



# HOŞGELDİNİZ! :)

Bi takım işlevsel işler : Fonksiyon

# BİR HATI ALATMA

- Sınıfa **ağız açık kaplarda** içeceklerle LÜTFEN gelmeyiniz (Çay, kahve...)
  - Tetikli Termos, Kapaklı Su şişesi...
- Ayrılırken;
  - + Masa üstlerinde LÜTFEN ÇÖP BIRAKMAYINIZ,
  - + Bilgisayarlarınızı LÜTFEN KAPATINIZ,
  - + Sandalyelerinizi LÜTFEN DÜZENLİ bırakınız,
- Ders başlangıcı : Max. 5 Dk. Tolerans...  
**19:05 Ders başlar!**

[illegible]

# Ne ola ki bu Fonksiyon

- Geliştirici tarafından tasarlanan,
- Kendine ait bir bloğu olan
- `def` anahtar kelimesi ile tanımlanan,
- Parametre/Arguman alan/almayan
- Sonuc donduren/dondurmeyen (Biz nasıl istersek...)

# Bir metodun hayatı...

*#Oluşturma*

```
def myFunction():  
    print("Hello World")
```

*#Çağırma*

```
myFunction()
```

# Ama niye...

- # Benzer işlerin kodunu tekrar tekrar yazmamak için*
- # Okunabilir kod yazmak için*
- # Ortalıktaki karışıklığı azaltmak için*
- # Kodun modüler olabilmesi için*

# Bir adım ileri: Argümanlar

*# Fonksiyonda kullanılmak üzere veri vermek için*

*# Adet sınırı yoktur, ne kadar lazımsa ama virgülle ayrılması gerekir,*

*# Fonksiyonlar, önce tanımlanır, sonra kullanılır!!!*

# Func (Args)

*#Oluşturma*

```
def myFunction(variable):  
    print(variable , "World")
```

*#Çağırma*

```
myFunction("Python")
```



# Func (Args)

*#Oluşturma*

```
def myFunction2(isim, soyisim):  
    print(isim , soyisim)
```

---

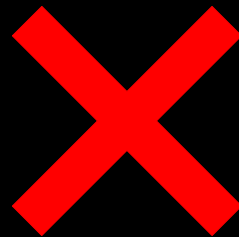
*#Çağırma*

```
myFunction2(isim, soyisim)
```

---

*#ÇALIŞMAZ !!!*

```
myFunction2(isim)
```



# Func (Args)

*#Oluşturma*

```
def myFunction(isim, soyisim):  
    print(isim , soyisim)
```

---

```
def myFunction(isim):  
    print(isim , "Hoşgeldiniz")
```

---

*# KANDIRAMAZSINIZ, YEMEZ!!!*



# Func (Args)

*#Oluşturma*

```
def myFunction(a,b):  
    c = a + b  
    return c
```

*#Çağırma*

```
sonuc = myFunction(3,5)  
print(sonuc)
```

*#ya da*

```
print(myFunction(3,5))
```

**Bu sevdâ bitmez!**

**Az kaldı!!!**

- [illegible]

# Ne ola ki bu “Scope/Kapsam”

- Bir değişken sadece ve sadece yaratıldığı alanın içerisinde geçerlidir. Buna scope/kapsam denir.
- Bir fonksiyonun içinde yaratılan bir değişken, o fonksiyonun local scope/yerel kapsamı na girer ve sadece o fonksiyon içinde kullanılabilir!!!
- Fonksiyonun dışında tanımlanan değişken, fonksiyon içinden erişilebilir ancak değiştirilemez.
- Fonksiyonun dışında tanımlanmış bir değişkeni değiştirmek için *global* anahtar kelimesi kullanılır,

# Scope / Kapsam

```
isim = "Python"
```

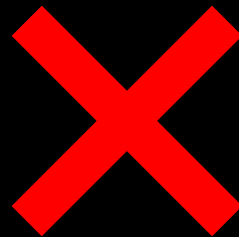
```
def isimYaz (var) :  
    print(var)
```

*#Çağırma*

```
myFunction2 (isim, soyisim)
```

*#ÇALIŞMAZ !!!*

```
myFunction2 (isim)
```



**SİZ FARKINDA  
OLMASANIZ DA...**



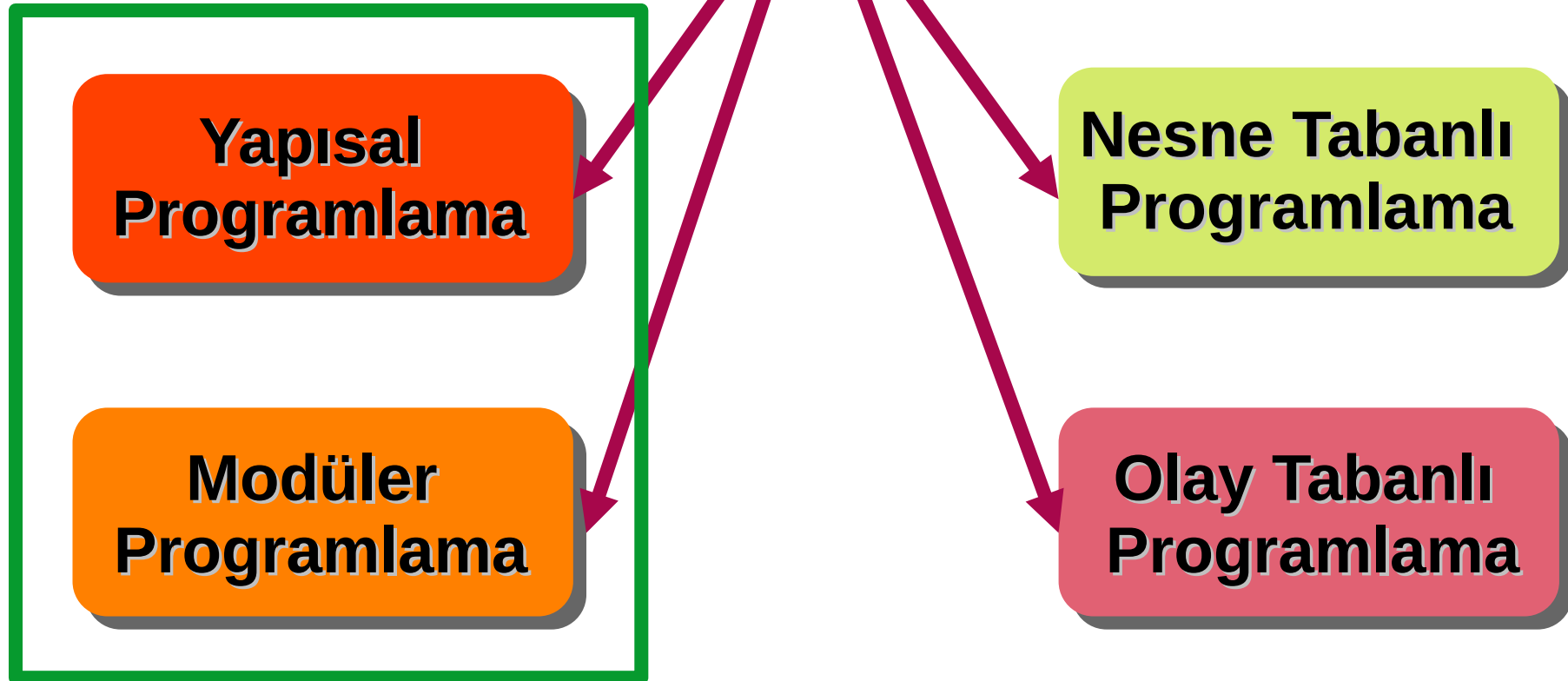
# PROGRAMLAMA TÜRLERİ

**Yapısal  
Programlama**

**Modüler  
Programlama**

**Nesne Tabanlı  
Programlama**

**Olay Tabanlı  
Programlama**



# BİR HATI ALATMA

- Ayrılırken;
  - + Masa üstlerinde LÜTFEN ÇÖP BIRAKMAYINIZ,
  - + Bilgisayarlarınızı LÜTFEN KAPATINIZ,
  - + Sandalyelerinizi LÜTFEN DÜZENLİ bırakınız,

**TEŞEKKÜRLER, GÖRÜŞMEK ÜZERE...**