

# O'ZGARUVCHILAR

## (Variables)

### 1. O'zgaruvchi nima?

O'zgaruvchi — bu dasturlashda **ma'lumotni saqlash uchun ajratilgan joy**. U xuddi qutiga o'xshaydi: qutiga nima solsak, uni keyinchalik olib ishlatishimiz mumkin.

Misol:

```
ism = "Ali"  
yosh = 20
```

- ism — o'zgaruvchi nomi
- "Ali" — qiymati (matn)
- yosh — boshqa o'zgaruvchi
- 20 — qiymati (butun son)

---

### 2. O'zgaruvchilarni nomlash qoidalar

✓ To'g'ri yo'llar:

- Nom raqam bilan boshlanmaydi (2ism noto'g'ri).
- Faqat harf, raqam va \_ (pastki chiziq) ishlatiladi.
- Nom tushunarli bo'lishi kerak (x1, x2 o'rniga yosh, ism).

Misollar:

```
ism = "Dilshod"    # to'g'ri  
yosh_2025 = 25     # to'g'ri
```

```
_raqam = 50      # to'g'ri  
2yosh = 18      # noto'g'ri
```

---

### 3. O'zgaruvchilarni ishlatish

O'zgaruvchilarni turli amallarda ishlatish mumkin.

```
a = 10  
b = 5  
c = a + b  # c = 15  
  
matn = "Salom"  
matn2 = " Dunyo"  
natija = matn + matn2  # "Salom Dunyo"
```

---

### 4. O'zgaruvchilar turini aniqlash

Python'da type() funksiyasi yordamida o'zgaruvchining turini bilish mumkin.

```
x = 15  
y = 3.14  
ism = "Hasan"  
katta = True  
  
print(type(x))  # <class 'int'>  
print(type(y))  # <class 'float'>  
print(type(ism)) # <class 'str'>  
print(type(katta))# <class 'bool'>
```

---

### 5. O'zgaruvchilar ustida amallar

O'zgaruvchilarni turli matematik yoki mantiqiy amallarda ishlatish mumkin.

## Matematik amallar

```
a = 12
b = 4

print(a + b) # 16
print(a - b) # 8
print(a * b) # 48
print(a / b) # 3.0
print(a % b) # 0 (qoldiq)
print(a ** 2) # 144 (daraja)
```

## Matnli amallar

```
ism = "Ali"
familiya = "Karimov"
print(ism + " " + familiya) # Ali Karimov

salom = "Salom!\n" * 3
print(salom) # matn 3 marta chiqariladi
```

## Mantiqiy amallar

```
x = 7
y = 5

print(x > y) # True
print(x < y) # False
print(x == y) # False
print(x != y) # True
```

---

## 6. Dasturiy misol: foydalanuvchidan qiymat olish

```
# Foydalanuvchidan ism va yosh olish
ism = input("Ismingizni kiriting: ")
yosh = int(input("Yoshingizni kiriting: "))
```

```
print("Salom,", ism)
print("Siz", yosh, "yoshdasiz.")
print("Keyingi yilingizda", yosh + 1, "yoshda bo'lasiz.")
```

## 7. Xulosa

- O'zgaruvchi — ma'lumotni saqlash joyi.
- Har bir o'zgaruvchining ma'lumot turi bor (int, float, str, bool).
- Ular ustida matematik, matnli va mantiqiy amallar bajarish mumkin.
- Foydalanuvchi bilan muloqot qilishda input() orqali qiymat olinadi.

# Python'dagi Data Types (Ma'lumot Turlari)

Python'da ma'lumotlar turli xil bo'ladi. Har bir qiymat ma'lum bir **data type** ga tegishli.

## 1. Butun sonlar (int)

- **Ta'rif:** Butun sonlarni saqlash uchun ishlatiladi.
- Musbat, manfiy yoki 0 bo'lishi mumkin.
- Decimal (10lik), Binary (2lik), Octal (8lik), Hexadecimal (16lik) ko'rinishda yozilishi mumkin.

```
a = 10    # decimal
b = -7    # manfiy son
c = 0     # nol
```

```
print(type(a)) # <class 'int'>
```

---

## 2. Kasrli sonlar (float)

- **Ta’rifi:** O‘nli kasrlarni saqlaydi.
- Haqiqiy sonlar hisoblash uchun ishlatiladi.
- . nuqta bilan yoziladi.

```
pi = 3.14  
son = -2.5  
nol = 0.0  
  
print(type(pi)) # <class 'float'>
```

---

## 3. Matn (str)

- **Ta’rifi:** Belgilar ketma-ketligi (text).
- Qo‘sh tirnoq " " yoki bitta ' ' ichiga yoziladi.
- Ko‘p qatordan iborat bo‘lishi uchun """ """ ishlatiladi.

```
ism = "Ali"  
matn = 'Salom, Python!'  
kop_qator = """Bu matn  
ko'p qatordan  
iborat!"""  
  
print(type(ism)) # <class 'str'>
```

### Matnlarda amallar:

```
ism = "Ali"  
familiya = "Karimov"  
print(ism + " " + familiya) # Birlashtirish  
print(ism * 3)             # Uch marta takrorlash  
print(len(ism))            # Uzunligini topish
```

---

#### 4. Mantiqiy qiymatlar (bool)

- **Ta’rifi:** Faqat ikki qiymatga ega: True (rost) yoki False (yolgʻon).
- Asosan shartlarda ishlatiladi.

```
katta = True
kichik = False

x = 7
y = 3
print(x > y) # True
print(x < y) # False
```

---

#### 5. Roʻyxatlar (list)

- **Ta’rifi:** Bir nechta elementlarni ketma-ket saqlash uchun ishlatiladi.
- Elementlar turli xil data type boʻlishi mumkin.
- [ ] ichida yoziladi.

```
mevalar = ["olma", "banan", "uzum", 5, 3.14]
print(type(mevalar)) # <class 'list'>
print(mevalar[0])    # olma
mevalar.append("nok") # oxiriga qoʻshish
print(mevalar)
```

---

#### 6. Oʻzgarmas roʻyxat (tuple)

- **Ta’rifi:** list ga oʻxshaydi, lekin oʻzgarmaydi.
- ( ) ichida yoziladi.

```
ranglar = ("qizil", "yashil", "ko'k")
print(type(ranglar)) # <class 'tuple'>
print(ranglar[1])    # yashil
```

---

#### 7. Toʻplam (set)

- **Ta’rifi:** Unikal (takrorlanmaydigan) elementlar to‘plami.
- { } ichida yoziladi.

```
sonlar = {1, 2, 3, 3, 4, 5}
print(sonlar)    # {1, 2, 3, 4, 5} (takror yo‘q)
print(type(sonlar)) # <class 'set'>
```

## 8. Lug‘at (dict)

- **Ta’rifi:** Kalit-qiymat (key-value) ko‘rinishidagi ma’lumotlar to‘plami.
- {kalit: qiymat} ko‘rinishida yoziladi.

```
talaba = {
    "ism": "Ali",
    "yosh": 20,
    "kurs": 2
}
print(talaba["ism"]) # Ali
print(type(talaba)) # <class 'dict'>
```

## 9. NoneType (None)

- **Ta’rifi:** Qiymat yo‘qligini bildiradi.
- Ko‘pincha dasturda “hali qiymat belgilanmagan” holat uchun ishlatiladi.

```
x = None
print(type(x)) # <class 'NoneType'>
```

## (Asosiy Data Types jadvali)

Data Type	Tavsifi	Misol
int	Butun son	5, -10, 0
float	Kasrli son	3.14, -2.5
str	Matn	"Salom", 'Ali'

Data Type	Tavsifi	Misol
<b>bool</b>	Mantiqiy qiymat	True, False
<b>list</b>	O'zgaruvchi ro'yxat	[1, 2, "Ali"]
<b>tuple</b>	O'zgarmas ro'yxat	(1, 2, 3)
<b>set</b>	Takrorlanmaydigan elementlar	{1, 2, 3}
<b>dict</b>	Kalit-qiymat to'plami	{"ism": "Ali", "yosh": 20}
<b>None</b>	Qiymat yo'q	None

The Elements