

FONKSİYONLAR

Fonksiyon nedir?
Fonksiyon tanımlama
Parametresiz fonksiyonlar
Parametrelili fonksiyonlar
Değer dönderen fonksiyonlar

Fonksiyon nedir?

Fonksiyonlar, belirli işlemleri gerçekleştiren kod bloğudur.

Fonksiyonlar tanımlandıktan sonra istediğimiz kadar kullanabiliriz.

Fonksiyon Tanımlama

Fonksiyonları program içerisinde kullanabilmek için iki aşama vardır.

1-Fonksiyon tanımlanmalı

2-Fonksiyon çağrılmalı

Fonksiyonlar

```
graph TD; A[Fonksiyonlar] --> B[Parametresiz Fonksiyonlar]; A --> C[Parametrelili Fonskiyonlar];
```

Parametresiz
Fonksiyonlar

Parametrelili
Fonskiyonlar

PARAMETRESİZ FONKSİYONLARIN KULLANIMI

Bu tür fonksiyonları tanımlarken fonksiyon ismi yanındaki parantez içerisine parametre girilmez. Fonksiyon ,içerisinde yer alan komutları icra eder

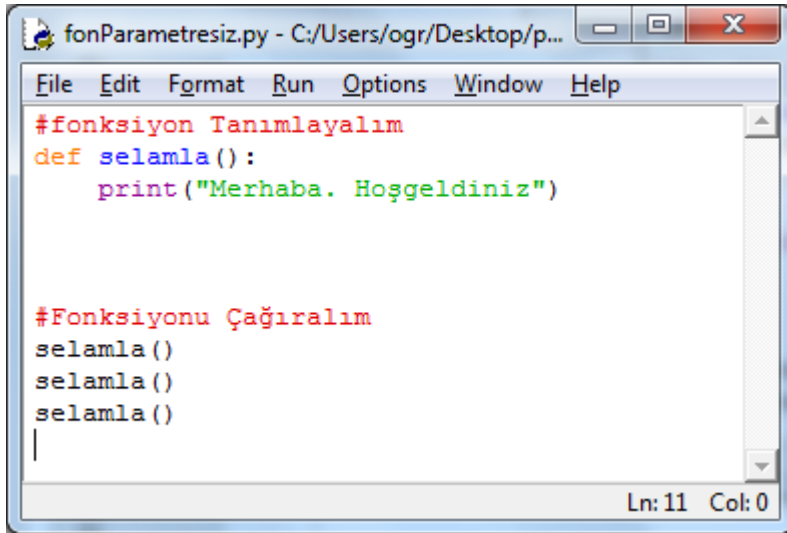
```
def fonksiyon_ismi():  
    Fonksiyona ait kodlar  
    Fonksiyona ait kodlar  
    Fonksiyona ait kodlar
```

def: Bu ayrılmış sözcük ile fonksiyon tanımlama başlar.

İsim: Fonksiyon için bir isim verilmelidir. Aynı değişken tanımlamada olduğu gibi. Fonksiyon isminden sonra parantez () konulmalıdır

Gövde: Fonksiyon için gerekli olan kod blokundan oluşur.

Parametresiz Fonk Örnek

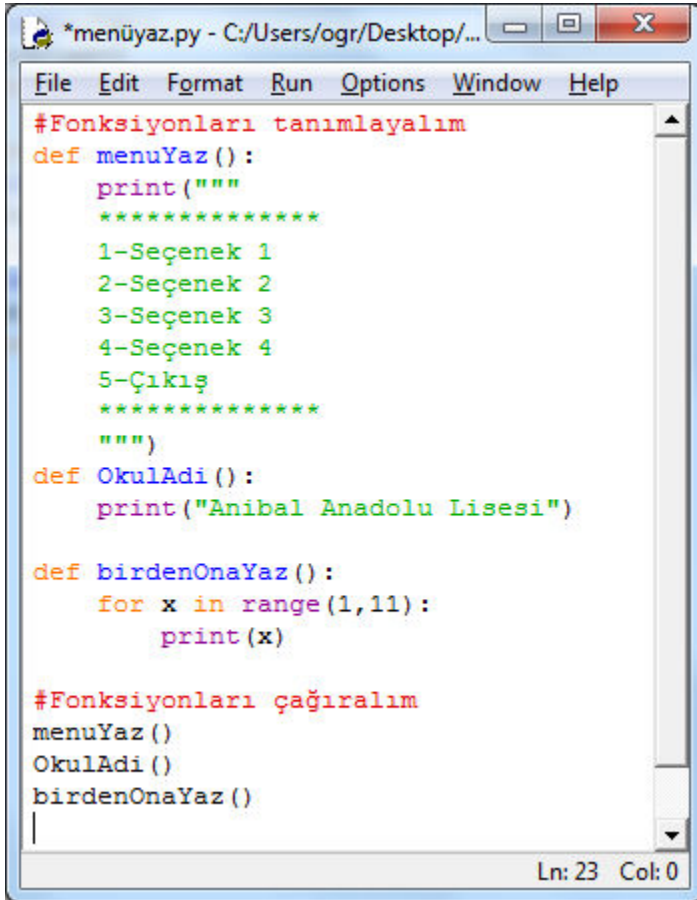


```
fonParametresiz.py - C:/Users/ogr/Desktop/p...
File Edit Format Run Options Window Help
#fonksiyon Tanımlayalım
def selamla():
    print("Merhaba. Hoşgeldiniz")

#Fonksiyonu Çağıralım
selamla()
selamla()
selamla()
|
Ln: 11 Col: 0
```

```
Merhaba. Hoşgeldiniz
Merhaba. Hoşgeldiniz
Merhaba. Hoşgeldiniz
>>>
```

Parametre girilmeyen fonksiyonlara uyan örnek yapalım



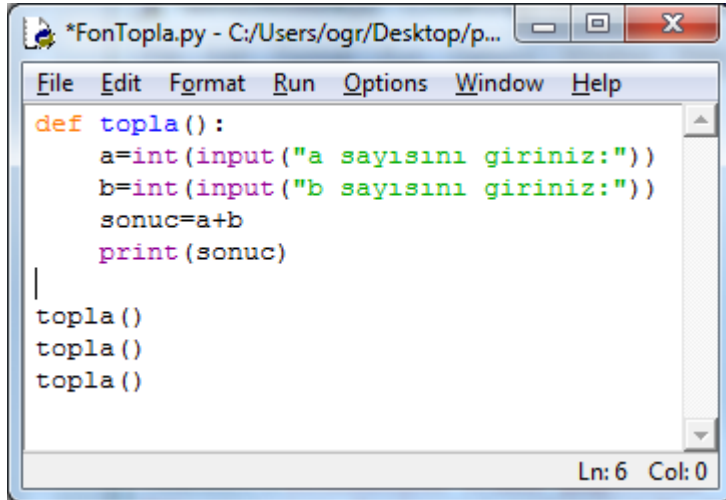
```
*menüyaz.py - C:/Users/ogr/Desktop/...
File Edit Format Run Options Window Help
#Fonksiyonları tanımlayalım
def menuYaz():
    print("""
*****
1-Seçenek 1
2-Seçenek 2
3-Seçenek 3
4-Seçenek 4
5-Çıkış
*****
""")
def OkulAdi():
    print("Anibal Anadolu Lisesi")
def birdenOnaYaz():
    for x in range(1,11):
        print(x)

#Fonksiyonları çağıralım
menuYaz()
OkulAdi()
birdenOnaYaz()
|
Ln: 23 Col: 0
```

```
*****
1-Seçenek 1
2-Seçenek 2
3-Seçenek 3
4-Seçenek 4
5-Çıkış
*****
```

```
Anibal Anadolu Lisesi
1
2
3
4
5
6
7
8
9
10
.
```

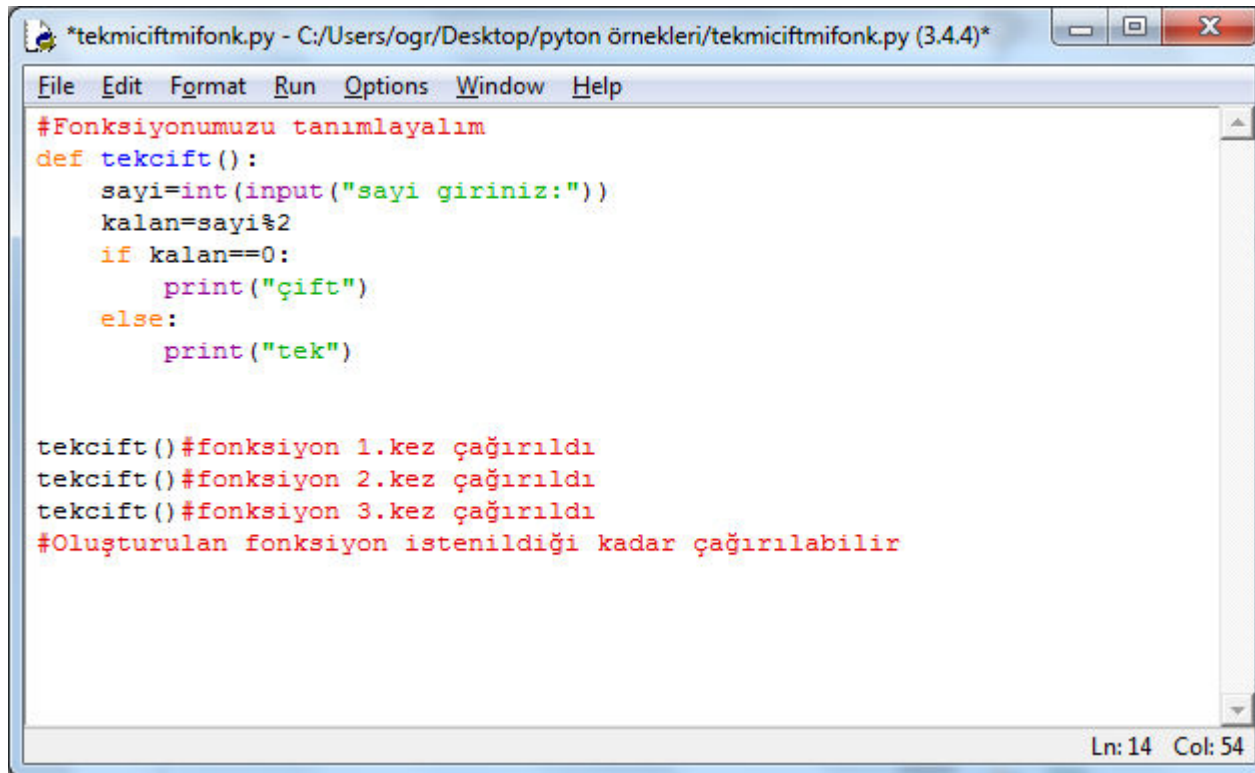
İki sayıyı toplayan parametresiz fonksiyonu oluşturalım



```
*FonTopla.py - C:/Users/ogr/Desktop/p...
File Edit Format Run Options Window Help
def topla() :
    a=int(input("a sayısını giriniz:"))
    b=int(input("b sayısını giriniz:"))
    sonuc=a+b
    print(sonuc)

topla()
topla()
topla()
Ln: 6 Col: 0
```

Tek mi Çift Mi Fonksiyonu Oluşturalım



```
*tekmiciftmifonk.py - C:/Users/ogr/Desktop/pyton örnekleri/tekmiciftmifonk.py (3.4.4)*
File Edit Format Run Options Window Help
#Fonksiyonumuzu tanımlayalım
def tekciift():
    sayi=int(input("sayi giriniz:"))
    kalan=sayi%2
    if kalan==0:
        print("çift")
    else:
        print("tek")

tekciift()#fonksiyon 1.kez çağırıldı
tekciift()#fonksiyon 2.kez çağırıldı
tekciift()#fonksiyon 3.kez çağırıldı
#Oluşturulan fonksiyon istenildiği kadar çağırılabilir

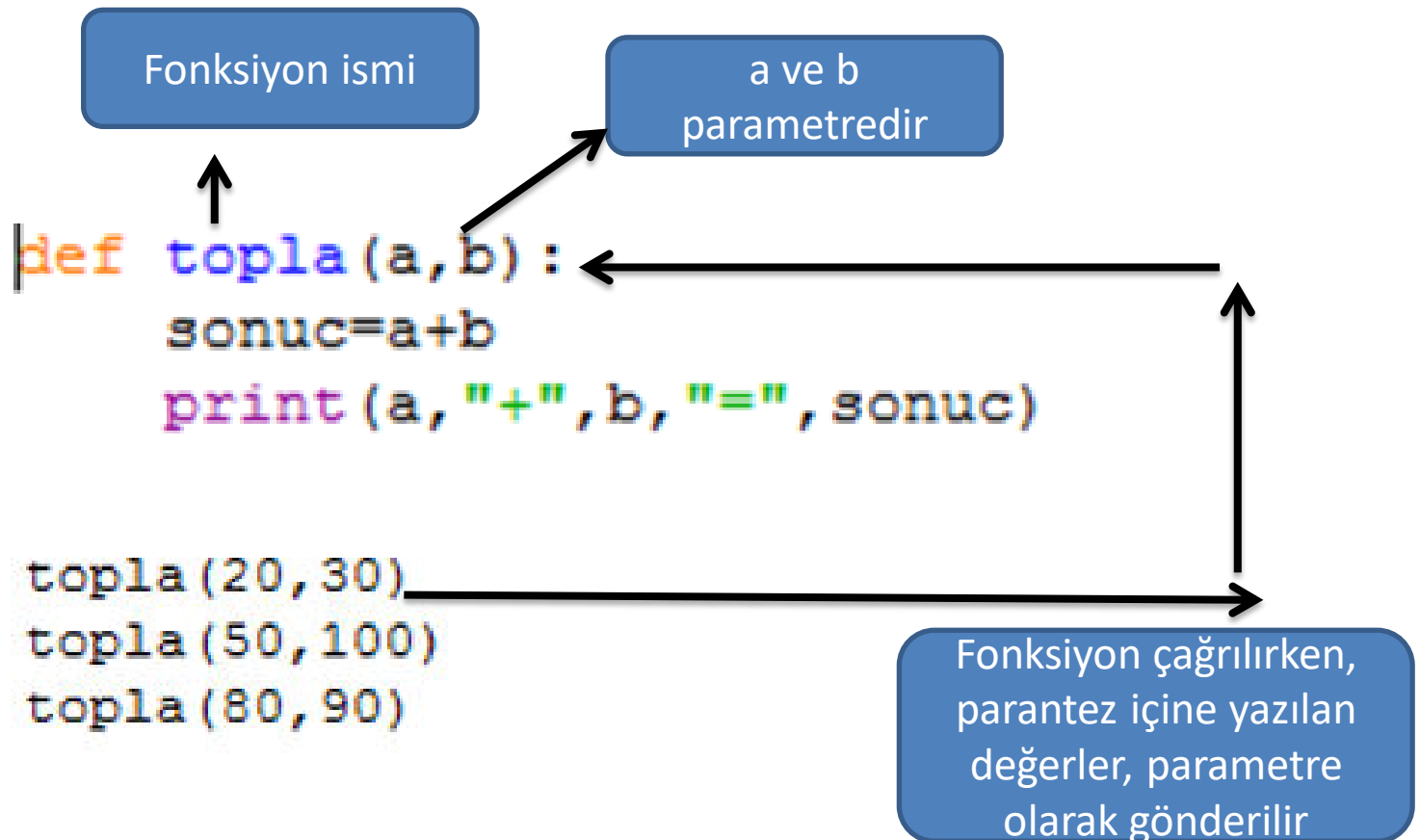
Ln: 14 Col: 54
```

Alıştırma

- Çarpma , bölme, mod alma işlemlerini yerine getiren parametresiz fonksiyonları oluşturunuz.

PARAMETRE ALAN FONKSİYONLAR

Parametre alan fonksiyonlar tanımlanırken, Fonksiyon isminin yanındaki parantez içine ,kullanılacak parametreler virgülle ayrılarak yazılmalıdır. Parametreler fonksiyon içerisindeki kodlarda kullanılır



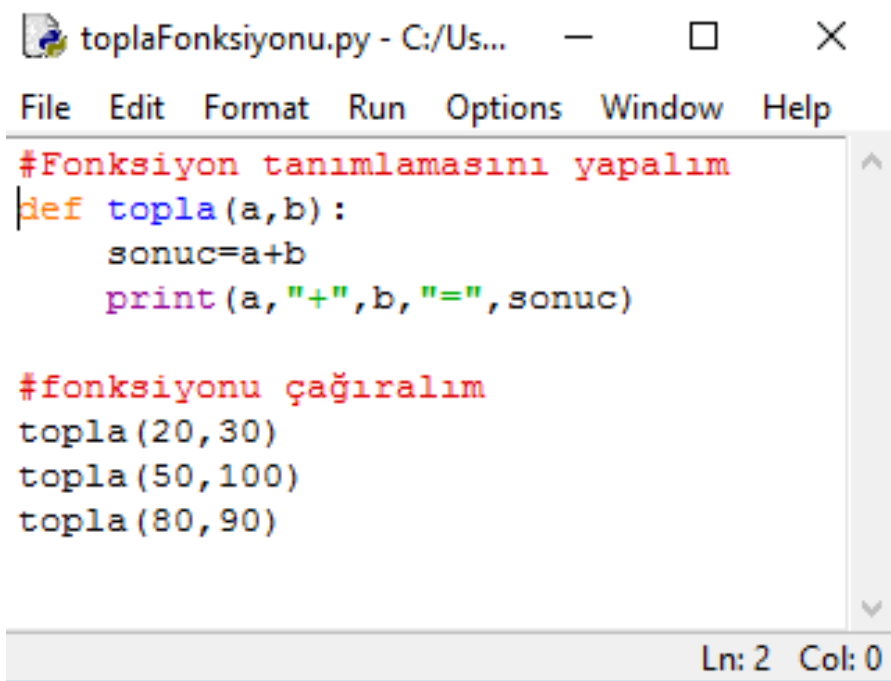
Parametre olarak gönderilen isme
«Hoş geldin isim» şeklinde cevap
veren fonsiyonu yazalım

```
def selamla(isim):  
    print("Hoş geldin ",isim)
```

```
selamla("Ahmet")  
selamla("Mehmet")  
selamla("Fatma")  
|
```

```
Hoş geldin Ahmet  
Hoş geldin Mehmet  
Hoş geldin Fatma  
>>> |
```

Toplama işlemi yapan fonksiyon yapalım



```
toplaFonksiyonu.py - C:/Us...  
File Edit Format Run Options Window Help  
#Fonksiyon tanımlamasını yapalım  
def topla(a,b):  
    sonuc=a+b  
    print(a,"+",b,"=",sonuc)  
  
#fonksiyonu çağıralım  
topla(20,30)  
topla(50,100)  
topla(80,90)  
Ln: 2 Col: 0
```

```
20 + 30 = 50  
50 + 100 = 150  
80 + 90 = 170  
>>> |
```

Fonksiyonları Kullanarak Matematiksel işlemler yapan fonksiyonu oluşturalım

```
def topla(a,b):  
    sonuc=a+b  
    print("Toplama İşleminin Sonucu",sonuc)  
  
def cikar(a,b):  
    sonuc=a-b  
    print("Çıkarma İşleminin Sonucu=",sonuc)  
  
def carp(a,b):  
    sonuc=a*b  
    print("Çarpma İşleminin sonucu=",sonuc)  
  
def bol(a,b):  
    sonuc=a/b  
    print("Bölme İşleminin sonucu=",sonuc)  
  
def us(a,b):  
    sonuc=a**b  
    print("Üs Alma İşleminin sonucu=",sonuc)  
  
def mod(a,b):  
    sonuc=a%b  
    print("Mod Alma İşleminin sonucu=",sonuc)  
  
def karekok(a):  
    sonuc=a**0.5  
    print("Karakök  
        İşleminin sonucu=",sonuc)  
  
topla(10,20)  
cikar(70,35)  
carp(5,10)  
bol(100,10)  
us(5,3)  
mod(15,3)  
karekok(25)
```

```
Toplama İşleminin Sonucu 30  
Çıkarma İşleminin Sonucu= 35  
Çarpma İşleminin sonucu= 50  
Bölme İşleminin sonucu= 10.0  
Üs Alma İşleminin sonucu= 125  
Mod Alma İşleminin sonucu= 0  
Karakök İşleminin sonucu= 5.0,  
... !
```

Öğrenci Bilgisi yazan fonksiyon tanımlayalım

File Edit Format Run Options Window Help

```
#FONKSİYON TANIMLAMASI YAPIYORUZ
```

```
def ogrenci(ad,soyad,no,sinif):  
    print("=====")  
    print("Öğrencinin adı:",ad)  
    print("Öğrencinin soyadı:",soyad)  
    print("Öğrencinin Numarası:",no)  
    print("Öğrencinin Sınıfı:",sinif)  
    print("=====")
```

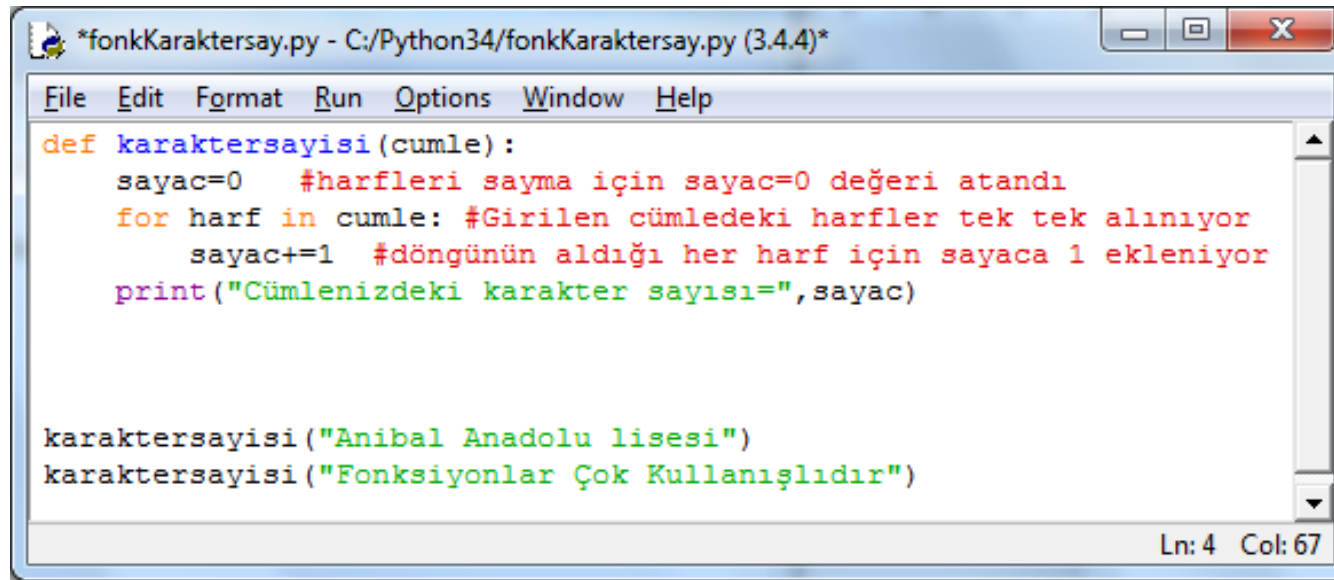
```
#HAZIRLADIĞIMIZ FONKSİYONU İSTEDİĞİMİZ KADAR ÇAĞIRABİLİRİZ
```

```
ogrenci("Ahmet","ÇELİK",363,"9H")# fonksiyon çağrıldı ve parametreler gönderildi
```

```
ogrenci("Ali","SOYLU",654,"9H")#Fonksiyonu istediğimiz kadar çağırabilir ve kullanabiliriz
```

Fonksiyona
gönderilen
parametreler
birden fazla ise
virgül ile ayrılır

Karakter sayısını bulan kendimize ait fonksiyonu tanımlayalım

A screenshot of a Python IDE window titled '*fonkKaraktersay.py - C:/Python34/fonkKaraktersay.py (3.4.4)*'. The window has a menu bar with 'File', 'Edit', 'Format', 'Run', 'Options', 'Window', and 'Help'. The code editor contains the following Python code:

```
def karaktersayisi(cumle):  
    sayac=0    #harfleri sayma için sayac=0 değeri atandı  
    for harf in cumle: #Girilen cümledeki harfler tek tek alınıyor  
        sayac+=1    #döngünün aldığı her harf için sayaca 1 ekleniyor  
    print("Cümledeki karakter sayısı=",sayac)  
  
karaktersayisi("Anibal Anadolu lisesi")  
karaktersayisi("Fonksiyonlar Çok Kullanışlıdır")
```

The status bar at the bottom right shows 'Ln: 4 Col: 67'.

Girilen yazıdaki karakter sayısı= 21

Girilen yazıdaki karakter sayısı= 30

>>> |

Belirtilen sayılar arasındaki sayıları toplayan programı oluşturalım

```
def arasiTopla(x,y):  
    toplam=0;  
    for sayi in range(x,y+1):  
        toplam+=sayi  
    print(x,"ile", y," arasındaki sayıların toplamı=",toplam)
```

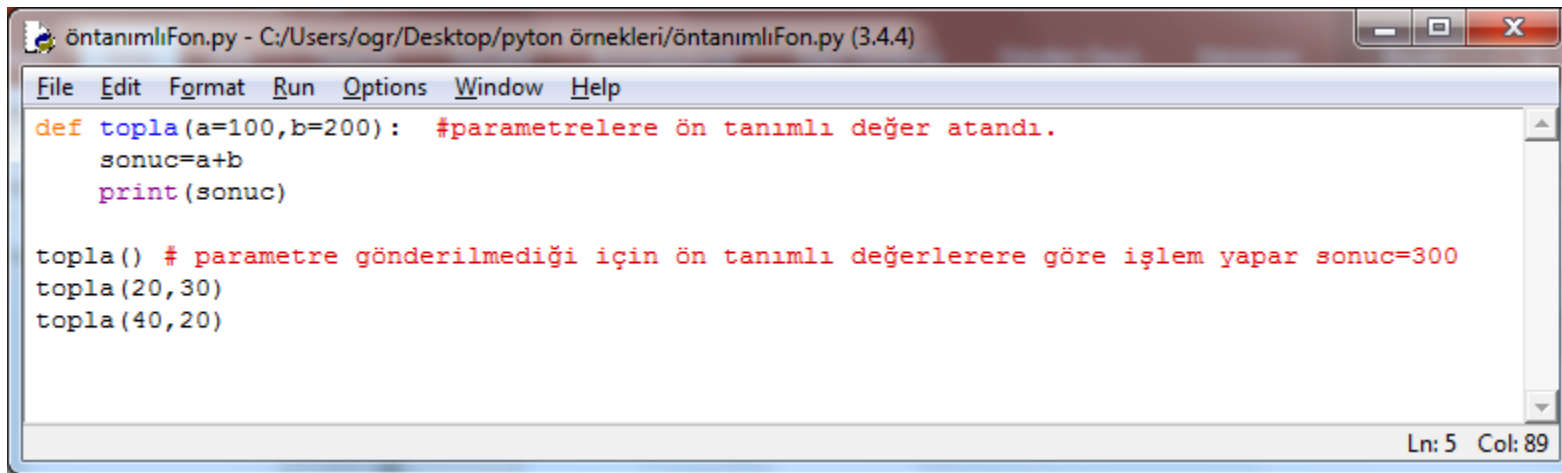
```
arasiTopla(1,10)  
arasiTopla(1,5)
```

```
|
```

```
1 ile 10  arasındaki sayıların toplamı= 55  
1 ile 5  arasındaki sayıların toplamı= 15  
... |
```

Fonksiyon Oluştururken Parametreye varsayılan Değer Atama

Fonksiyon içerisindeki parametrelere ön tanımlı değer atanabilir. Aşağıdaki Örneği İnceleyiniz



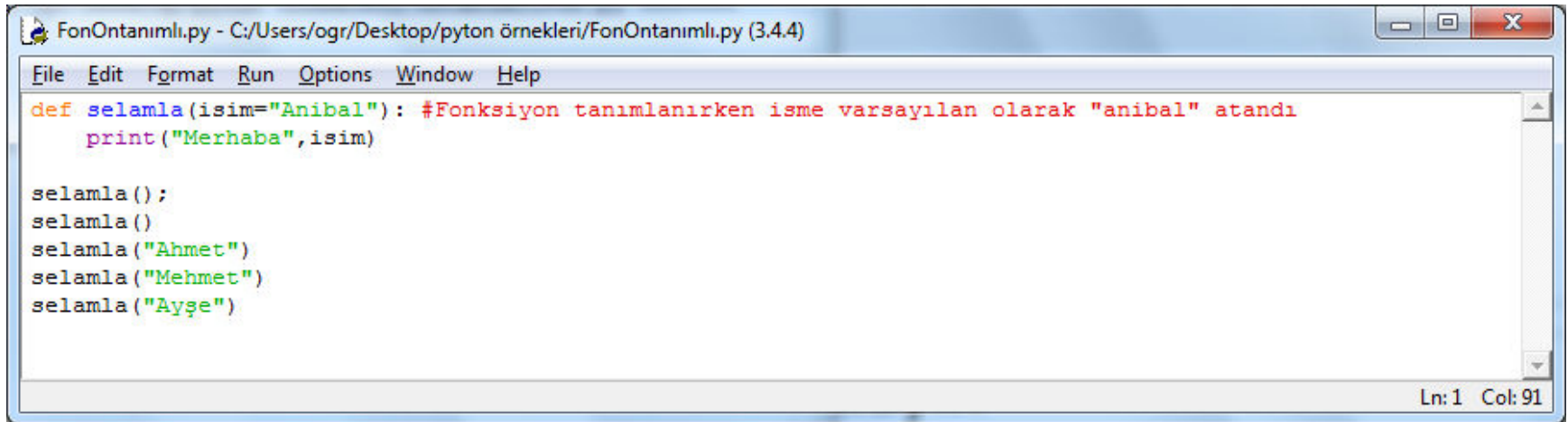
```
öntanımlıFon.py - C:/Users/ogr/Desktop/pyton örnekleri/öntanımlıFon.py (3.4.4)
File Edit Format Run Options Window Help
def toplama(a=100,b=200): #parametrelere ön tanımlı değer atandı.
    sonuc=a+b
    print(sonuc)

toplama() # parametre gönderilmediği için ön tanımlı değerlerere göre işlem yapar sonuc=300
toplama(20,30)
toplama(40,20)
```

Ln: 5 Col: 89

```
300
50
60
>>>
```

Varsayılan Değerli Fonksiyona Örnek



The screenshot shows a Python IDE window titled "FonOntanımlı.py - C:/Users/ogr/Desktop/pyton örnekleri/FonOntanımlı.py (3.4.4)". The window contains the following Python code:

```
File Edit Format Run Options Window Help
def selamla(isim="Anibal"): #Fonksiyon tanımlanırken isme varsayılan olarak "anibal" atandı
    print("Merhaba", isim)

selamla();
selamla();
selamla("Ahmet");
selamla("Mehmet");
selamla("Ayşe");
```

The status bar at the bottom right indicates "Ln: 1 Col: 91".

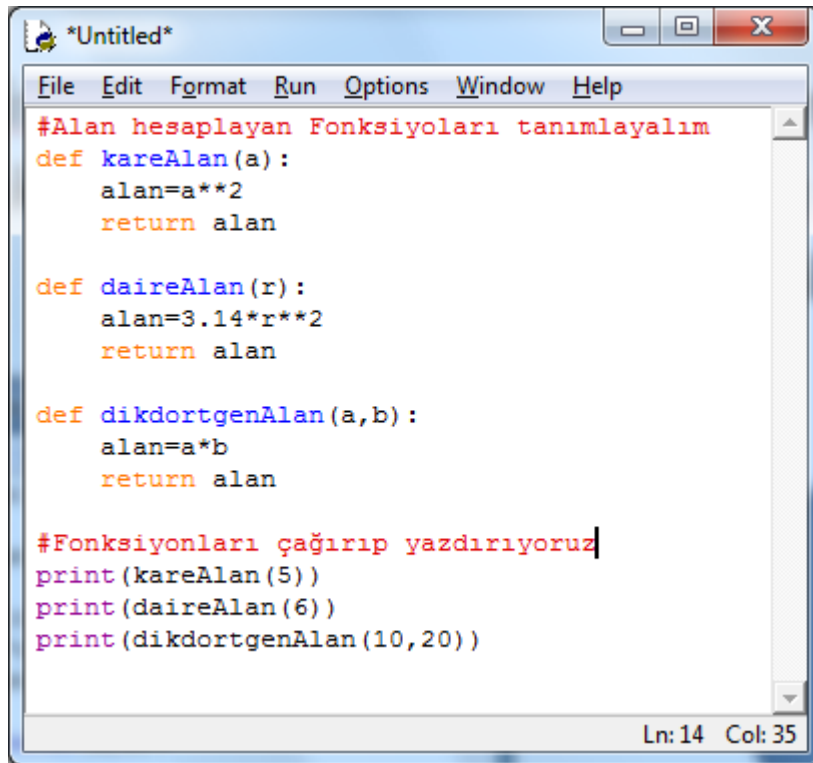
Return ile fonksiyondan değer döndürme

Fonksiyon içerisinde işlenip, üretilen değerler, programın herhangi bir yerinde kullanılmak istenirse «return» komutu ile gönderilmelidir.

ÖRNEK UYGULAMA

```
def topla(a,b):  
    toplam=a+b  
    return toplam # Return ile bulunan toplam değeri geri gönderiliyor  
  
def cikar(a,b):  
    fark=a-b  
    return fark # Return ile bulunan fark değeri geri gönderiliyor  
  
def carp(a,b):  
    sonuc=a*b  
    return sonuc # Return ile bulunan sonuc değeri geri gönderiliyor  
  
sonuc1=topla(10,20) # topla fonksiyonu çağırıldı parametre olarak 10 ve 20  
                    # değerleri gönderildi. Yukarıdaki fonksiyon toplama işlemini  
                    # yaptı ve bulunan sonuc return ile gönderilip sonuc1 değişkenine atandı  
sonuc2=cikar(60,20)  
sonuc3=carp(60,3)  
  
print("toplama işleminin sonucu", sonuc1)  
print("Çıkarma işleminin sonucu", sonuc2)|  
print("Çarpma işleminin sonucu", sonuc3)
```

Dairenin, karenin, ve dikdörtgenin alanını bulan fonksiyonları oluşturup ,return ile değerleri geri gönderip kullanalım



```
File Edit Format Run Options Window Help
#Alan hesaplayan Fonksiyonları tanımlayalım
def kareAlan(a):
    alan=a**2
    return alan

def daireAlan(r):
    alan=3.14*r**2
    return alan

def dikdortgenAlan(a,b):
    alan=a*b
    return alan

#Fonksiyonları çağırıp yazdırıyoruz
print(kareAlan(5))
print(daireAlan(6))
print(dikdortgenAlan(10,20))

Ln: 14 Col: 35
```

Programı çalıştırdıktan sonra fonksiyon içerisindeki return ile başlayan satırları silip tekrar çalıştıralım. Farkı görelim.....

Girilen fiyatın kdv'sini hesaplayan ve
return ile değer döndüren fonksiyonu
oluşturalım($kdv = fiyat * yüzde / 100$)

```
def kdv(fiyat,yuzde):  
    kdvsi=(fiyat*yuzde/100)  
    return kdvsi
```

```
print(kdv(1000,18))  
print(kdv(100,5))
```

```
|
```

$$\sqrt{3^2 + 4^2}$$

Bu denkleme uygun çözümü fonksiyonlar kullanarak yapalım

```
def kok(sayi):  
    sonuc=sayi**0.5  
    return sonuc
```

```
def usal(sayi,us):  
    sonuc=sayi**us  
    return sonuc
```

```
#denklem için gerekli kodları yazalım
```

```
c=kok(usal(3,2)+usal(4,2))
```

```
print("bulunan değer=",c)
```

Fonksiyon Tanımlamanın Faydaları

- Programın yönetimi kolaylaşır.
- Program daha anlaşılır olur
- Daha kolay hata ayıklama yapılabilir.
- Kod satırlarını değiştirmek/genişletmek kolaylaşır.
- Oluşturulan fonksiyonlar istenildiği kadar kullanılabilir.