

Nesneye Yönelimli Programlama – BLM 205

## Hafta 2: Python Temelleri I



Fenerbahçe Üniversitesi

# Öğretim Elemanları

Öğretim Üyesi: Dr. Vecdi Emre Levent

Ofis: 311

Email: [emre.levent@fbu.edu.tr](mailto:emre.levent@fbu.edu.tr)

Asistan: Arş. Gör. Uğur Özbalkan

Ofis: 307

Email: [ugur.ozbalkan@fbu.edu.tr](mailto:ugur.ozbalkan@fbu.edu.tr)

Asistan: Arş. Gör. Ecenur Alioğulları

Ofis: 307

Email: [ecenur.aliogullari@fbu.edu.tr](mailto:ecenur.aliogullari@fbu.edu.tr)

# Ders Planı

- Python Temelleri
  - Değişkenler ve Operatörler
  - Seçim ve Kontrol Yapıları
  - Fonksiyonlar
  - Diziler
  - Dosya İşlemleri

# Python Temelleri

- Python, düşük seviyeli dillere (Donanıma yakın diller C, C++ gibi) göre çok daha basit bir dildir.
- Birçok işlemi, düşük seviyedeki gibi uzun kod parçacıkları yerine, oldukça sadece ve kısa kod parçacıkları ile gerçekleştirmek mümkündür.
- Ders kapsamında Python 3 ile çalışmalar yapılacaktır.

# Python Temelleri

- En temel Python komutu "print" fonksiyonudur.

Örnek:

- `print("Deneme.")`

Örnek Kod Parçası

```
print("Deneme.")
```

Çıktı

Deneme.

- Print fonksiyonu, kendisi newline `\n` karakterini koyar.

# Python Temelleri

- Yeni bir satır bırakmadan print yapmak için, print fonksiyonuna 2. argüman olarak `end = " "` verilmelidir.

Örnek:

## Örnek Kod Parçasığı

```
print("deneme", end = " ")  
print("test")  
print("test2")
```

## Çıktı

```
Deneme test  
test2
```

# Python Temelleri

- Girintiler (indentation) bir çok programlama dilinde kodun okunabilirliği açısından önemlidir. Ancak Python dilinde girintiler, kodun okunabilirliğinin yanı sıra aynı zamanda kodu blok halinde kullanmaya yarar (Örneğin C'de ki süslü parantez { } kullanarak kodu blok haline getirmek gibi).

## Örnek Kod Parçasığı

```
if 1 > 0:  
    print("Deneme")  
else:  
    print("Deneme2")
```

## Çıktı

Deneme

# Python Temelleri

## Örnek Kod Parçasığı

```
if 1 > 0:  
    print("Deneme")  
else:  
    print("Deneme2")  
    print("Deneme3")  
  
print("Tamamlandi")
```

## Çıktı

Deneme  
Tamamlandi



# Python Temelleri

Girinti en az bir boşluk karakteri kadar olmalıdır.

Girintilerin sayısı fark etmemektedir.

## Örnek Kod Parçası

```
if 5 > 2:  
    print("Deneme")
```

```
if 5 > 2:
```

```
print("Deneme2")
```

## Çıktı

Deneme

Deneme2

# Python Temelleri

Aynı kod bloğu içerisindeki girinti sayısı aynı olmalıdır.

```
if 5 > 2:
```

```
    print("test!")
```

```
        print("test2!")
```

Syntax hatası verecektir.

# Python Temelleri

Kullanıcıdan giriş almak için,

```
username = input("Kullanıcı adı girin:")  
print("Kullanıcı adı: " + username)
```

Klavyeden girildi

```
Kullanıcı adı girin:test  
Kullanıcı adı: test  
Press any key to continue . . .
```

# Python Temelleri

Düşük seviyeli dillerde olduğu gibi Python'da da kod parçacıklarını açıklamak için yorum satırları eklenebilmektedir.

```
#Yorum satırı
```

```
print("Deneme")
```

```
print("Deneme") #Yorum satırı
```

```
"""
```

```
Çoklu
```

```
Yorum
```

```
Satırları
```

```
"""
```

```
print("Test")
```

# Python Temelleri

Değişken tanımlarken, düşük seviyeli dillerin aksine değişken tanımlamak için bir tanım yoktur.

Değişkenler, üzerlerine ilk atama yapıldıkları yerde tanımlanmış olurlar.

# Python Temelleri

## Örnek Kod Parçasığı

```
x = 5  
y = "Test"  
  
print(x)  
print(y)
```

## Çıktı

```
5  
Test
```

# Python Temelleri

Örnek:

X önce int olarak tanımlandı sonra string'e dönüşmektedir.

## Örnek Kod Parçasığı

```
x = 4 # x int olarak tanımlandı  
x = "Deneme" # x'in türü artık  
string  
  
print(x)
```

## Çıktı

Deneme

# Python Temelleri

Değişken isimleri verilirken,

- Bir harf veya \_ alt çizgi karakteri ile başlamalıdır
- Değişken ismi alfabetik, rakam ve altçizgi'den oluşabilir.
- Değişken isimleri büyük/küçük harf duyarlıdır.

Hatalı değişken ismi örnekleri:

```
2myvar = "deneme"  
my-var = "deneme"  
my var = "deneme"
```



# Python Temelleri

## Çoklu değişken ataması

### Örnek Kod Parçasığı

```
x, y, z = "aa", "bb", "cc"  
print(x)  
print(y)  
print(z)
```

### Çıktı

```
aa  
bb  
cc
```

# Python Temelleri

## Değişkenler ile birlikte print

### Örnek Kod Parçasığı

```
x = "test"  
print("Python " + x)
```

### Çıktı

Python test

# Python Temelleri

## Değişkenler ile birlikte print

### Örnek Kod Parçasığı

```
x = "test"  
y = 123  
print("Python " + x + " " + str(y))
```

### Çıktı

Python test 123

# Python Temelleri

Python'da arka planda bir çok veri tipi bulunur.

Bunlar;

Sayılar:	int, float, complex
Metinler:	str
Diziler:	list, tuple, range
Adresleme:	dict
Set Türleri:	set, frozenset
Mantıksal:	bool
İkilik Türler:	bytes, bytearray, memoryview

# Python Temelleri

## Sayılar

### Örnek Kod Parçasığı

```
x = 1      # int  
y = 2.8    # float  
z = 1j     # complex
```

```
print(type(x))  
print(type(y))  
print(type(z))
```

### Çıktı

```
<class 'int'>  
<class 'float'>  
<class 'complex'>
```

# Python Temelleri

## Sayılar

### Örnek Kod Parçasığı

```
x = 1      # int
y = 2.8    # float
z = 1j     # complex

t = x + y

print(t)
print(type(t))
```

### Çıktı

```
3.8
<class 'float'>
```

# Python Temelleri

## Sayılar

### Örnek Kod Parçasığı

```
x = 1      # int
y = 2.8    # float
z = 1j     # complex

#int'den float'a donusum
a = float(x)

#float'tan inte'e donusum
b = int(y)

#int'den complex'e donusum
c = complex(x)

print(a)
print(b)
print(c)

print(type(a))
print(type(b))
print(type(c))
```

### Çıktı

```
1.0
2
(1+0j)
<class 'float'>
<class 'int'>
<class 'complex'>
```

# Python Temelleri

## String'ler

### Örnek Kod Parçasığı

```
a = "Hello, World!"  
print(a[1])
```

```
b = "Hello, World!"  
print(b[2:5])
```

### Çıktı

```
e  
llo
```

2'den 5'e (5 dahil değil)



# Python Temelleri

## String'ler

### Örnek Kod Parçasığı

```
b = "Hello, World!"  
print(b[-5:-2])
```

### Çıktı

```
orl
```

-5'den -2'e (-2 dahil değil)

Sondan başlayarak -5. adrese gidilir. -2. adrese kadar olan karakterler alınır

# Python Temelleri

## String'ler

Bir string'in boyutu len fonksiyonu ile alınır

Örnek

Örnek Kod Parçası

```
a = "Hello, World!"  
print(len(a))
```

Çıktı

13

# Python Temelleri

## String'ler

Bir string'in başında ve sonundaki boşlukları silmek için strip fonksiyonu kullanılabilir

Örnek

### Örnek Kod Parçasığı

```
a = " Hello, World! "  
print(a.strip())
```

### Çıktı

```
Hello, World!
```

# Python Temelleri

## String'ler

Bir string'in tüm karakterlerini küçük yapmak için `lower`, büyük yapmak için `upper` fonksiyonu kullanılabilir.

### Örnek

#### Örnek Kod Parçasığı

```
a = "Hello, World!"  
print(a.lower())
```

```
a = "Hello, World!"  
print(a.upper())
```

#### Çıktı

```
hello, world!  
HELLO, WORLD!
```

# Python Temelleri

## String'ler

Bir string'in içerisindeki bulunan bir küçük parçayı, başka bir parça ile değiştirmek için replace fonksiyonu kullanılabilir.

### Örnek

#### Örnek Kod Parçasığı

```
a = "Hello, World!"  
x= a.replace("H", "J")  
print(x)
```

```
x= a.replace("Hello", "Test")  
print(x)
```

#### Çıktı

```
Jello, World!  
Test, World!
```

# Python Temelleri

## String'ler

Bir string içerisindeki bir alt string'e göre dizilere bölünmek istendiğinde split fonksiyonu kullanılabilir.

### Örnek

#### Örnek Kod Parçası

```
a = "Hello, World!"  
x= a.split(",");  
print(x[0] + " " + x[1])
```

#### Çıktı

```
Hello World!
```

# Python Temelleri

## String'ler

Bir string'in içerisinde bir alt metin aranmak istendiğinde "in" ile arama yapılabilir

### Örnek Kod Parçası

```
txt = "Deneme yazisi"  
x = "neme" in txt  
print(x)  
  
x = "asd" in txt  
print(x)
```

### Çıktı

```
True  
False
```

# Python Temelleri

## String'ler

Format metodu ile string'lerin içerisine dinamik olarak argüman yerleştirilebilir.

Örnek:

### Örnek Kod Parçasığı

```
txt = "Merhaba, benim yasim {}"  
  
yas1 = 22  
print(txt.format(yas1))  
  
yas2 = 23  
print(txt.format(yas2))
```

### Çıktı

```
Merhaba, benim yasim 22  
Merhaba, benim yasim 23
```



# Python Temelleri

## String'ler

Format metodu ile string'lerin içerisine dinamik olarak argüman yerleştirilebilir.

Örnek:

### Örnek Kod Parçasığı

```
adet = 5
numara = 1234
fiyat = 12.12

siparis = "{} nolu urunden {} adet, {}
TL toplam fiyat ile siparis alindi."

print(siparis.format(numara, adet,
fiyat))
```

### Çıktı

1234 nolu urunden 5 adet,  
12.12 TL toplam fiyat ile siparis  
alindi.

# Python Temelleri

## String'ler

Format metodu ile string'lerin içerisine dinamik olarak argüman yerleştirilebilir.

Örnek:

### Örnek Kod Parçacığı

```
adet = 5  
numara = 1234  
fiyat = 12.12  
  
siparis = "{2} nolu urunden {0} adet, {1}  
TL toplam fiyat ile siparis alindi."  
  
print(siparis.format(numara, adet, fiyat))
```

İstendiğinde argüman sırası  
değiştirilebilir

### Çıktı

12.12 nolu urunden 1234 adet,  
5 TL toplam fiyat ile siparis  
alindi.

# Python Temelleri

## String'ler

Metnin içerisinde özel karakterler kullanımı

<code>\"</code>	" karakteri
<code>\'</code>	' karakteri
<code>\\</code>	\ karakteri
<code>\n</code>	Yeni satır
<code>\t</code>	Tab
<code>\b</code>	Sağdaki bir karakteri siler
<code>\ooo</code>	Octal değer ( <code>\110 = H</code> )
<code>\xhh</code>	Hex değeri ( <code>\x65 = A</code> )

# Python Temelleri

## Operatörler

### Aritmetik Operasyonlar

Operatör	İsim	Örnek
+	Toplama	$x + y$
-	Çıkartma	$x - y$
*	Toplama	$x * y$
/	Bölme	$x / y$
%	Mod	$x \% y$
**	Üstsel	$x ** y$
//	Bölme ve en yakın küçük sayıya yuvarlama	$x // y$

# Python Temelleri

## Operatörler

## Atama Operatörleri

Operator	Örnek	Diğer Gösterim
=	x = 5	x = 5
+=	x += 3	x = x + 3
-=	x -= 3	x = x - 3
*=	x *= 3	x = x * 3
/=	x /= 3	x = x / 3
%=	x %= 3	x = x % 3
//=	x //= 3	x = x // 3
**=	x **= 3	x = x ** 3
&=	x &= 3	x = x & 3
=	x  = 3	x = x   3
^=	x ^= 3	x = x ^ 3
>>=	x >>= 3	x = x >> 3
<<=	x <<= 3	x = x << 3

# Python Temelleri

## Operatörler

### Karşılaştırma Operatörleri

Operatör	İsim	Örnek
==	Eşittir	<code>x == y</code>
!=	Eşit Değildir	<code>x != y</code>
>	Büyüktür	<code>x &gt; y</code>
<	Küçüktür	<code>x &lt; y</code>
>=	Büyük eşittir	<code>x &gt;= y</code>
<=	Küçük eşittir	<code>x &lt;= y</code>

# Python Temelleri

## Kontrol Yapıları

### If else yapıları ile kontrol

#### Örnek Kod Parçasığı

```
a = 44
b = 22
if b > a:
    print("b daha büyüktür")
elif a == b:
    print("esittir")
else:
    print("a daha büyüktür")
```

#### Çıktı

a daha büyüktür

# Python Temelleri

## Kontrol Yapıları

### İç içe kontrol yapıları

#### Örnek Kod Parçacığı

```
x = 41

if x > 10:
    print("deneme")
    if x > 20:
        print("test")
    else:
        print("test2.")
```

#### Çıktı

```
deneme
test
```



# Python Temelleri

## Kontrol Yapıları

İf kontrol mekanizmaları boş tanımlanamamaktadır. Eğer boş bir içerikte tanımlama ihtiyacı var ise, içerisine pass yazılmalıdır.

### Örnek Kod Parçasığı

```
x = 41

if x > 10:
    print("deneme")
    if x > 20:
        pass
    else:
        print("test2.")
```

### Çıktı

deneme

# Python Temelleri

## Döngüler

### While

#### Örnek Kod Parçasığı

```
x = 1
while x < 6:
    print(x)
    x += 1
```

#### Çıktı

```
1
2
3
4
5
```

# Python Temelleri

## Döngüler

### While-break

#### Örnek Kod Parçasığı

```
x = 1
while x < 6:
    print(x)
    if x == 3:
        break
    x += 1
```

#### Çıktı

```
1
2
3
```

# Python Temelleri

## Döngüler

### While-continue

#### Örnek Kod Parçasığı

```
x = 0
while x < 6:
    x += 1
    if x == 3:
        continue
    print(x)
```

#### Çıktı

```
1
2
4
5
6
```

# Python Temelleri

## Döngüler

### For

#### Örnek Kod Parçasığı

```
for x in range(6):  
    print(x)
```

#### Çıktı

```
0  
1  
2  
3  
4  
5
```

# Python Temelleri

## Döngüler

### For

#### Örnek Kod Parçasığı

```
for x in range(2, 6):  
    print(x)
```

#### Çıktı

```
2  
3  
4  
5
```

# Python Temelleri

## Döngüler

### For

#### Örnek Kod Parçasığı

```
for x in range(2, 30, 3):  
    print(x)
```

#### Çıktı

```
2  
5  
8  
11  
14  
17  
20  
23  
26  
29
```

# Python Temelleri

## Fonksiyonlar

Fonksiyon tanımı:

### Örnek Kod Parçasığı

```
def testFonksiyonu():  
    print("Test")  
  
testFonksiyonu()
```

### Çıktı

Test



# Python Temelleri

## Fonksiyonlar

### Fonksiyon Arguman:

#### Örnek Kod Parçasığı

```
def testFonksiyonu(arguman):  
    print(arguman + " XYZ")
```

```
testFonksiyonu("AAA")  
testFonksiyonu("BBB")  
testFonksiyonu("CCC")
```

#### Çıktı

```
AAA XYZ  
BBB XYZ  
CCC XYZ
```

# Python Temelleri

## Fonksiyonlar

### Fonksiyon Çoklu Arguman:

#### Örnek Kod Parçasığı

```
def testFonksiyonu(arg1, arg2):  
    print(arg1 + " " + arg2)  
  
testFonksiyonu("XYZ", "ABC")
```

#### Çıktı

XYZ ABC

# Python Temelleri

## Fonksiyonlar

### Fonksiyon Sınırsız Arguman:

#### Örnek Kod Parçasığı

```
def testFonksiyonu(*args):  
    print("Arguman Testi " + args[2])  
  
testFonksiyonu("XYZ", "ABC", "CBA")
```

#### Çıktı

Arguman Testi CBA

# Python Temelleri

## Fonksiyonlar

### Fonksiyon Keyword Argümanlar:

#### Örnek Kod Parçasığı

```
def testFonksiyonu(arg33, arg12, arg24):  
    print("Deneme metni " + arg33)
```

```
testFonksiyonu(arg24 = "AAA", arg33 =  
"BBB", arg12 = "CCC")
```

#### Çıktı

Deneme metni BBB

# Python Temelleri

## Fonksiyonlar

### Fonksiyon Sınırsız Keyword Argümanlar:

#### Örnek Kod Parçasığı

```
def testFonksiyonu(**argumanlar):  
    print("Deneme metni " +  
          argumanlar["giris2"])  
  
testFonksiyonu(giris1 = "ABC",  
               giris2 = "XYZ")
```

#### Çıktı

Deneme metni XYZ

# Python Temelleri

## Fonksiyonlar

### Fonksiyon Default Değerler:

#### Örnek Kod Parçasığı

```
def testFonksiyonu(country =  
"Turkiye"):  
    print("Ulke: " + country)  
  
testFonksiyonu("Amerika")  
testFonksiyonu("Almanya")  
testFonksiyonu()  
testFonksiyonu("Ingiltere")
```

#### Çıktı

```
Ulke: Amerika  
Ulke: Almanya  
Ulke: Turkiye  
Ulke: Ingiltere
```

# Python Temelleri

## Fonksiyonlar

### Fonksiyon Geri Dönüş Değeri:

#### Örnek Kod Parçasığı

```
def testFonksiyonu(x):  
    return 10 * x  
  
print(testFonksiyonu(3))  
print(testFonksiyonu(5))  
print(testFonksiyonu(9))
```

#### Çıktı

30  
50  
90

# Python Temelleri

## Fonksiyonlar

### Fonksiyon Özyineleme (Recursion):

#### Örnek Kod Parçasığı

```
def recur_fibo(n):  
    if n <= 1:  
        return n  
    else:  
        return(recur_fibo(n-1) +  
recur_fibo(n-2))  
  
rtnVal = recur_fibo(6)  
  
print(rtnVal)
```

#### Çıktı

8



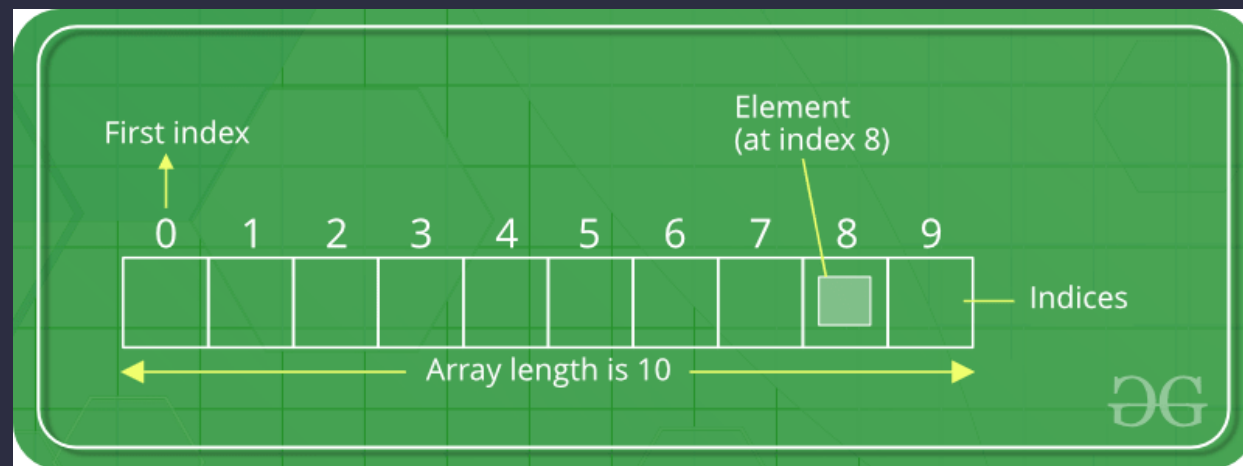
# Python Temelleri

## Diziler

Aynı türden elemanları grup halinde saklamak için kullanılır.

Syntax'ı:

`array(Veri Türü, Değerler)`



# Python Temelleri

## Diziler

### Örnek Kod Parçasığı

```
import array as arr

a = arr.array('i', [1, 2, 3]) # integer türü ile

for i in range (0, 3):
    print (a[i], end = " ")

print()

b = arr.array('d', [2.5, 3.2, 3.3]) # float türü ile

for i in range (0, 3):
    print (b[i], end = " ")

print()
```

### Çıktı

```
1 2 3
2.5 3.2 3.3
```

# Python Temelleri

## Dosya İşlemleri

Dosyalar ile okuma, yazma işlemlerinin yapılabilmesi için dosyanın açılması gerekmektedir.

Open fonksiyonu ile dosya açılır.

Syntax:

```
f = open("test.txt")
```

# Python Temelleri

## Dosya İşlemleri

Dosyalar ile okuma, yazma işlemlerinin yapılabilmesi için dosyanın açılması gerekmektedir.

Open fonksiyonu ile dosya açılır.

Syntax:

```
f = open("test.txt")
```

# Python Temelleri

## Dosya İşlemleri

Syntax:

```
f = open("test.txt")
```

```
f = open("demofile.txt", "r")
```

Dosya Modu	Açıklama
r	Read – Okuma
a	Append – Ekleme
w	Write – Yazma
x	Oluşturma

# Python Temelleri

## Dosya İşlemleri

### Örnek Kod Parçasığı

```
f = open("test.txt", "r")  
x = f.read()  
print(x)
```

### test.txt

```
Deneme yazisi  
1234  
Merhaba
```

### Çıktı

```
Deneme yazisi  
1234  
Merhaba
```

# Python Temelleri

## Dosya İşlemleri

### İlk 17 karakterin okunması

#### Örnek Kod Parçasığı

```
f = open("test.txt", "r")  
print(f.read(17))
```

#### Çıktı

```
Deneme yazisi  
123
```

#### test.txt

```
Deneme yazisi  
1234  
Merhaba
```

# Python Temelleri

## Dosya İşlemleri

### Satır satır okuma

#### Örnek Kod Parçasığı

```
f = open("test.txt", "r")  
for x in f:  
    print(x)  
f.close()
```

Dosya  
kapatılması

#### test.txt

```
Deneme yazisi  
1234  
Merhaba
```

#### Çıktı

```
Deneme yazisi  
  
1234  
  
Merhaba
```



# Python Temelleri

## Dosya İşlemleri

### Dosyaya ekleme:

#### Örnek Kod Parçasığı

```
f = open("test.txt", "a")  
f.write("Yeni metin!")  
f.close()
```

#### test.txt

```
Deneme yazisi  
1234  
Merhaba
```

#### test.txt

```
Deneme yazisi  
1234  
Merhaba  
Yeni metin!
```

Yeni test.txt dosyası

# Python Temelleri

## Dosya İşlemleri

Dosyaya yazma (Önceden dosya var ise, içeriği sıfırlanır):

### Örnek Kod Parçası

```
f = open("test.txt", "w")  
f.write("Deneme icerik!")  
f.close()
```

test.txt

Deneme icerik!