

In []:

1

FOKSIYONLAR(FUNCTIONS): FONSIYONLAR KULLANIMA ÖNCEDEN HAZIRLANMIŞ FORMÜLLERDİR, HIZLI VE AZ HATALI BİR ŞEKİLDE TEKRARLI KULLANIM OLANAĞI SAĞLAR

SCOPE: LOCAL===>GLOBAL===>BULTIN

LOCAL: Fonksiyon içinde tanımlanan değişkendir ve fonksiyon dışında bir anlamı yoktur.

GLOBAL :Fonksiyon dışında tanımlanmış(aynı sayfa (kernel) içinde) ama fonksiyon içinde de kullanılan değişkenlerdir.

BULTIN : PYTHON'UN kendi içersinde tanımlanmış değişkenlerdir(Pi sayısı gibi).

fonksiyonu tanımlarken def ile başlıyoruz parantez içersinde kullanacağımız değişkenleri yazıyoruz.

In []:

```
1 # matematik işlemleri ni yapmak için math kütüphanesini yüklüyoruz
2 import math
```

```
In [ ]: 1 # foo() fonksiyonu x,y ve pi sayısını yazdırır x,y,pi
        2 def foo():
        3     global x_1
        4     x_1 = 36
        5     x_2=11
        6     print(x_1,x_2,math.pi)
```

```
In [ ]: 1 # foo fonksiyonunu çalıştırsak
        2 foo()
```

```
In [ ]: 1 # x'i yazdırabilir ve her türlü işlemde kullanabiliriz çünkü global değişken
        2 print(x_1)
```

```
In [ ]: 1 # y' yi yazdıramayız çünkü local değişken
        2 print(x_2)
```

def func_name(var1,var2,...):

operations

return

```
In [ ]: 1 #ders_geçme fonksiyonu
2 def ders_geçme(final_notu,ev_ödevi_notu,ara_sınav_notu):
3     final = final_notu*0.60
4     ev_ödevi = ev_ödevi_notu*0.10
5     ara_sınav = ara_sınav_notu*0.30
6     toplam = final + ev_ödevi + ara_sınav
7     if toplam >= 70:
8
9         print('Toplam not {} , Tebrikler dersini geçtin'.format(toplam))
10
11     else:
12
13         print('Toplam not {} , Üzgünüm dersini geçemedin'.format(toplam))
14
```

```
In [ ]: 1 ders_geçme(final_notu=60,ev_ödevi_notu=60,ara_sınav_notu=60)
```

```
In [ ]: 1 ders_geçme(60,60,60)
```

```
In [ ]: 1 ders_geçme(90,60,60)
```

```
In [ ]: 1 # Şifre doğrulama fonksiyonu
2 def şifre_doğrulama(şifre):
3     şifre = str(şifre)
4     harf_sayısı =0
5     # higher or equal to 8 and lower or equal to 16
6     if (len(şifre) < 8) | (len(şifre)>16):
7
8         print("Şifreniz 8 ile 16 haneden oluşmadığı için kabul edilmedi")
9     else:
10        for i in şifre:
11            if i.isalpha()== True:
12                harf_sayısı = harf_sayısı +1
13            if harf_sayısı!=1:
14                print("Şifreniz en az bir harf içermediğinden dolayı kabul edilmedi")
15            else:
16                print("Şifre kabul edildi")
```

```
In [ ]: 1 şifre_doğrulama(1234567)
```

```
In [ ]: 1 şifre_doğrulama('12345678')
```

```
In [ ]: 1 şifre_doğrulama('1234567A')
```

EĞER FONSIYON SONUCUNUN GÖZÜKMESİNİ İSTİYORSAK RETURN KELİMESİ İLE GÖSTERMEK İSTEDİĞİMİZ SONUCU BELİRTMELİYİZ

```
In [9]: 1 # Eğer return ifadesini koymazsak fonksiyon işlemi yapar ama sonuç gözükmez ve işlemde kullanılamaz
2 def log_ret_A(P0,P1):
3     ret = math.log(P1/P0)*100
4     ret = round(ret,2)
5
```

```
In [10]: 1 # Eğer return ifadesini kullanırsak fonksiyon işlemi yapar ve sonuç gözükür ve işlemlerde kullanabiliriz
2 def log_ret(P0,P1):
3     ret = math.log(P1/P0)*100
4     ret = round(ret,2)
5     return ret
```

```
In [11]: 1 # fonksiyon sonucu yazılmayacaktır
2 log_ret_A(100,110)
3 print(log_ret_A(100,110))
```

NameError Traceback (most recent call last)

~\AppData\Local\Temp\ipykernel_14068\3626328953.py in <module>

```
1 # fonksiyon sonucu yazılmayacaktır
----> 2 log_ret_A(100,110)
3 print(log_ret_A(100,110))
```

~\AppData\Local\Temp\ipykernel_14068\4085024584.py in log_ret_A(P0, P1)

```
1 # Eğer return ifadesini koymazsak fonksiyon işlemi yapar ama sonuç gözükmez ve işlemde kullanılamaz
2 def log_ret_A(P0,P1):
----> 3     ret = math.log(P1/P0)*100
4     ret = round(ret,2)
5
```

NameError: name 'math' is not defined

```
In [ ]: 1 # İşlemlerde kullanılamıyacaktır
        2 log_ret_A(100,110)+1
```

```
In [ ]: 1 # sonucu yazdırabiliriz
        2 log_ret(100,110)
```

```
In [ ]: 1 # sonucu işlemlerde de kullanabiliriz
        2 log_ret(100,110)+1
```

FONSIYONLARI LOOP İÇERSİNDE KULLANABİLİRİZ

```
In [ ]: 1 # portföyümüzdeki hisse isimleri, fiyat bilgileri ve fiyat bilgilerin tarihleri
        2 faang_stocks_getileri = {'sembol':['FB','AMZN','AAPL','NFLX','GOOGL'],
        3                             'fiyat':[[338.54,160.03],[170.40,108.92],[181.26,136.53],[597.37,174.87],[145.07,109.37]]
        4                             'tarih':['01-03-2022','06-30-2022']}
```

```
In [ ]: 1 # log_ret() fonksiyonu döngü içinde kullanılabilir
        2 for i,j in zip(faang_stocks_getileri['sembol'],faang_stocks_getileri['fiyat']):
        3
        4     print("{} hisse senedi getirisi {}".format(i,round(log_ret(j[0],j[1]),2)))
```

```
In [ ]: 1
```