

Pythonda string metodi haqida oz bo'lsa ham tushunchaga ega bo'ldik. Endi esa pythonda butun sonlarni o'rganamiz(INTEGERS(int)).

INTERGERS -Butun sonlar

Butun sonlar ham o'zgaruvchida saqlash, ularning ustida +,-,*,./ kabi arifmetik amallarni bajarishimiz mumkin:

```
a = 20 #musbat son
```

```
b = -30 #manfiy son
```

```
c = 0
```

```
d = a+b
```

```
print(d)
```

Python operatorlar orasidagi bo'shliqlarni inobatga olmaydi, shuning uchun tenglik belgisi yoki arifmetik operatorlardan oldin va keyin bo'sh joy tashlab yozish ko'dimizni o'qish uchun qulay hisoblanadi.

FLOATS - o'nlik sonlar

Pythonda o'nlik sonlar floating point numbers yoki qisqa qilib floats deyiladi.

```
#Aylana uzunligini hisoblovchi kod
```

```
pi = 3.14159 #o'nlik son(float)
```

```
radius 10 #butun son(integer)
```

```
diametr = 2*radius
```

```
print("Aylana uzunligi ", pi*diametr, " ga teng.")
```

```
#Kvadrat yuzini hisoblovchi kod
```

```
kv_tm = 20 #kvadrat tomoni 20ga teng
```

```
kv_yuzi = kv_tm**2 Kvadrat yuzini hisoblaymiz
```

```
print(kv_yuzi)
```

BUTUN SONDAN O'NLIK SONGA

Avval aytganimizdek ikki butun sonni bo'lish (/) natijasida o'nlik son hosil bo'ladi (natija butun son bo'lsa ham).

```
a = -20
```

```
b = 40
```

```
c = b/a
```

```
print(c) #natija o'nlik son bo'ladi
```

Shuningdek butun son va o'nlik sonlar o'rtasidagi har qanday arifmetik amallarning natijasi ham on'lik son bo'ladi.

```
a = 2 #butun son
```

```
b = 3.0 #o'nlik son
```

```
print(a + b) #5.0
```

```
print(a * b) #8.0
```

```
print(2* (a + b)) #10.0
```

Pythonda uzun sonlar:

Uzun sonlarni kiritishda qulaylik uchun raqamlarni pastki chiziq(_) yordamida ki ritishimiz mumkin. Bunda python pastki chiziqni inobatga olmaydi.

```
aholi = 7_500_000
```

```
print(f'O'zbekiston aholisi {aholi} dan ortiq!')
```

KONSTANTA

Aksar dasturlash tillarida konstatns tushunchasi bor. Konstantalar o'zgarmas bo'ladi (misol uchun (n) ning qiymati konstanta o'zgarmas) Pythonda konstanta tushunchasi yuq. Biroq ko'plab pythonchilar buni yechimini topgan holda o'zgarmaslarni katta harfda yozishga kelishib olishgan(ogohlantirish sifatida).

```
PI = 3.14159 #konstanta
```

Bir nechta o'zgaruvchiga qiymat berish

Birdaniga bir nechta o'zgaruvchiga qiymat berishimiz mumkin bu imkoniyat hozirgi vaqtda bir nechta dasturlash tillarida mavjud shung bilan pythonda ham.

```
x, y, z = 10, -7, 30.2
```

Bunda x ga 10 ,y ga -7, z ga 30.2 ni yuklamoqda

Bu usulni istalgan turdagi o'zgaruvchiga qo'llash mumkin:

```
yosh, ism = 19, "Xusanbek"
```

```
print(f'{ism.title()} {yosh} yoshda ")
```

O'zgaruvchi turini aniqlash

Ko'dimizda o'zgaruvchilar ko'payib ketdi Xatolar qilmaslik uchun ba'zida o'zgaruvchi turini aniqlash talab qilinadi. Bunda bizga type() usuli yordam beradi:

```
ism = "Xusanbek"
yosh = 19
print(type(ism))
print(type(yosh))
```

Keling quyidagi kodni ko'rib chiqaylik. Maqsadimiz ism hamda yosh o'zgaruvchisidagi ma'lumotlarni xabar o'zgaruvchiga yuklab "Javohir 11 yoshda" degan matnni konsolga chiqarish

```
ism = "Javohir"
yosh = 11
xabar = ism + ' ' + yosh + ' ' + 'yoshda '
print(xabar)
Natija!
```

Demak pythonda matn va son turdagi ma'lumotlarni qo'shib bo'lmaydiekan Xosh bunga yechim bormi? Albatta! Pythondagi bir turdagi o'zgaruvchini boshqa turga o'tkazish mumkin. Bu usul **"typecasting"** deyiladi. Buning uchun pythonda mahsus belgilar mavjud ular bilan tanishamiz:

str() - int yoki float turdagi sonlarni matn ko'rinishida qaytaradi.

int() - str yoki float ko'rinishidagi o'zgaruvchilarni butun son ko'rinishida qaytaradi. Bunda matn butun son ko'rinishida berilishi kerak.

float() - str yoki int ko'rinishidagi ma'lumotlarni o'nlik son ko'rinishida qaytaradi.

Demak yuqoridagi kod to'g'ri ishlashi uchun 3- qatorni quyidagicha uzgartiramiz:

```
ism = "Javohir"
yosh = 36
xabar = ism + ' ' + str(yosh) + ' ' + "yoshda"
```

str(), int() va float() funksiyalari berilgan qiymatni boshqa ko'rinishda qaytaradi xolos. Bunda asl o'zgaruvchi turi o'zgarmay qoladi.

input() va sonlar

Avvalgi bo'limda foydalanuvchidan ma'lumot olish uchun input() funksiyasidan foydalanishni o'rgandik Keling shu funksiyadan foydalanib foydalanuvchidan son olishni ko'rib chiqamiz. Quyidagi kod foydalanuvchining foydalanuvchining tug'ilgan yilini soaydi va uning yoshini hisoblab beradi:

#1 qator foydalanuvchidan tug'ilgan yilini soraymiz

```
t_yil = input("Tug'ulgan yilingizni kiriting\t")
```

#2 foydalanuvchi yoshini hisoblaymiz

```
yosh = 2023 - t_yil
```

#3 foydalanuvchi yoshini konsolga chiqaramiz

```
print("Siz " + yosh + "yoshda ekansiz!")
```

Natija!

Kutilgan natija o'rniga xatolik. Xatolikni tarjima qilamiz.