**Python 3: Fonksiyonlar, Karar Kontrol Yapıları, Döngüler**

* **Fonksiyonlar:**

Print fonksiyonunu kullanırken sep = “ –“ yazarsak çıktılar arasına – işareti koyarak çıktı verir.

**Matematiksel İşlemler:**

\* = Çarpma

/ = Bölme

-- = Çıkartma

+ = Toplama

\*\* = Üst kuvvetini almak

**Fonksiyon Yazmak:**

Def ile fonksiyon tanımlamaya başlarız sonra fonksiyona ad veririz (neye bağlı olduğunu yaz) :

Bir alt satıra geçilir ve boşluk bırakarak onun alt satırı olduğunu belirtir ve yukarda verdiğimiz değer üstünde bizim yazdığımız fonksiyonu uygular. Yani

def fonksiyon\_adı (bağlı değer[x gibi]) :

Fonksiyon

Örneğin ;

def kare\_al(x):

Print(x\*\*2)

**Bilgi Notuyla Çıktı Üretme:**

Bir çıktı elde ederken bir yazı ile birlikte çıktı almak istediğimiz zaman Print fonksiyonuna (“Çıktısı alınmak istenilen yazı” + str(Fonksiyonumuz) ) ile elde edilir . Peki hem işlem yapıcağımız sayı vb. değeri hem de fonksiyon işlem yaptıktan sonra elde ettiğimiz değeri görmek istiyorsak o zaman ;

Print( “ Çıktısı alınacak yazı “ + str(Bağlı değer)+ “çıktısı alınacak 2. Yazı” + str(fonksiyon) )

**İki Argümanlı Fonksiyon Tanımlamak:**

İki argüman( değişkenli ) fonksiyon tanımlamak istediğimizde kodu şu şekilde yazmamız gerekir

Def fonksiyonun\_adı(birinci değişken , ikinci değişken) :

Print(fonksiyon)

Örneğin ;

Def çarpma\_yap(x,y) :

Print(x\*y)

**Ön Tanımlı Argümanlar:** Kullanıcı istemeden bir argüman değerini girmediğini varsaydığımızda program hata vermesin diye önlem olarak argümana ön tanımlı değer atarız örneğin ;

Def çarpma\_yap (x=1 , Y= 1)

Print(x\*y)

Kullanıcı x değerini girip y değerini yazmayı unuttuğunu düşünürsek sonucu direkt olarak x değeri olarak verecektir

**Ne Zaman Fonksiyon Yazılır?**

Fonksiyonlar, tekrar eden görevleri yerine getirmek ve var olan işleri daha programatik şekilde gerçekleştirmek için kullanılır.

**Fonksiyon Çıktılarını Girdi Olarak Kullanmak (Return):**

Bir fonksiyonda eğer fonksiyon işleminin sonucunu başka bir yerde kullanmak istiyorsak yani başka bir işlemin girdisi olarak (o fonksiyon değerini 9 ile çarpmak vb. şekilde) işleme sokmak için Return ifadesi kullanılır. Yani:

Def fonksiyon\_adı ( 1.fonksiyon değeri,2. Fonksiyon değeri,3.fonksiyondeğeri)

Return fonksiyon

Örneğin;

Def direk\_ortalama (isi,nem,sarj):

Return (isi+nem)/sarj

Cikti = direk\_ortalama (25,40,70)

**Local ve Global Değişkenler:**

Local değişken fonksiyonun etkisi altında kalan değişkendir

Global değişkenler ise değişkene doğrudan atadığımız değerlerdir

**Local Etki Alanından Global Etki Alanını Değiştirmek:**

Fonksiyonun etkisi altında kalan değişkeni global olarak atayabilmek için yapılan işlemdir.Örneğin ;

Bir X isimli 0 elemanlı bir listemiz olduğunu var sayalım. Bu listeye fonksiyona verdiğimiz sayıyı yazmasını istiyoruz. Bunu şu şekilde yaparız:

Kod:

X =[]

Def eleman\_ekle(y):

x.append(y)

print(y) +” ifadesi eklendi.”)

eleman\_ekle(5)

Çıktı :

X=[5]

* ifadesi eklendi.
* **Karar Kontrol Yapıları :**

**True-False Sorgulamaları :**

Mantıksal sorgulamalardır. ( == ) denk mi eşit mi anlamına gelir. Kontrol ve şart tanımlamakta kullanılır.

**İf (Eğer), Else ve Elif Yapısı :**

İf = Eğer bu durum böyleyse şöyle yap der.İf’in sorguladığı şey true olarak döndüğü zaman çalışır.

Elif = Birden Fazla koşul olduğu zaman kullanılır.

Else = Eğer bu durum şöyle değilse böyle yap der.Else İse if false döndiğü zaman çalışır.

Kullanımı

İf koşul :

Yapmasını istediğimiz şey

Elif koşul:

Yapmasını istediğimiz şey

Else:

Yapmasını istediğimiz şey

Örneğin ;

İf gelir < sinir:

Print(“gelir sinirdan küçüktür.”)

Elif gelir == sinir:

Print(“gelir sinira eşittir”)

Else :

Print(“gelir sinirdan büyüktür”)

**Döngüler :**

**For Döngüsü :**

İteratif (yani tek tek seçme) işleri yapmak amaçlı kullanılır.

Kullanımı :

For i(değişken) in liste adı:

Yapmasını istediğimiz işlem

**Break ve Continue :**

.sort = Listedeki elemanları küçükten büyüğe sıralar.

Eğer döngüyü kesmek istiyorsak if koşulunun içine break yazarız

Bir ifadeyi atlamka istediğimizde ise if koşulunun içine continue yazarız

**While :**

Bu şart sağlandığı sürece demektir.Yani bir koşul devam ettiği sürece çalışmasını istediğimiz zaman kullanırız.

Kullanımı :

While değişken\_adı koşul:

Yapmasını istediğimiz işlem

Örneğin :

Sayi = 0

While sayi < 10 :

Sayi += 1

Print(sayi)