**Python Data Types**

Data nədir ?.

Kompüter mühitində məlumatların proqramlar tərəfindən işlənəbilməsini təmin etmək üçün informasiyanın tərtib və formalaşdırılan formuna data deyilir.

Datanın fərqli formlarına rəqəmləri , mətnləri və.s örnək göstərə bilərik.Məsələn marketə getmədən öncə hazırladığımız alacaqlar siyahısınıda buna misal çəkə bilərik. Bir çox proqramlaşdırma dillərində olduğu kimi Python da datanın fərqli növlərini istifadə etməyimizə icazə verir. Digər dillərlə müqayisədə Python data növlərindən istifadədə daha rahatdır . Çünki Python bizə data növlərini bir birinə rahatlıqla çevirməyə, yenidən data əlavə etməyə icazə verir .Əlbəttə bu rahatlığın özünə xas avantajları və dezavantajları var. Məsələn ən önəmli avantajı istifadə rahatlığıdır. Yəni bir data tipini digərinə çevirmək daha rahatdır . Dezavantajı isə odur ki xətaya daha əlverişlidir. Proqramçının gözündən qaçan data tipi ilə bağlı xətaları tapmaq Python kimi nəzərdən keçirilə bilən dillərdə daha çətindir .

Datanın nə olduğu haqqında məlumat sahibi olduqdan sonra , indi isə data tipləri ilə daha yaxından tanış olaq.Bu məqalədə data tiplərinin bir qismi haqqında danışacam.Digər növlər haqqında isə sonrakı məqalələrimdə ətraflı bəhs edəcəyəm.

*Text Type ( mətn növü ): str*

*Numeric Types (ədədi növlər ): int , float , complex*

*Sequence Types (ardıcıllıq növləri) : list , tuple , range*

*Mapping Types ( xəritəçəkmə növləri ) : dict*

*Set Types(set növləri) : set , frozenset*

*Boolean Type(məntiqi növ) : bool*

*Binary Types(ikili növlər) : bytes , bytearray , memoryview*

## Niyə dataları müxtəlif kateqoriyalara ayırmağa ehtiyac duyuruq ?.

Bir proqramda fərqli məlumat növlərini emal etməyimiz lazım ola bilir. Məsələn tam ədədlər , kəsr ədələr , mətnlər (string ), məntiqi (boolean ) dəyərlər ( doğru =true , yanlış=false ) ilk ağlımıza gələn data növləridir. Bu data növlərinin böyüklüyü(yəni yaddaşda tutduğu yer üçün ayrılan bit sayısı) və üzərlərində aparılacaq əməliyyatlar bir birindən fərqlidir. Belə də deyə bilərik ki biz rəqəmlərlə riyazi hesablamalar apara bilərik ancaq bu əməliyyatı mətinlərlə edə bilmərik.Buradan belə nəticə çıxara bilərik ki , Python və digər proqramlaşdırma dilləri dataları müxtəlif növlərə ayırır. Yəni dəyişən təyin edilərkən dəyişənin hansı tip dataları saxlayacağı göstərilir.Beləliklə yaddaş bu dəyişən üçün kifayət qədər yer ayırır və bu məlumat növü üçün müvafiq əməliyyatların aparılmasına imkan verir.

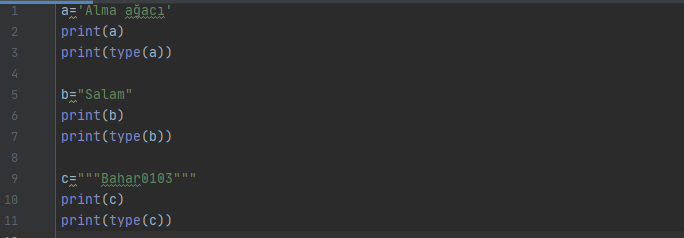
# Text Type ( string )

String data növü unicode simvollarından meydana gələn simvollar qrupudur.Pythonda string datalar tək dırnaq (‘-’), cüt dırnaq(“-’’) və üçlü dırnaq (‘“ -’’’) istifadə edilərək yazılır. İndi isə nə demək istədiyimizi kodlarla daha ətraflı başa düşək.

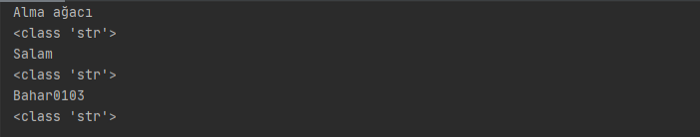
## ( Dil) : Python

## (Data tipi ) : str()

## (Örnək) : ‘Hello world’ ,“Hello world ! ”, ‘“ Hello world !’’’



Onu da qeyd etməliyəm ki string data növlərində sadəcə hərflərdən istifadə edilmir . Yəni dırnaq işarəsi içərisində biz rəqəmlərdəndə istifadə edə bilərik. Beləliklə rəqəmlərdən dırnaq içərisində istifadə edildiyinə görə data növümüz int deyildə string qəbul edilir.



İndi yəqin ki çoxunuzun ağlında belə bir sual yarana bilər. Biz dəyişən təyin edərkən verəcəyimiz datanın tipini demədik , bəs necə oldu ki **print (type(a))**yazdığımızda datamızın tipini ekranda çap etdirə bildik.Bunun səbəbi isə Pythonın**interpreter**dil olmasıdı. Yəni biz datamızın növünü daxil etməsəkdə o özü datanın növünü təyin edə bilir.

## Numeric Types ( int , float , complex )

# Numeric Types(İnteger )

İnteger istifadəsi int() dir . Riyazi tam ədələri ifadə etmək üçün istifadə edilir . Biz mənfi , müsbət , tam olan ( yəni kəsr olmayan ) hər bir ədədi ifadə edə bilərik. İndi isə kodlarla integer data tipinin nə olduğunu daha yaxından öyrənəlim.



İnteger data tipində yazdığımız ədədin uzunluğu müxtəlif ölçülərdə ola bilər. Əsas önəmli olan odur ki, ondalıqlı ədəd olmasın.

Output:



# Numeric Types (Float)

Float data tipindən kəsr ədələri(yəni tam ədəd olmayan ) ifadə etmək üçün istifadə olunur . Float data tipinin iteger data tipindən fərqi də budur .

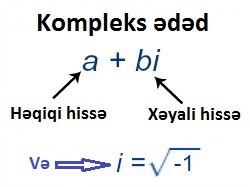


Output:



# Numeric Types (Complex)

Compex data tipi komplex ədədlərin ifadə edilməsində istifadə olunur. İndi bir az komplex ədədlərin nə olduğu ilə tanış olaq. z=a+bi şəklində ifadə olunan ədələrə deyilir .



Əsasən xəyali hissələr i ilə göstərilir. Ancaq bu Pythonda belə deyil yəni i yerinə j istifadə edilir . 2+4i=2+4j (Pythonda).İndi isə kodlarla ifadə edək.





# Boolean Types (Bool)

4. Bool (Boolean) - yazilisi String tipine, gorunusuyse integer tipine benzeyir. Ancaq ikisindende ferqlidir. String tipinde deyisenin icerisine gedecek deyer herf, soz ve ya cumle tek yaxud cut dirnaq icerisinde olmaga mecburdur. Ancaq Boolean tipinde dirnaq icerisinde yazilmir. Buda bize Integer tipini xatirladir. Ancaq Integer tipindeyse, deyisenin icerisine gedecek deyer reqem tipindedir. Boolean, string tipine benzeyen yalniz 2-deyer ala biler: True - Dogru, False - Yalnis.

Meselen:

man = True

woman = False

# List Types (List)

List özündə fərqli tiplərdə datalar saxlayan data strukturdur. Hər bir datanın index dəyəri var. listdən data silmək əlavə etmək və s mümkündür. Boş bir list iki yolla yaratmaq olar.

a=[] #boş list

a=list() #boş list

***Listlərin digərlərində fərqi :***

* *Mutable (dəyişdirilə bilən)’dır.*
* *Sıralanır.*
* *Elementlər****Square brackets (kvadrat mötərizələr) “[]”****içərisində yazılır.*

***List-ləri necə yaradacayıq?***

Python-da, list elementləri **kvadrat mötərizələr**(**square brackets**) içərisində yazılır və **vergül**lər ilə ayrılır. Listlərin içərisinə istənilən qədər və istənilən məlumat növündə(integer, float, string və s.) element yerləşdirmək mümkündür:

# boş list  
 list1 = []  
  
# tam ədədlərdən ibarət list  
 list1 = [1, 2, 3]  
  
# müxtəlif məlumat növlərindən (data types) ibarət list  
 list1 = [1, "Salam", 3.4]

Həmçinin biz list-lərin içərisinə də list **(nested list)** yerləşdirə bilərik:

# iç-içə list (nested list)  
nested\_list = [["medium"], [8, 4, 6], ['a', 'b', 'c']]

**Qaynaq:**

<https://medium.com/pragmatech/python-data-types-1-ci-hiss%C9%99-ec44f2ab1129>

# Tuple Types (Tuple)

# List data tipinden ferqli olaraq, moterizeler icerisinde yazilir ve sonradan dəyişdirilə bilmir, tərkibində müxtəlif tiplər ola bilər.

(10,"Python", **True**)

# Dictionary Types (Dict)

# Açar və dəyər cütlüyündən ibarət qrupdur. Burada Key ve Value istirak edir.

{

'bir':1,

'on':10

}

# Bytes Types (Byte)

# Bitlər ardıcıllığı:

b'ASCII bit'

# Set Types (Set)

# Sıralanmamış çoxluq, tərkibində duplikat ola bilməz

set([10,"Python", **True**])