

3. O`ZGARUVCHILAR

O`zgaruvchilar ma`lumotlar qiymatlarini saqlash uchun konteyner vazifasini bajaradi

3.1.1 O`zgaruvchilarni yaratish

Pythonda o`zgaruvchilarni e`lon qilish buyrug`i yo`q. O`zgaruvchi unga qiymat bergan vaqtida yaratiladi

```
x = 5
y = "John"
print(x)
print(y)
```

3.1.2 O`zgaruvchini yaratganizdan so`ng undagi ma`lumotni ya`na o`zgartirishingiz mumkin. O`zgaruvchi oxirgi kiritilgan ma`lumotni o`zida saqlaydi

```
x = 4          # x O`zgaruvchi tipi raqam (int)
x = "Sally"    # x O`zgaruvchi tipi satr (str)
print(x)       # printda oxirgi ma`lumot yani Sally chiqadi
```

```
x = 4          # x O`zgaruvchi tipi raqam (int)
print(x)       #bu yerda printda 4 chiqadi

x = "Sally"    # x O`zgaruvchi tipi satr (str)
print(x)       #bu yerda esa Sally chiqadi. Sabab birinchi printdan keyin
x o`zgaruvchisi Sallyga o`zgartirildi
```

3.1.3. Kasting

Agar siz o'zgaruvchining ma'lumotlar turini belgilamoqchi bo'lsangiz, bu casting yordamida amalga oshirilishi mumkin.

```
x = str(3)     # x o`zgaruvchi ichidagi ma`lumot '3' ko`rinishida bo`ladi
y = int(3)     # y o`zgaruvchi ichidagi ma`lumot 3 ko`rinishida bo`ladi
z = float(3)   # z o`zgaruvchi ichidagi ma`lumot 3.0 ko`rinishida bo`ladi
```

Ma`lumot turlari xaqida batafsil o`rganamiz.

3.1.4. O`zgaruvchi tipini aniqlash. O`zgaruvchi tipi biz uchun juda muxim. $x = \text{str}(3)$ va $y = \text{int}(3)$ o`zgaruvchilari orasida juda katta farq bor.  $x$  o`zgaruvchisida matematik amallarni bajarish mumkin emas. x o`zgaruvchisi 3 raqamini harf deb xisoblaydi va unda matematik amallarni bajarishga ruxsat bermaydi.

O`zgaruvchi tipini `type()` funksiyasi yordamida aniqlash mumkin

```
x = 5
y = "John"
print(type(x)) → <class 'int'>
print(type(y)) → <class 'str'>
```

3.1.5 Bir tirnoq va qo`sh tirnoq

Satr (str) tipidagi o`zgaruvchilarni yaratganimizda va printda biror matn chiqarmoqchi bo`lsak bir yoki qo`sh tirnoqdan foydalanamiz. Ikkala tipda xam foydalansak bo`ladi python xatolik bermaydi.

```
x = "John"
# ikkalasi bir xil
x = 'John'
```

Lekin bir tirnoqdan foydalanishni kamchiligi bor. Agar matnda o` yoki g` yoki tutuq belgisida foydalansak python bu ` belgini bir tirnoq deb xisoblaydi

```
x = `mustaqil o`zbekiston` #x o`zgaruvchisiga faqat `mustaqil o` qismini
saqlaydi. zbekiston`qismiga xatolik beradi.
```

```
File "./prog.py", line 1
    x = 'mustaqil o'zbekiston
          ^
SyntaxError: invalid syntax
```

3.1.6. O`zgaruvchilar harflar katta kichikligiga sezgir

```
a = 4
A = "Sally"
#A va a ikkalasi aloxida o`zgaruvchi xisoblanadi
```

3.2 O`zgaruvchi nomlari

O`zgaruvchilarni qisqa (masalan x yoki y) va o`zgaruvchini tasvirlab(yosh, ism, mening_kasbim) yaratishingiz mumkin:

- ✓ O`zgaruvchi xar doim harf(katta yoki kichik) yoki pastki chiziqcha bilan boshlanish shart
- ✓ O`zgaruvchi raqam bilan boshlanishi mumkin emas
- ✓ O`zgaruvchi nomi faqat alfa-numerik belgilar bilan yaratilishi shart (A-z, 0-9 va _)
- ✓ O`zgaruvchilar harflar katta kichikligiga sezgir(yosh, Yosh, yOsh, YOSH bular barchasi turli xil o`zgaruvchi xisoblanadi)

- ✓ O'zgaruvchilar yaratilayotganda python keyword laridan foydalanish mumkin emas (Python keywords mavzusida keywordlar xaqida ma'lumot berilgan.)

mumkin

```
myvar = "John"  
my_var = "John"  
_my_var = "John"  
myVar = "John"  
MYVAR = "John"  
myvar2 = "John"
```

mumkin emas

```
2myvar = "John"  
my-var = "John"  
my var = "John"
```

3.2.1 Ko'p so'zli o'zgaruvchilar yaratish

Bir nechta so'zli o'zgaruvchilarni nomini o'qish qiyin bo'lishi mumkin.

Ularni o'qilishi mumkin bo'lgan bir nechta usullardan foydalanishingiz mumkin

1. Tuya o'rkachi usuli. Har bir so'z, birinchisidan tashqari, bosh harf bilan boshlanadi:

```
myVariableName = "John"
```

2. Paskal usuli. Har bir so'z bosh harf bilan boshlanadi:

```
MyVariableName = "John"
```

3. Ilon izi usuli. Har bir so'z pastki chiziq belgisi bilan ajratiladi:

```
my_variable_name = "John"
```

3.3.1 O'zgaruvchilarga bir nechta qiymat belgilash

Bir qatorda bir nechta o'zgaruvchiga bir nechta qiymat belgilash

```
x, y, z = "Orange", "Banana", "Cherry"  
print(x)  
print(y)  
print(z)
```

```
Orange  
Banana  
Cherry
```

Eslatma: O'zgaruvchilar soni qiymatlar soniga mos kelishiga ishonch hosil qiling, aks holda siz xatoga yo'l qo'yasiz.

3.3.2 Bir qiymatdan bir nechta o'zgaruvchilarga

Bir qatorda bir nechta o'zgaruvchilarga *bir xil* qiymatni belgilashingiz mumkin :

```
x = y = z = "Orange"
print(x)
print(y)
print(z)
```

```
Orange
Orange
Orange
```

3.3.3 To'plamni ochmoq. Listdagi qiymatlarni tartib bo'yicha aloxida o'zgaruvchilarda saqlash

Agar sizda list yoki tuple da qiymatlar bo'lsa uni o'zgaruvchilarga chiqarib olishingiz mumkin. Bu pythonda unpacking deb ataladi

```
fruits = ["apple", "banana", "cherry"]
x, y, z = fruits
print(x)
print(y)
print(z)
```

```
apple
banana
cherry
```

Unpacking xaqida ko'proq tuple ga bag'ishlangan darsda o'rganasiz

3.4 O'zgaruvchilarni chiqarish (Output Variables)

3.4.1. print() funksiyasi o'zgaruvchilarni chiqarish uchun ishlatiladi

```
x = "Python is awesome"
print(x)
```

```
Python is awesome
```

3.4.2. print() funksiyasi orqali bir nechta o'zgaruvchilarni vergul bilan ajratgan xolatda xam chiqarishingiz mumkin

```
x = "Python"
y = "is"
z = "awesome"
print(x, y, z)
```

```
Python is awesome
```

3.4.3 O'zgaruvchilar orasiga vergul o'rniga + qo'yib chiqarishingiz ham mumkin

```
x = "Python "  
y = "is "  
z = "awesome"  
print(x + y + z)
```

```
Python is awesome
```

!!! Xar bir qiymatdan keyin qo'shtirnoqni yopishdan oldin bitta joy (probel) tashlashni unutmang. Yu`qoridagi misolda is dan keyingi bo'sh joyni ochirib qo'ysak natijani ko`ring

```
x = "Python|" #bo'sh joy bor  
y = "is" #bosh joy yo`q  
z = "awesome"  
print(x + y + z)
```

```
Python isawesome
```

!! is va awesome ni qo'shib yubordi

3.4.4 Raqamli o'zgaruvchilar uchun + matematik ishora sifatida ishlaydi

```
x = 5  
y = 10  
print(x + y)
```

```
15
```

!!! Diqqat bilan o'rganing.

str (satr) tipidagi o'zgaruvchi bilan int (raqam) tipidagi o'zgaruvchini + ishorasi orqali qo'shib chiqarsangiz python xatolik beradi

```
x = 5  
y = "John"  
print(x + y)
```

```
TypeError: unsupported operand type(s) for +: 'int' and 'str'
```

!!! Diqqat bilan o'rganing.

```
x = 5 #python buni raqam deb biladi  
y = "5" #python buni harf deb biladi  
print(x + y)
```

```
TypeError: unsupported operand type(s) for +: 'int' and 'str'
```

Bir nechta o'zgaruvchini birg alikda chop etishning eng yaxshi usuli ularni vergul bilan ajratish

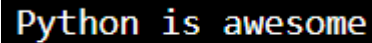
```
x = 5  
y = "John"  
print(x, y)
```

```
5 John
```

4.5 Global o`zgaruvchilar

4.5.1. Shu vaqtgacha o`rgangan o`zgaruvchilarimi global o`zgaruvchilar deyiladi va biz bu o`zgaruvchilarni istalgan joyda funksiya ichida va tashqarisida xatto klaslar ichida xam foydalanishimiz mumkin

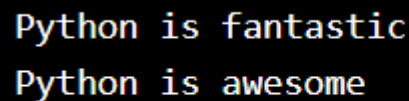
```
x = "awesome"
def myfunc():
    print("Python is " + x)
myfunc()
```



```
Python is awesome
```

4.5.2. Agar siz biror funksiya ichida oldin yaratgan o`zgaruvchingiz bilan bir xil nomdagi o`zgaruvchi yaratmoqchi bo`lsangiz bu o`zgaruvchi local o`zgaruvchi deyiladi va faqat funksiya ichida ishlatishingiz mumkin. Funksiyadan tashqarida yaratilgan global o`zgaruvchi qiymati esa o`zgarmay qoladi

```
x = "awesome"
def myfunc():
    x = "fantastic"
    print("Python is " + x)
myfunc()
print("Python is " + x)
```



```
Python is fantastic
Python is awesome
```

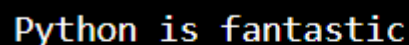
!!! E`tabor bering funksiya tashqaridagi o`zgaruvchi `x = "awesome"` bo`lsa, funksiya ichidagi `x = "fantastic"` ga ten bo`lyapti va o`z qiymatlarini saqlab qolayapti. Lekin funksiya ichidagi `x` ni xech qacho funksiadan tashqarida ishlata olmaymiz

4.5.3. Global kalit so`zi

Odatda, funksiya ichida o`zgaruvchi yaratganingizda, u o`zgaruvchi mahalliy bo`lib, faqat shu funksiya ichida ishlatilishi mumkin.

Funktsiya ichida global o'zgaruvchini yaratish uchun siz `global` kalit so'zdan foydalanishingiz mumkin.

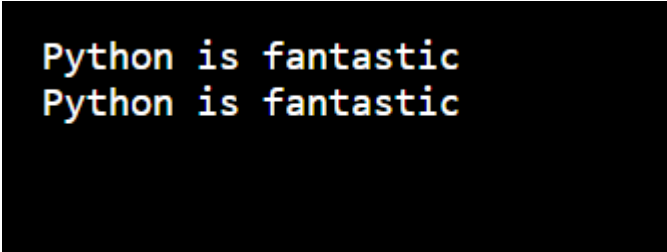
```
def myfunc():
    global x
    x = "fantastic"
myfunc()
print("Python is " + x)
```



```
Python is fantastic
```

4.5.4. Funksiyadan tashqarida yaratilgan o`zgaruvchi qiymatini funksiya ichida o`zgartirish uchun ham **global** kalit so`zidan foydalanasiz.

```
x = "awesome"
def myfunc():
    global x
    x = "fantastic"
    print("Python is " + x)
myfunc()
print("Python is " + x)
```



```
Python is fantastic
Python is fantastic
```

!!! 4.5.2 va 4.5.3 punktdagi misollar bilan solishtirib farqini o`rganing

Funksialar mavzusini dars davomida to`liq o`rganamiz