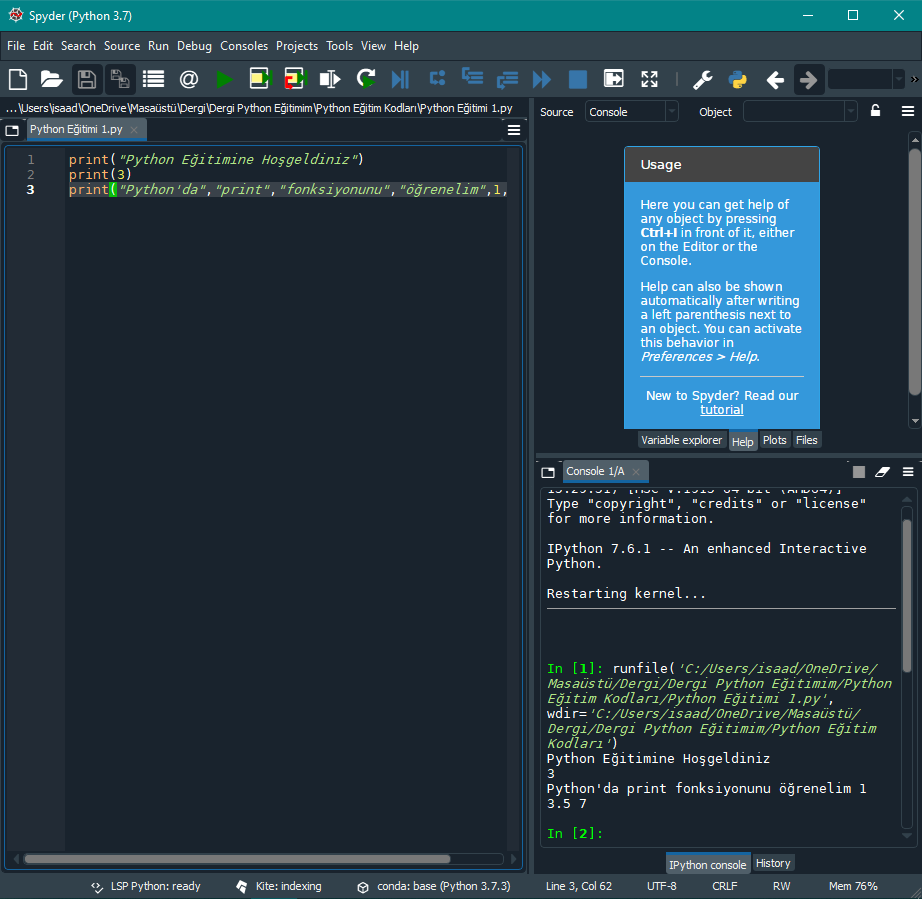
**Print Fonksiyonu**

İlk sayımızda giriş yaptığımız Python eğitimimizde artık python kodlamaya başlayabiliriz. Python dilinin ilk konusu ise print fonksiyonu. Print fonksiyonu içine aldığı değerleri ekrana yazdırmaya yarayan bir fonksiyondur. İçine eğer herhangi bir yazı yazılacaksa bu yazı tırnak (“ ”) içine alınarak yazılır.

* print(“Python Eğitimimize Hoşgeldiniz”)

Şeklinde kodunuzu yazdıktan sonra üst kısımdaki yeşil başlatma tuşuna veya F5 tuşuna basarak kodu çalıştırabilirsiniz. Kodunuzu çalıştırdıktan sonra ise sağ alt kısımdaki konsol kısmından kodunuzun çıktısını görebilirsiniz.

Bunun yanında print fonksiyonu ile sayı bastırmak isterseniz;

* print(3)

şeklinde sayı değerlerini tırnak işareti kullanmadan da bastırabiliriz.

Ayrıca eğer print fonksiyonu ile birden fazla değer bastırmak isterseniz değerlerin aralarına virgül koyarak bu değerleri bastırabilirsiniz. Örneğin;

* print("Python’da”,”print”,”fonksiyonunu”,”öğrenelim”,1,3.5,7)

şeklindeki bir kod ile istediğimiz kadar değeri aynı anda ve aralarına birer boşluk bırakarak bastırabiliriz.

**Matematik İşlemleri**

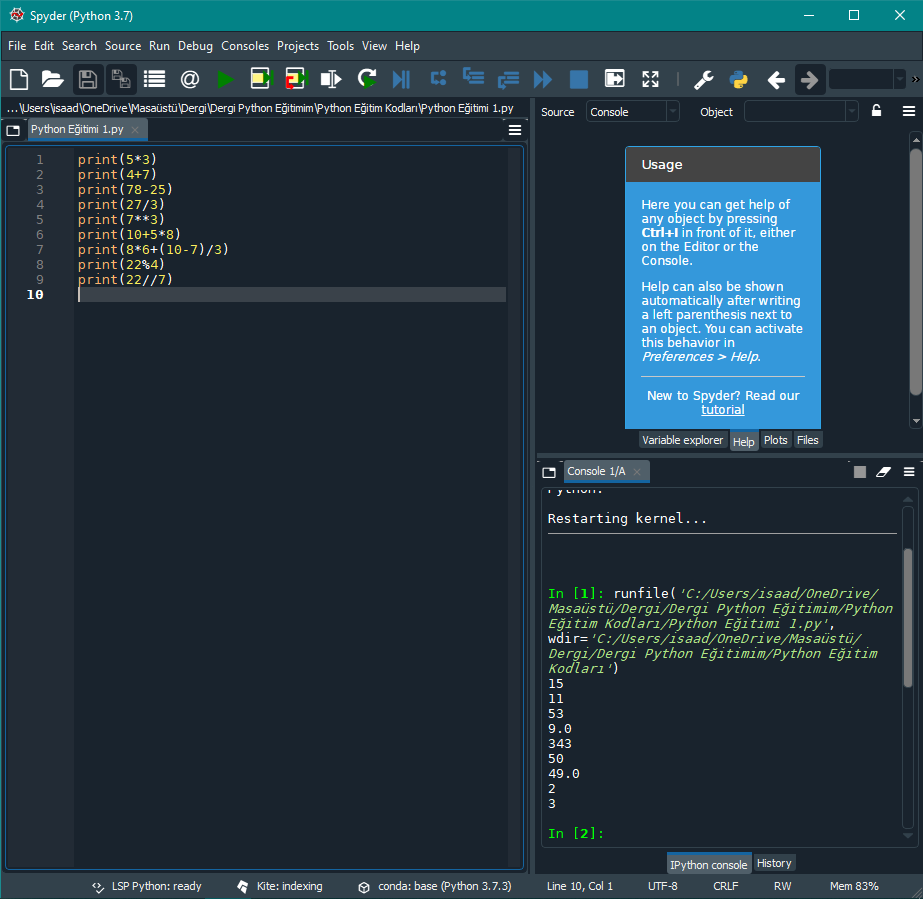
Python kodlama dili ve neredeyse diğer kodlama dillerinin tamamında tam sayı olan değerlere “integer” değerler denir. Ondalık sayılı değerlere ise “float” değerler denir. Bu değerler ile normal matematiktekine benzer olarak matematiksel işlemler yapılabilmektedir.

Matematiksel işlemlerden toplama işlemi Python’da “+” işareti ile, çıkarma işlemi “-“ işareti ile, çarpma işlemi “\*” işareti ile, bölme işlemi ise “/” işareti ile ifade edilmektedir. Bunların yanında üs alma işlemi de “\*\*” işareti ile yapılmaktadır. Örneğin;

* print(5\*3)
* print(4+7)
* print(78-25)
* print(27/3)
* print(7\*\*3)

şeklindeki işlemleri print fonksiyonu ile ekrana yazdırarak sonuçlarına ulaşabiliriz.

Ayrıca matematikte bulunan işlem önceliği Python’da da bulunmaktadır. Bu işlem önceliği matematikteki gibi parantez ile de sağlanabilmektedir. Örnek olarak;

* print(10+5\*8)

işleminin sonucu çarpma önce yapılacağı için ekrana 50 olarak gösterilir.

* print(8\*6+(10-7)/3)

işleminin sonucu da parantez içi önce yapılıp daha sonra da çarpma ve bölme işlemi yapılıp en son ise de toplama işlemi yapılacağından 49 olarak ekrana yazılır.

Bu ifadelerin yanında matematikte olmayan bazı semboller Python’da bulunmaktadır. “%” işareti bölme işleminde kalanı vermektedir. Bunun yanında “ // “ işareti bölme işleminde sonuç kısmının virgülden sonraki kısmını almaz sadece virgülden önceki tam kısmını verir. Bunlara örnek vermek istersek;

* print(22%4)

kodunu yazarak çalıştırırsak sonucu ekranda 2 olarak görürüz.

* print(22//7)

kodunu yazarak çalıştırırsak da sonucu ekranda 3 olarak görürüz. Eğer sadece “/” işaretini kullanarak bölme işlemini yapsaydık 3.142857142857143 değerini ekranda görürdük ki bu da pi sayısına denk gelmektedir.

**Stringler**

Stringler aslında Python dilindeki yazılardır. Stringler Python’da “” veya ‘’ içinde tanımlanır. Örnek vermek istersek;

* print("FLOZOF dergisi”)

veya

* print(‘FLOZOF dergisi’)

şeklindeki kodlar sayesinde istediğimiz string değerini ekrana yazdırabiliriz.

Eğer yazdırmak istediğimiz string şeklindeki cümle içerisinde kesme işareti(‘) kullanacak olursak ve string değerini tanımlamak için de ‘’ işaretini kullanmak istersek kullandığımız kesme işaretini Python bitiş tırnağı olarak algılamakta ve hata vermektedir.Bu durumun önüne geçmek için ya string tanımlaması yapmak için “” işaretini kullanmalı ya da Python’da “Kaçış Dizileri” kullanılmaktadır. Kaçış dizilerini kullanmak için stringte kesme işaretinden önce \ işaretini kullanmamız gerekmektedir. Bu sayede kullandığımız kesme işareti bitiş tırnağı olarak algılanmamaktadır. Bu konumuzu da örnekle açıklamak istersek;

* print("FLOZOF’ta Python öğreniyoruz”)

şeklinde veya

* print(‘FLOZOF\’ta Python öğreniyoruz’)

şeklindeki bir kullanım ile bu sorunu çözebiliriz.

Stringlerde toplama ve çarpma yapmak da Python dilinde mümkündür.Örneğin;

* print("Yazılım” + “öğreniyoruz”)
* print(“FLOZOF dergisi”\*3)

şeklindeki kullanımlar ile stringlerde matematiksel işlemler yapılabilmektedir. Toplama işlemi ve çarpma işleminde eğer stringin içinde boşluk bırakmazsak kendiliğinden boşluk koymamaktadır.Yani ilk kodun çıktısı “Yazılımöğreniyoruz”, ikinci kodun çıktısı ise “FLOZOF dergisiFLOZOF dergisiFLOZOF dergisi” şeklinde olmaktadır.

Stringler içinden herhangi bir değere ulaşmak da mümkündür. Python’da stringlerde her bir değer sıfırdan başlayarak indexlenmektedir. Bu indexlere erişmek için de kesme işareti kullanılmaktadır. Örnek olarak;

* print("Python kodluyoruz”[0])

şeklinde bir kod yazarsak ekranda sonuç olarak “P” değerini görürüz. Bu indexleme negatif sayılarla sondan giderek de yapılabilmektedir. Örneğin;

* print(“Python öğreniyoruz”[-2])

şeklindeki bir kodumuzun ekrandaki çıktısı “u” olacaktır.

Bunların yanında indexleme işlemini belli bir aralık için yapmak istersek;

* print("FLOZOF dergisi”[2:5])

şeklindeki bir kodun çıktısı “OZO” olacaktır. Yani başta yazan index gösterilirken sondaki index gösterilmemektedir. Eğer baştan başlayarak belli bir yere kadar veya belli bir yerden sona kadar olan kısmı almak istersek :’nın başını veya sonunu boş bırakmamız gerekir.

* print(“Kodlamayı seviyoruz”[3:])

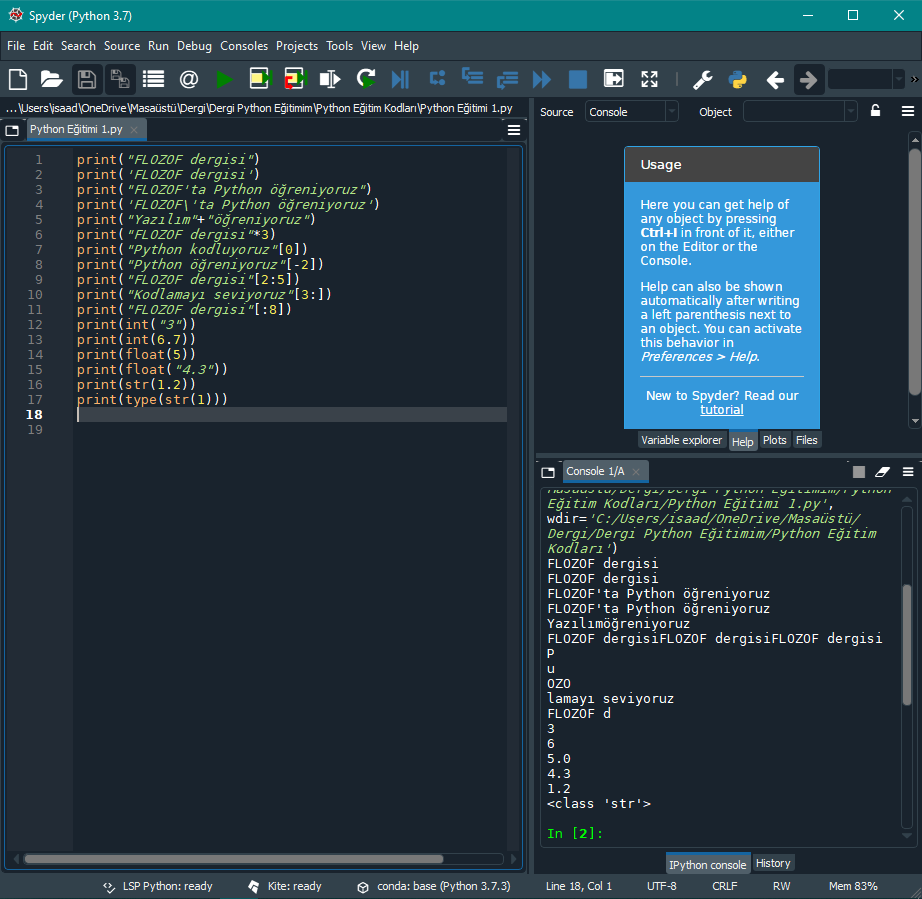
şeklindeki kodun çıktısı “lamayı seviyoruz” şeklinde olmaktadır.

* Print("FLOZOF dergisi”[:8])

şeklindeki kodun çıktısı ise “FLOZOF d” olacaktır.

ÖNEMLİ NOT!!: Stringlerde indexler belirlenirken harflerin yanında boşluklar da indexlere dahil edilmektedir.

Bu sayımızın son konusu olarak ise integer,float ve string değerleri birbirlerine nasıl dönüştürebileceğimizi göreceğiz. Bunun için bazı fonksiyonlar kullanmamız gerekmektedir. Bir string veya float değeri integer değere dönüştürmek için int() fonksiyonunu kullanırız.Bu fonksiyonda parantez içine dönüştürmek istediğimiz değeri yazarak çalıştırdığımızda içerideki değer integer değere dönüşmektedir. Tabiki string değeri integer değere dönüştürmek için “” içindeki değerimizin bir sayı olması gerekir. Aynı şekilde bir string veya integer değeri float değere dönüştürmek için de float() fonksiyonu kullanılmaktadır. Float veya integer bir değeri string bir değer haline getirmek istersek de str() fonksiyonunu aynı şekilde kullanabiliriz. Bu fonksiyonlar ile değerlerin tiplerinin değiştiğini görebilmek için değerlerin tiplerini gösteren type() fonksiyonu kullanılabilir. Bunlara örnek vermek gerekirse de;

* print(int(“3”))
* print(int(6.7))
* print(float(5))
* print(float(“4.3”))
* print(str(1.2))

şeklinde bu fonksiyonların da ekrandaki çıktılarını görerek örneklendirebiliriz.

* print(type(str(1)))

kodunun çıktısı <class 'str'> şeklinde olmaktadır.Yani integer bir değer olan 1 string bir değere yani “1” haline dönmüştür.

Bu sayımızdaki Python eğitimi bölümümüzün de burada sonuna geldik. Burada öğrendiniz şeyler üzerinde kendiniz örnekler oluşturarak pratik yapmayı da unutmayın. Bir sonraki sayımızdaki eğitimiz için takipte kalın.