**Neden Python?**

**1)** Dünya çapında büyük kuruluşlar **(Google, YouTube ve Yahoo! gibi)** bünyelerinde her zaman Python

programcılarına ihtiyaç duyuyor.

**2)** Python programlama dili ile programları kendiniz yazabilirsiniz.

**3)** Python’ı kullanarak **masaüstü, oyun, mobil, web ve ağ** gibi programlama alanlarında çalışmalar yapabilirsiniz.

**4)** Python kodları **sade,basit ve hızlıdır.** Bu nedenle program yazmak başka dillere kıyasla çok kolaydır. Örneğin

Python, C ve C++ gibi dillerin aksine derlenmeye ihtiyaç duymaz.

**5)** Python farklı işletim sistemleri ve platformları üzerinde çalışabilir. **Linux, Windows, Mac OS X, MS-DOS, iOS ve** **Android** gibi belki adını dahi duymadığınız pek çok ortamda Python uygulamaları geliştirebilirsiniz.

**6)** Herhangi bir ortamda yazdığınız bir Python programı, üzerinde hiçbir değişiklik yapılmadan veya ufak değişikliklerle başka ortamlarda çalışabilir

**Not:** Python programlama dilini öğrenmeye yeni başlayan Windows kullanıcıları için en uygun metin düzenleyici **IDLE aslında Python’ın etkileşimli kabuğudur. Dolayısıyla asıl kodları buraya yazmayacağız.** Python programlama diline yeni başlayanların en sık yaptığı hatalardan biri de, kaydetmek istedikleri kodları bu ekrana yazmaya çalışmalarıdır. Unutmayın, bu ekrana yazdığınız kodlar kapattığınızda kaybolur. IDLE’ı açtığınızda sol üst köşede **File [Dosya]** menüsüne tıklayın ve menü içindeki **New Window [Yeni Pencere]** düğmesine basın. Beyaz bir ekranla karşılaşacaksınız. İşte Python kodlarımızı bu ekrana yazacağız. Dikkat ettiyseniz artık >>> sembolü yok.

**Not:** >>> işareti Python’da komut yazmaya hazır olunduğu anlamına gelir.

**Not:** Python’a yeni başlayanların en sık yaptığı hatalardan biri >>> işareti ile komut arasında boşluk bırakmalarıdır.

Eğer bırakırsanız kod hata verecektir.

***!!!NOT:*** # işareti Python da **açıklama (yorum) satırı olarak kullanılır**. Program çalışırken # işaretinin yanında yazılan ifadeler programda gözükmez, sadece kullanıcıya bilgi vermek için kullanılır.!!!

**Python’ da Dört İşlem**

|  |  |  |
| --- | --- | --- |
| + | Toplama | 5 + 3 |
| - | Çıkarma | 5 - 3 |
| \* | Çarpma | 5 \* 3 |
| / | Bölme | 5 / 3 |
| \*\* | Üs alma | 5 \*\* 2 = 25 |
| // | Bölümün tamsayı kısmını verir. | 5//2=2 |
| % | Mod alma kalanı verir | 15 % 2=1 |

VERİ TÜRLERİ:

string: karakterler (harf, özel karakterker, rakamlar) **string ifadeler tırnak içinde yazılır!!**

>>>“Merhaba Dünya”

‘Merhaba Dünya’

Not: string fonksiyonunda çift tırnak kullanmak **şart değildir. Tek tırnak veya üç tırnak da kullanabiliriz.** Üç tırnak için, üç tek tırnak veya üç çift tırnak karakterleri kullanılabilir. Örnek;

>>>‘Merhaba Dünya’ tek tırnak kullanımı

‘Merhaba Dünya’

>>>“””Merhaba Dünya””” üç çift tırnak kullanımı

‘Merhaba Dünya’

Not: Python’da üç tırnak kullanmamızın sebebi alt satıra geçebilmektir. Ör:

print('''Bilgisayar

Bilimi Dersi''') Bilgisayar Bilimi Dersi

 Peki, neden bazen tek tırnak veya çift tırnak kullanmamız gerekiyor? Örnek olarak şöyle bir çıktı elde etmek isteyelim.

İstanbul'un havası

Yukarıdaki ifadede bir kesme işareti var. İçinde kesme işareti olan bir string’i tek tırnak içine alamayız. Bu yüzden diğer bir alternatif olan çift tırnağı ya da üç tırnağı kullanmalıyız.

>>>print("İstanbul'un havası") İstanbul'un havası

>>>print("””İstanbul'un havası””") İstanbul'un havası

integer: Tam sayılar. Pozitif, negatif ya da sıfır değeri alabilir. Kesirli değer içermez.

Ör: 10, 0, -10

float: Reel Sayılar. Kayan noktalı sayılar da denir. Ancak burada virgül yerine nokta kullanmamız gerekir. Ör: 12.6, 12.0, -12.0, 0.0

boolean: Herhangi bir ifadenin doğruluğunu veya yanlışlığını sorgular. **True, False**

FONKSİYONLAR

print(): Ekrana yazdırma fonksiyonudur. Örnekler;

print("Merhaba Dünya") stringleri tırnak içerisine almamız gerekir. Burada çift tırnak kullanıldı

Merhaba Dünya

print(Merhaba Dünya) tırnak işareti olmadığı için hata verir hata

print(“Merhaba Dünya) tırnak işaretini kapatmadığım için hata verir

print(‘Merhaba Dünya’)  stringlerde tek tırnak kullanabiliriz. Merhaba Dünya

print(‘’’Merhaba Dünya’’’)  stringlerde üç adet tek tırnak kullanabiliriz. Merhaba Dünya

ÖRNEKLER

print('Fırat' , 'Dicle') Çıktı: Fırat Dicle # , iki ayrı stringi birleştirirken araya bir boşluk koyar!

print ("bilgi"+"sayar") Çıktı: bilgisayar # + iki ayrı stringi boşluk koymadan birleştirir!

 print(3\* ‘w’) Çıktı: www # \* **stringi** sayı adeti kadar yan yana yazdırır! print(2\*’15’) Çıktı: 1515 #\* **stringi** sayı adeti kadar yan yana yazdırır! print(2\*15) Çıktı: 30 # \* **integer ifadelerde** çarpma işlemi yapar!

**Değişkenler ve Değerinin Yazdırılması**

Değişkenler programlama aşamasında hafızada bilgi tutmaya yarayan araçlardır. Değişkenlere bir isim verilerek gerektiğinde o isimlerle çağırılır. Değişkenlere atama yaparken = işareti kullanılır. Kullanım şekli aşağıdaki gibidir.

a=10 #(bu örnekte a isimli değişkene 10 değeri verilmiştir.)

print (a) #(değişkenin değerini yazdırmak istediğimizde tırnak kullanılmaz. Bu iki satır kod çalıştığında ekranda 10 görüntülenir.

Print(“a”) # çıktı a olarak gözükür. Çünkü a burada değişken olmaktan çıktı. a çift tırnak içinde yazıldığı için çıktısı a olur , print (a) yazsaydı çıktı 10 olacaktı ikisinin farkına dikkat edin!!!

Değişkenlere isim verirken dikkat etmemiz gereken kurallar:

1. Değişken isimleri rakam ile başlamaz

2. Değişken isimleri iki kelimeden oluşuyorsa arada boşluk kullanılmaz.(\_ kullanılır)

3. Değişken isimleri Türkçe karakterden (ç,ş,ı,ü,ö,ğ)oluşmaz.

4. Değişken isimleri verilirken programlamada kullanılan komutlar kullanılmaz. (for, if, elif, len,print, input…)

5. Değişken isimleri verilirken özel karakterler kullanılmaz. (\*,/, -,+, #,%,&,(,=,?,$,[,{ gibi…).

6. Değişkene içerdiği değer ile tutarlı isimler vermeliyiz.

7. Değişken isimlendirmede genellikle küçük harfle başlanır ve ikinci bir kelime yazılacaksa ilk kelimenin hemen ardından büyük harfle devam edilebilir. Örnek: tcKimlikNo

 *Değ işken isim lerin e ö rn ekler:*

sayi1, okulNo, soru\_2, tc\_kimlik, taksitSay, adSoyad

NOT: Değişlenlere atama yaparken = işaretini kullanırız. sayi1=15, ad='aslı' gibi

**Değişken Türleri**

Değişkenler tuttukları verinin içeriğine göre türlere ayrılırlar. Aşağıda örnekleri verilmiştir.

a=10 #(a değişkeni tam sayı bir değer tuttuğu için **integer** türünde bir değişkendir.)

b=3.14 #(b değişkeni ondalıklı sayı bir değer tuttuğu için **float** türünde bir değişkendir.)

c=”KAL-1977” #(c değişkeninin değeri tırnak içine yazıldığı için **string** türünde bir değişkendir.)

d="Ali" #(d değişkeninin değeri tırnak içine yazıldığı için ve karakterden oluştuğu için **string** türünde bir değişkendir