Boshqa dasturlash tilida	Python dasturlash tilida
a=1 b=2 tt=a a=b b=tt print(a,b)	a=1 b=2 a, b=b, a print(a, b)
21	21

SAVOL VA TOPSHIRIQLAR

1. *Qaysi operatorlar sikl ishini boshqaradi?*
2. *break operatorining vazifasi nima?*
3. *continue operatorining vazifasi nima?*
4. *O‘zgaruvchilarning o‘zaro qiymat almashtirishi qanday amalga oshiriladi?*

UY VAZIFASI

1. Qo‘shish, ayirish, ko‘paytirish va bo‘lish amallaridan iborat sodda kalkulyator dasturini tuzing.

2. Foydalanuvchi tomonidan kiritilgan sonlar yig‘indisini hisoblash dasturini tuzing. Agar manfiy son kiritilsa, sikl o‘z ishini to‘xtatishi lozim.

56–57-DARSLAR. QISM DASTURLAR: FUNKSIYALAR VA PROTSEDURALAR

Dastur tuzish jarayonida ma’lum bir amallar majmuini dasturning turli qismlarida takrorlashga to‘g‘ri keladi. Dasturning mana shu amallar majmuini o‘z ichiga olgan qismi **qism dastur** deb ataladi. Qism dasturlar ma’lum bir vazifani bajaradi, lekin alohida tizimni tashkil etmaydi.

Qism dasturga murojaat etilganda, unga murojaat etgan asosiy dastur to‘xtaydi va boshqaruv qism dasturga o‘tadi. Qism dastur bajarilishi tugaganidan so‘ng, boshqaruv yana asosiy dasturga qaytadi.

BUNI BILASIZMI?

1. *Dasturlashda qism dasturlardan qanday foydalaniladi?*
2. *Qism dasturlar qanday ishlaydi?*
3. *Funksiya nima va u qanday ishlaydi?*
4. *Protsedura nima va u qanday ishlaydi?*

Asosiy dasturda qism dasturlarni chaqirish quyidagi imkoniyatlarni beradi:

- Qism dastur zarurat tug‘ilganda chaqiriladi. U ayni bir kodni bir necha marta yozish zaruratini bartaraf qilib, butun dastur davomida ko‘p marta foydalanilishi mumkin. Bu kodning blokliligini oshiradi, tushunishni osonlashtiradi va xatolarni topishda yordam beradi.

- Xato bor yoki yo‘qligini bitta kod blokining o‘zida tekshirsa bo‘ladi. Agar xato qism dasturda bo‘lsa, faqat qism dasturning o‘zini tuzatishga zarurat tug‘iladi. Agar qism dasturdan foydalanmasdan, kod bir necha joyda takror-takror yozilsa, u holda butun dastur bo‘ylab xatolarni qidirishga to‘gri keladi.
- Kodni faqat bitta joyda yangilash kerak bo‘ladi: Kiritilgan barcha tuzatishlar qism dastur chaqirilishi bilan amal qila boshlaydi.

Qism dasturning turlari

- **Funksiya** – ma’lum bir vazifani bajaruvchi, qandaydir nomga ega, bir yoki bir necha qiymatni qabul qiluvchi, ishni tugatganidan keyin esa asosiy dasturga bir yoki bir necha natija qiymatlarni qaytaruvchi qism dastur.
- **Protsedura** – funksiyaga o‘xshash ko‘p marta foydalanilishi mumkin bo‘lgan qism dastur bo‘lib, yagona farqli jihati hech qanday qiymatni qaytarmaydi.

Python dasturlash tilining har xil masalalarni yechishga mo‘ljallangan bir necha foydali standart funksiyalari mavjud.

Standart funksiyalar

print() – foydalanuvchi uchun ma’lumotlarni chiqaradi. Masalan, turli ma’lumotlar va hisoblash natijalarini.

input() – print() funksiyasining zidi, foydalanuvchilar kiritgan ma’lumotlarni dasturga uzatadi.

randint() – tasodifiy sonni chiqaradi. Masalan, dasturda tasodifiy son kerak bo‘lib qolganda ishlatiladi.

Keyinchalik standart funksiyalar bilan batafsilroq tanishib chiqamiz.

Funksiyani e’lon qilish va chaqirish

Har bir yaratilgan qism dasturga, xususan, funksiya hamda protseduraga albatta nom berish kerak va bu nom Pythonda define (ing. define – aniqlash) so‘zidan olingan def kalit so‘zi bilan boshlanadi.

Sintaksisi:
def *funksiya_nomi ([parametrlar ro‘yxati]):*
    buyruqlar_bloki

def – funksiyani e’lon qiluvchi kalit so‘z.

funksiya_nomi – funksiya nomi.

parametrlar ro‘yxati – ushbu ro‘yxat bir necha parametrdan iborat bo‘lishi mumkin va ular vergul bilan ajratib yoziladi.

buyruqlar_bloki – funksiya tanasi boshqa operatorlar kabi bitta xat boshi tashlab yozilishi shart.

Funksiya nomi orqali chaqirilganda uning tarkibidagi buyruqlar ketma-ketligi bajariladi. Shundan so‘ng dastur funksiya chaqirilgan satrga qaytadi va shu satrdan keyingi buyruqlarga o‘tadi.

Misol. Xabarni chiqarish.

```
def welcome():
    msg='Xayrli kun! '
    return msg

print(welcome())
```

```
Xayrli kun!
```

salomlashish nomli funksiya e'lon qilindi.
xabar o'zgaruvchisiga qiymat berish.
Funksiyaning vazifasi – xabar o'zgaruvchisi qiymatini qaytarish.
Funksiyani chaqirib, ekranga chiqarish.

Protsedurani e'lon qilish

```
def msg():
    print('Bugun soat 14.00 da ota-
                    onalar majlisi!')
```

```
>>> msg()
Bugun soat 14.00 da ota-onalar majlisi!
```

Pythonda protseduralar deyarli funksiyalardek yoziladi. Farqi shundaki, protseduralar hech qanday qiymatni qaytarmaydi. Quyida protseduraga misol keltirilgan:

Funksiyaga qiymat uzatish

Funksiyaga qayta ishlashi uchun qiymatlar berish mumkin.

Misol. Aylananing radiusi kiritilganda, uning uzunligini topish dasturini tuzing.

```
def circle(r):
    PI=3.14
    len=2*PI*r
    return len
radius = int(input('Aylana radiusi: '))
uz= circle(radius)
print('Aylananing uzunligi: ', uz)
```

```
Aylananing radiusi: 4
Aylananing uzunligi:  25.136
```

aylana nomli funksiya e'lon qilindi, uning qabul qiluvchi qiymati – **r**.
Aylananing uzunligi hisoblandi.
Funksiya aylana uzunligini qaytaradi.
Foydalanuvchi tomonidan kiritilgan radiusni butun qiymatga o'zgartirish.
Aylana nomli funksiya chaqirilmoqda.
Aylana uzunligi chiqarilmoqda.

Misol. *n* faktorialini hisoblash dasturini tuzing.
S=1*2*3*…*n=n!

```
def factor(n):
    res=1
    for i in range(2,n+1):
        res*=i
    return res
n=int(input('n sonini kiriting:'))
print(factor(n))
```

```
n sonini kiriting: 5
120
```

factor nomli funksiya e'lon qilindi.
Ko'paytmaning birinchi qiymati kiritildi.
Sikl 2 dan boshlab n+1 gacha, ya'ni 1 marta takrorlanadi.
res=1*2*…*n
res natijasini qaytaradi.
n ga qiymat beriladi.
n faktorialni hisoblovchi funksiya chaqirilib, ekranga natija chiqariladi.

Rekursiya

Funksiyaning o‘zini o‘zi chaqirishiga *rekursiya* deyiladi va bunday funksiyalar **rekursiv funksiyalar** deb ataladi.

Rekursiv funksiyalar dasturlashning kuchli mexanizmi hisoblanadi, lekin ular har doim ham samarali emas. Chunki aksariyat hollarda xatolarga yo‘l qo‘yadi. Xatolar ichidan eng ko‘p tarqalgani – *cheksiz rekursiya.* Unda funksiyaning chaqiruv zanjiri cheksiz bo‘lib, kompyuter bo‘sh xotirasi tugamaguncha davom etaveradi. Cheksiz rekursiya ro‘y berishining sabablari:

– rekursiyada shartni noto‘g‘ri qo‘llash. Masalan, faktorialni hisoblashda if n==0 ni unutib qo‘ysak, factorial(0) funksiyasi factorial(−1)ni, factorial(−1) funksiyasi esa factorial(−2) va hokazolarni chaqiradi;

– rekursiv funksiyani noto‘g‘ri parametr bilan chaqirish. Masalan, factorial(n) funksiya factorial(n)ni chaqirsa, yana cheksiz zanjir yuzaga keladi.

Shu boisdan, rekursiv funksiyani yaratishda rekursiyani yakunlash sharti hamda rekursiyani qachon, qanday tugatish haqida o‘ylab ko‘rish lozim.

```
def factor(n):
    if n==0:
        return 1
    else:
        res= n*factor(n-1)
        return res
n=int(input('n sonini kiriting: '))
print(factor(n))
```

```
n sonini kiriting: 5
120
```

factor nomli rekursiv funksiya e’lon qilindi.
n==0 da funksiya 1 ni qaytaradi, aks holda ishini davom ettiradi.
Funksiya o‘zi o‘zini chaqirib, res=n*(n-1)*…*3*2*1ni hisoblaydi, qachonki factor(0) bo‘lguncha.
res natijasini qaytaradi.
n ga qiymat beriladi.
n faktorialni hisoblovchi funksiya chaqirilib, ekranga natija chiqariladi.

SAVOL VA TOPSHIRIQLAR

1. *Qism dastur nima?*
2. *Dasturda protsedura va funksiyalar qanday maqsadda qo‘llaniladi?*
3. *Qism dasturning qanday turlari mavjud?*
4. *Qism dastur qanday afzalliklarga ega?*
5. *Protsedura va funksiyaning farqi nimada?*
6. *Qachon funksiyaning o‘rniga protsedurani qo‘llash mumkin?*

UY VAZIFASI

1. *n* va *k* butun musbat sonlar berilgan. *n* va *k* qatnashgan ushbu ifodani hisoblang.

$$\frac{n!}{k!(n-k)!}$$

Funksiyadan foydalaning.

2. *n* natural son berilgan. Kvadrati *n* dan kichik bo‘lgan barcha natural sonlarni chiqaruvchi dastur tuzing.

3. Bir birlik uzunlik ‘-’ ga teng. Berilgan *n* uzunlikdagi ‘-’ belgidan iborat chiziq chizuvchi dastur tuzing. Protseduradan foydalaning.

Kiruvchi ma’lumot	Chiquvchi ma’lumot
n	*n* ta chiziq (‘-’)
5	-----