

Python 301

1.Fonksiyonlar

Fonksiyon Nasıl Yazılır?

```
def fonk_adi(paramtereler):  
    işlem
```

Fonksiyon Nasıl Çağırılır?

```
fonk_adi(parametre)
```

Not : Parametreler opsiyoneldir. Parametresiz çalışan fonksiyonlar da vardır.Örneğin print() fonksiyonu.

Fonksiyon Örneği : Sayının karesini bulup ekrana yazdıran fonksiyon yazalım.

```
def kare_bul(x):  
    print(x**2) → bu x^2 demektir
```

kare_bul(2) → 2 nin karesini bulacak.

Örnek2:Girilen 2 sayının çarpımını bulan fonksiyon

```
def carp(x,y):  
    print("Girilen 2 sayının çarpımı : " + str(x*y))  
carp(2,3)
```

Not : `print("Girilen 2 sayının çarpımı : " + str(x*y))` bunu yapmamızın sebebi bi taraf string diğer taraf int olacağı için bunları yan yana getiremiyoruz.o Yüzden ekrana gösterirken çarpma işleminin sonucunu stringe çevirdik

Fonksiyon Çıktılarını Girdi Olarak Kullanmak

Biz fonksiyonlarda `print` kullanarak ekrana birşey yaptırmak yerine orda yapmak istediğimiz şeyi hesaplatıp ihtiyacımız olan yerlerde o fonksiyonu çağırıp kullanıyoruz.Yani örneğin bize karenin alanını hesaplayacak fonksiyon lazım olsun;

```
def alan_hesapla(x):
```

```
    return x*x
```

buradaki `return` ifadesi alanı hesaplayıp bulduğu sonucu bize döndürmeye yarar.Biz bambaşka bi yerde kullanmak istediğimizde ise

`alan_hesapla(3)` şeklinde çağırıp burdaki sonucu kullanırız.

Local ve Global Değişkenler

Local değişkenler tanımlandıkları yerde geçerli olan değişkenlerdir.Örneğin fonksiyonda tanımlanmış bir değişken veya `if else` bloğunda tanımlanmış bir değişken

Global değişkenler ise fonksiyon,`if else` gibi yapıların dışında yani projenin ana kısımlarında tanımlanmış değişkenlerdir.

Buradaki mantık çok basit.İkisi de tanımlı olduğu blokta geçerli.Yani local değişken sadece fonsiyon gibi `if` gibi yapıların içinde kullanılabilir.Bunların dışında bir yerde bunları kullanamayız.Aşağıdaki örnek mantığı kavratacaktır(kanaatimce 🤓)

`x = 10` → global değişken

`y = 20` → global değişken

`def ornek_fonk(x,y)` → local değişken

`return x + y` → local değişken

#buradaki x ve y değerleri `ornek_fonk(3,4)` deki 3 ve 4 ten gelecek.

`ornek_fonk(3,4)`

`toplam2= x + y` → burdaki x ve y değerleri `x = 10` ve `y = 20` olacak.Yani global değişkenden gelecek.

2.Karar Kontrol Yapıları

if Yapısı

Kelime anlamı olarak if **eğer** demek.Yani **eğer koşul sağlanıyorsa bu bloğa girip işlemleri yapacak**.Kullanımı ;

if **koşul**:

koşul doğru ise yapılacak işlemler

if ... else yapısı

Koşulun **true** (doğru) olduğu durumlarda **if**, **false** (yanlış) olduğu durumlarda ise **else** ifadesi yürütülür.

if **koşul**:

koşul doğru ise yapılacak işlemler

else:

koşul yanlış ise yapılacak işlemler

if...elif...else yapısı

if...elif ifadesindeki herhangi bir koşul sağlandığında o girintideki kodlar yürütülür. İstersek if...elif bloğunun en altına else ifadesi de yazabiliriz.

if **koşul**:

koşul doğru ise yapılacak işlemler

elif **koşul**:

koşul doğru ise yapılacak işlemler

else:

koşul yanlış ise yapılacak işlemler

3.Döngüler

1.For Döngüsü

Basitçe kullanım amacı , **for döngüsü belirli bir koşul sağlanana kadar tekrar tekrar aynı işlemi yapmamızı sağlar**.Kullanımı anlamak açısından basit bi örnek yapacak olursak ;

```
dizi = ["Nigar" , "derslerinde" , "başarılı" , "olsun" , "inşallah" , "amin" ]
```

```
for i in dizi
```

```
    print(i)
```

Bu kod listenin elemanlarını tek tek gezip hepsini ekrana yazdıracak.Yani bizi

```
print(liste[0])
```

```
print(liste[1])
```

```
print(liste[2])
```

```
print(liste[3])
```

```
print(liste[4])
```

```
print(liste[5])
```

yazmaktan kurtaracak

Break ve Continue

Bi döngüde veya if else yapısında bi işlem yaparken belirli şartlar sağlandığında bu döngü veya if bloğu bitirmek istenebilir veya bu şartı sağlayan şey **görmezden gelinmek** istenebilir. Diğer yandan belirli şartlara göre **döngü kırılmak** istenebilir. Örnek üzerinde daha iyi anlaşılacaktır.

break örneği : Her bir maaşı gezicez ancak maaşı 3000 den fazla olan kişi görmezden gelinecek. Yani maaş 3000 olduğunda biz döngüden çıkmak istiyoruz.

```
maaslar = [8000,5000,2000,1000,3000,7000,1000]
```

```
maaslar.sort() → maasların tüm elemanlarını küçükten büyüğe sıralama
```

```
for i in maaslar:
```

```
    if i == 3000:
```

```
        print("kesildi")
```

```
        break → maaş 3000 e geldiğinde döngüyü kırıp çıkacak
```

continue örneği : Maaşlara zam yapmak istiyoruz. Maaşların içinde gezerken 3000 maaş alanlara geldiğinde onlara işlem yapmayalım.

```
maaslar = [8000,5000,2000,1000,3000,7000,1000]
```

```
maaslar.sort()
```

```
for i in maaslar:
```

```
    if i == 3000:
```

```
        print("geçildi")
```

```
        continue
```

```
#burada da(if bloğundan çıktık burda) maaşa zam yapma kodları yazılır
```

2.While Döngüsü

Şart sağlandığı sürece döngü çalışır.Eğer şans sonsuza kadar sağlanıyorsa sonsuza kadar bile çalışıp bilgisayarı yakar. 💻 🔥 🧯

```
sayi = 5
```

```
while sayi<10:
```

```
    print(sayi)
```

bunu böyle çalıştırırsak sonsuz döngüye girer.o yüzden bunu şu şekilde kullanırız;

```
sayi = 5
```

```
while sayi<10
```

```
    print(sayi)
```

sayi -= 1 → print işlemlerinden sonra sayi yi 1 azaltıp koşul sağlanıyor mu diye bakacak.Eğer koşul sağlanıyorsa aynı şekilde devam edecek,sağlanmıyorsa döngü bitecek.