushirib olish mumkin.

Os moduli protsesslar bilan ishlash uchun quyidagi funksiyalarni taqdim etadi (ular ham UNIX hamda windowsda ishlaydilar).

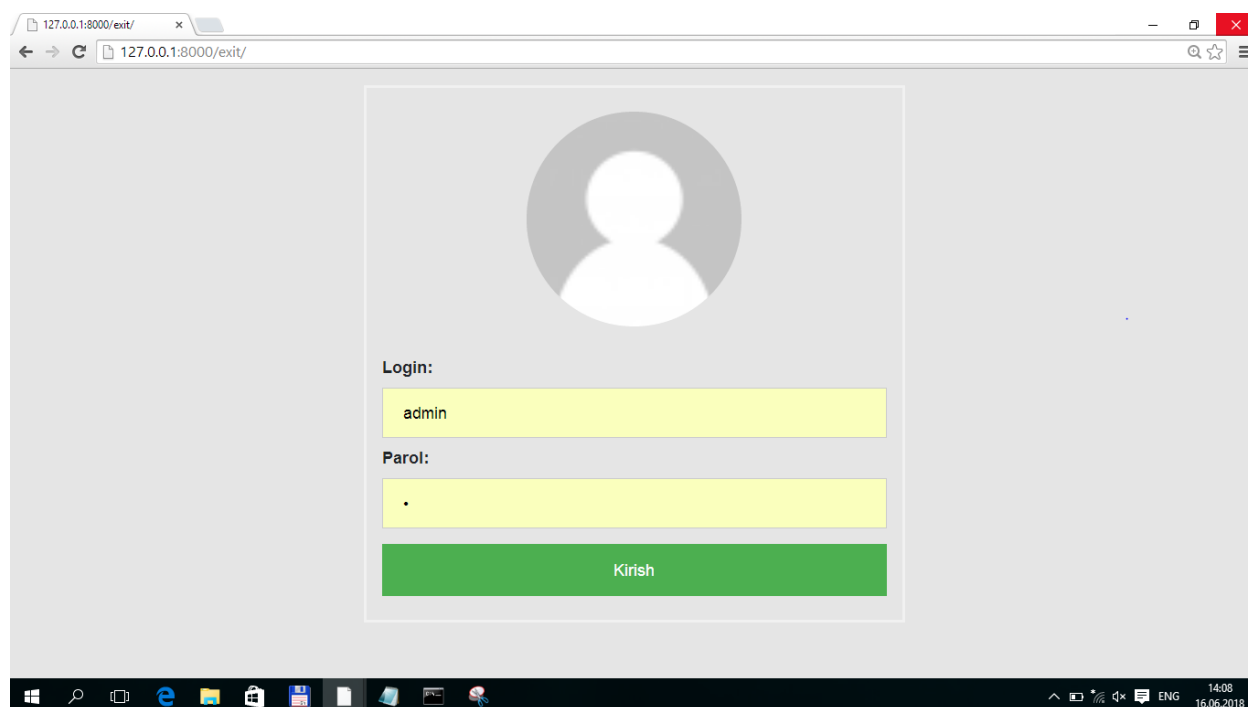
System(cmd)- alohida oynada cmd buyruqlar satrini bajaradi. U C tilining system kutubxonasi chqirig`iga analogik bo`ladi. Qaytarilgan qiymat foydalanadigan platformadan tobe bo`ladi.

Times()- beshta elementdan iborat bo`lgan kortejni qaytaradi. U ish jarayoni vaqtini lahzalarda ko`rsatadi, qo`shimcha protsesslar vaqtini, qo`shimcha protsesslarning axborot tizimlari vaqtini, va o`tgan zamonda qotib qolgan vaqtni ko`rsatadi (masalan tizim ishga tushgan paytdan).

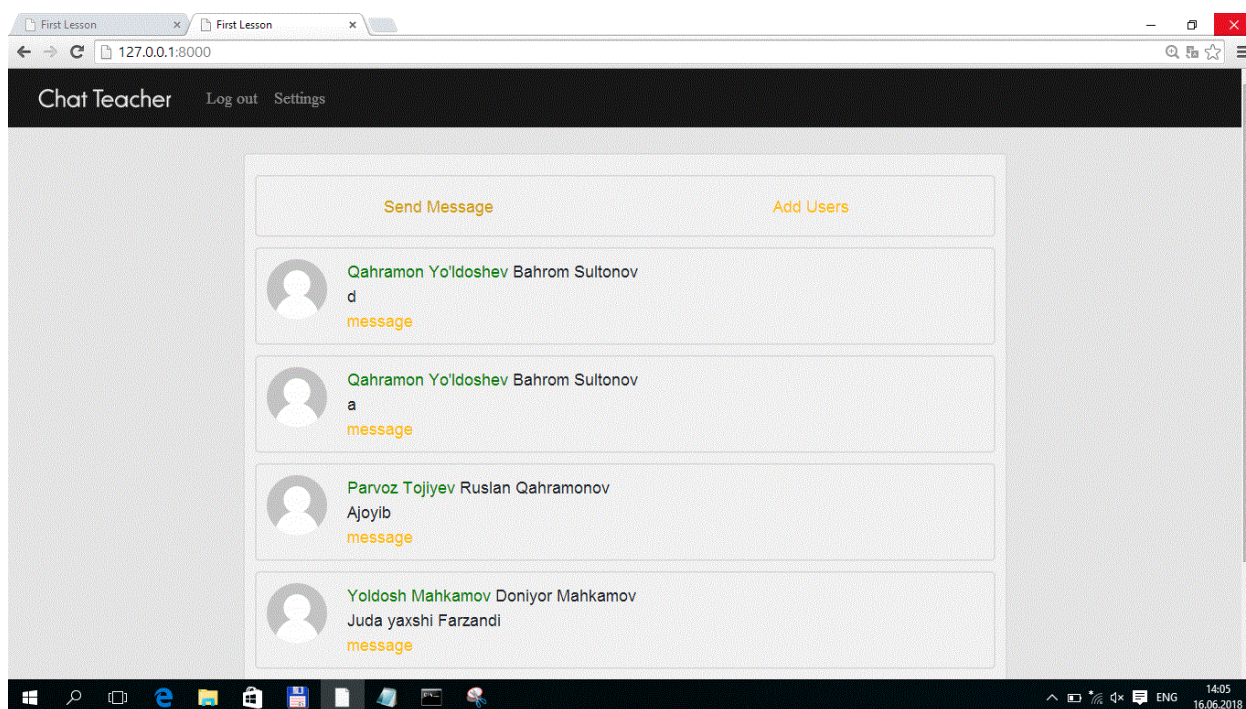
Getloadavg()- coo, uchta qiymatlik kortejni qaytaradi.

2.3-§. Foydalanuvchi uchun dastur yo`riqnomasi

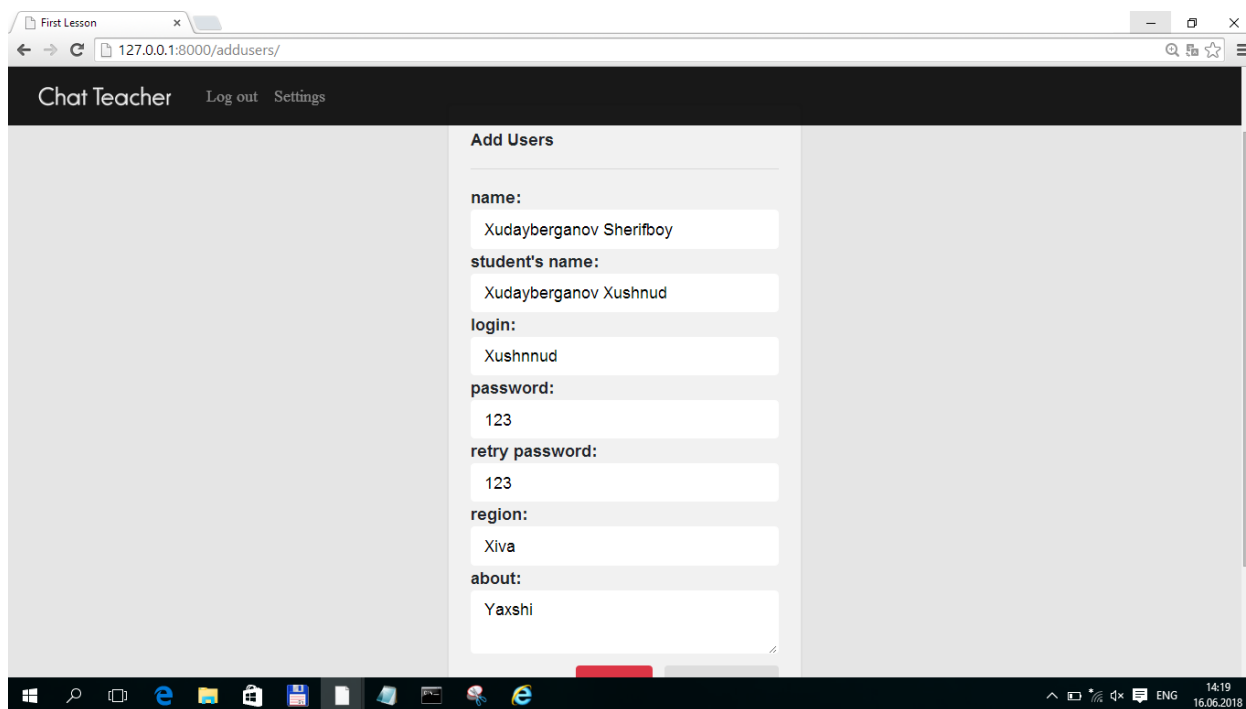
Yaratilgan dasturning bosh oynasidan guruh rahbari o`zining login parol yordamida guruh talabalarining ota-onalari bilan almashgan xabarlarni ko`rib chiqadi.



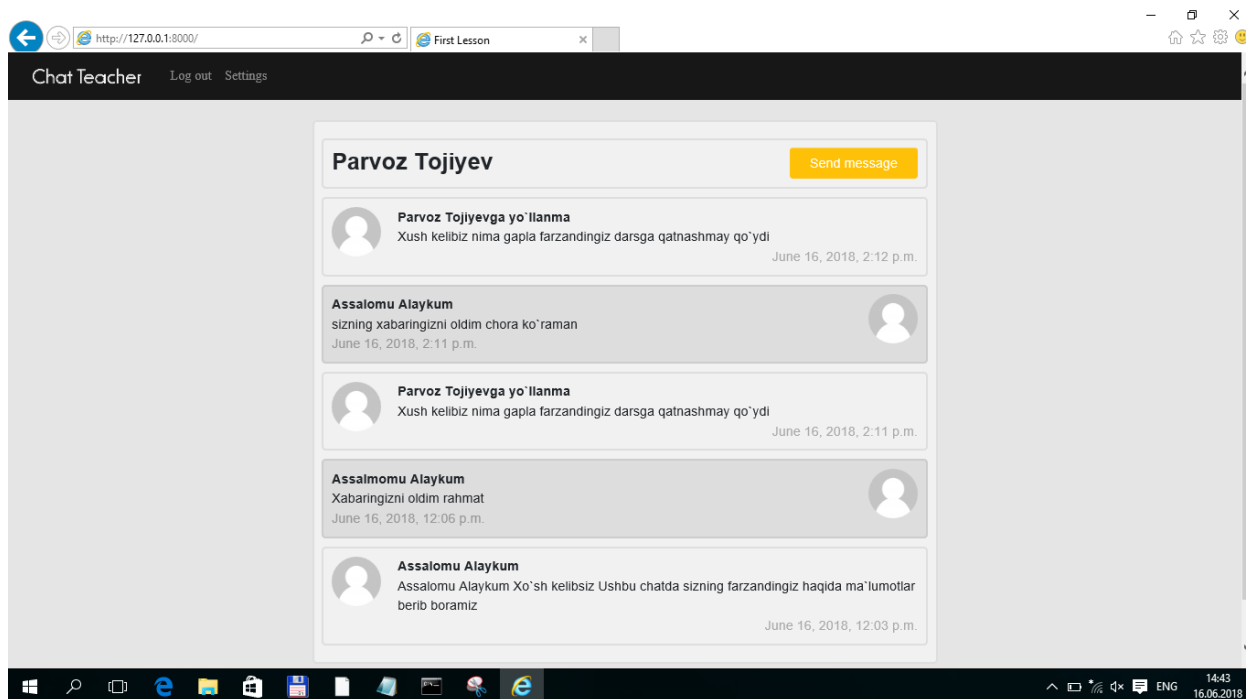
Bu yerda guruh rahbarining ishchi oynasida ota-onalar bilan yozishgan sms xabarlar bo'limi va foydalanuvchi qo'shish bo'limi mavjud bo'lib, quyidagi ko'rinishga ega:



Foydalanuvchi qo'shish bo'limida guruh talabalarini ota-onalarini ism-familiyasini, talabaning ism-familiyasi, login-parolini va boshqa ma'lumotlarni kiritish mumkin.



Bu yerda o'z navbatida mijoz ham ya'ni talabalarning ota-onalari ham o'zlariga tegishli login-parol yordamida o'zining "chat" qismini ko'rib chiqishi mumkin bo'ladi. Albatta bu yerda xabarning qachon kelgani, kimdan kelganligini va guruh rahbariga javob qaytarib yuborishi mumkin bo'ladi.



XULOSA.

Xulosa o'rnida shuni aytish joizki bittiruv malakaviy ishini ishlash natijasida ta'lim muassasalarida talabalarning ota-onalari bilan muloqot dasturini ishlash jarayonida bu dasturning yoshlar tarbiyasiga qanchalik ijobiy ta'sir qilishi ko'rib chiqildi. Undan tashqari quyidagi vazifalar hal qilindi:

- Ota-onalar bilan muloqot dasturiy ta'minotning mobil ilovasi ishlab chiqildi;
- Bu ilovani ishlash jarayonida java dasturlash tilining yanada yangi imkoniyatlari o'rganildi;
- Dastur ni ishlab chiqish jarayonida Mysql dasturlash tilida ma'lumotlar bazisi yaratildi;
- Dasturiy ta'minotga guruh rahbarlari va guruhlar to'g'risida ma'lumotlar joylashtirildi;

Foydalanilgan adabiyotlar:

1. S.S.Qosimov. Axborot texnologiyalari. Toshkent "Aloqachi" 2006.
2. Спейнауэр С., Куэрсиа В. Справочник Web-мастера. - К: "ВНУ", 1997. - 368 с.
3. Яргер Р., Риз Дж., Кинг Т. MySQL и mSQL. Базы данных для небольших предприятий и Интернета. - СПб: Символ-Плюс, 2000 - 560 с.
4. Хилайер С., Мизик Д. Программирование Active Server Pages. - М: "Русская редакция", 1999. - 296 с.
5. Xaitov F.N., Yusupov R.M., Botirov D.B., Sattarov A.R, Shukurov E.X.
6. Web texnologiyalar. Jizzah. 2005
7. T. Staufer. Sozdanie web-stranits. Samouchitel. – «Piter», Sankt-Peterburg, 2003 g.
8. A. Goncharev. HTML. Samouchitel. – «Piter», Sankt-Peterburg, 2001 g.
9. Allen Vayk. JavaScript. Entsiklopediya polzovatelya: per. s. ang. – «TID» «DS», Kiev, 2001 g.
10. www.Arxitektura.uz
11. www.Lex.uz
12. www.Davarx.uz
13. www.MFA.uz
14. www.Xorazm.uz
15. www.Gov.uz
16. www.Akt.uz