

Hafta1_Konu

February 22, 2021

1 CENG 220 PROGRAMLAMA DİLLERİ LABORATUVARI

Instructor: Dr. İbrahim KÖK

Şube 1/2 (Salı/Çarşamba)

Değerlendirme Sistemi: Ara sınav %40 - Final Sınavı %60

- Python -> Course Length: 3 weeks
- Ruby -> Course Length: 3 weeks
- Haskell -> Course Length: 3 weeks
- C++ -> Course Length: 3 weeks

Uygulama Ortamı <https://bilmoodle.pau.edu.tr/>

1.1 Python

2 1. Hafta İçeriği

- Giriş
- Python IDE ve platformları
- Online IDE'ler
- Python Etkileşimli Kabuk
- Python Sürümleri
- Print() komutu
- Değişkenler ve Atama İşlemleri
- İşleçler
- Karar ve Kontrol Yapıları (If / For / While)
- Haftanın Deneyleri (3 uygulama)

Giriş Python is a general-purpose interpreted, interactive, object-oriented, and high-level programming language. It was created by Guido van Rossum during 1985- 1990.

!! Adını sanılanın aksine bir YILAN'dan DEĞİL, Guido van Rossum'un çok sevdiği, Monty Python adlı altı kişilik bir İngiliz komedi grubunun Monty Python's Flying Circus adlı gösterisinden almıştır.

- Python, öğrenmesi kolay, tamamen özgür ve ücretsiz bir programlama dilidir.

- Nesnelere dayalı bir dil olup okunabilirliği yüksektir. Python'un dili başka programlama dilleri ile kıyaslandığında, bunun daha az kod ile işlemleri yapmasının mümkün olduğu görülecektir.
- Python, bütün işletim sistemleri ile uyum içerisinde çalışmaktadır.
- Programlama yapısı içerisinde birçok kütüphaneyi barındırmaktadır. Bu kaynaklarla daha az kod yazmak mümkündür.
- Python ile masaüstünde çalışan uygulamalar geliştirilebileceği gibi, web üzerinde çalışan uygulamalar geliştirmek hatta Raspberry-Pi gibi donanımları da programlamak mümkündür.
- Python Yorumlanabilir - Python, yorumlayıcı tarafından çalışma zamanında işlenir. Programınızı çalıştırmadan önce derlemenize gerek yoktur. Bu, PERL ve PHP'ye benzer.
- Python Etkileşimlidir -Python komut istemine oturabilir ve programlarınızı yazmak için doğrudan yorumlayıcıyla etkileşime girebilirsiniz.
- Python Nesne Yönelimlidir - Python, Nesne Yönelimli programlama tekniğini destekler.
- Python Başlangıç Dilidir - Python, başlangıç seviyesindeki programcılar için harika bir dildir

Python IDE ve platformları

- Spyder (Anaconda içinde mevcut)
- Jupiter Notebook
- PyCharm
- Thonny, Sublime Text, Atom

Free online IDEs

- https://www.onlinegdb.com/online_python_compiler
- https://www.w3schools.com/python/python_compiler.asp
- <https://www.jdoodle.com/python3-programming-online/>
- <https://repl.it/languages/python3>

In []:

Python Etkileşimli Kabuk Python kurulumu sonrası- Kabuk, Terminal ekranında python komutu verilerek açılır

- Kurulu python versiyon kontrolü için >>> python -version
- Çıkış için >>> exit()

```
In [33]: from IPython.display import Image
         Image(filename="/Users/ibrahim/Programlama Dilleri Laboratuvarı/images/kabuk.png")
```

Out[33]:

```
ibrahim — python — 86×10
jupyter_mac.command — python -bash ~ — python
IbrahimMB:~ ibrahim$ python
Python 3.6.8 |Anaconda custom (x86_64)| (default, Dec 29 2018, 19:04:46)
[GCC 4.2.1 Compatible Clang 4.0.1 (tags/RELEASE_401/final)] on darwin
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

Python Sürümleri Python programlama dilinin Python 2.x ve Python 3.x olmak üzere bilinen iki büyük sürümü mevcuttur. Bazı durumlarda Python 2'nin kullanılması avantajlı olabileceken yeni başlayan birinin Python 3 öğrenmesi daha iyi olacaktır. Python 2 ve 3 arasındaki belli başlı farklar Tablo 1'de görülebilir.

Sürümler arasındaki farklılıklar, özellikle ileride görülecek olan fonksiyonların, print komutu gibi bazı komutların farklı olarak ifade edilmesini kapsamaktadır.

In [34]: `Image(filename="/Users/ibrahim/Programlama Dilleri Laboratuvarı/images/sur`

Out [34]:

Tablo 1: Python 2-3 Karşılaştırması

	Python 2	Python 3
Kullanım	Eskiden kalan: Hala belirli firmalarda kullanılmaktadır	Gelecek: 2020 yılında Python 2'nin yerini tamamen alacak
Kütüphane	Python 2 için geliştirilen bir çok eski kütüphane ileriye yönelik uyumlu değildir	Günümüzdeki geliştiricilerin bir çoğu kütüphaneleri özellikle Python 3 ile kullanılacak şekilde geliştirmektedir.
Kodlama	Metinler varsayılan olarak ASCII biçiminde kodlanır.	Metinler varsayılan olarak Unicode biçiminde saklanır.
Bölme	Bölme işlemi sonucu aşağı yuvarlanır. ($5/2=2$)	Bölme işlemi sonucu yuvarlama uygulanmaz. ($5/2=2.5$)
Yazdırma	<code>print "hello"</code>	<code>print("hello")</code>

In [1]: `# Tek satırlık açıklama ve yorum satırları için kare kullanılır`

In []: `"""Çok satırlı yorum ve
açıklamalar için üç tırnak
"""`

Yazdırma Komutu - print()

- print ("String")
- print ("Hello", degiskenismi)
- print ("Hello" + degiskenismi)
- print ("Hello %s" % degiskenismi)
 - %s -> string
 - %d -> Integer
 - %f -> Float
 - %o -> Octal
 - %x -> Hexadecimal
 - %e -> exponential
 - %r -> boolean
- print with format()
print("{} {} {}".format(a,b,c))

```
In [2]: print ("Hello Pau !")
```

```
Hello Pau !
```

```
In [3]: dersgunu="23 Subat Salı"  
        print ("1. hafta",dersgunu)
```

```
1. hafta 23 Subat Salı
```

```
In [4]: dersgunu="23 Subat Salı"  
        print ("1. hafta" + dersgunu)
```

```
1. hafta23 Subat Salı
```

```
In [5]: x=10  
        print ("x sayısının değeri %d"%x)
```

```
x sayısının değeri 10
```

```
In [6]: metin="Bilgisayar Mühendisliği"  
        print ("Bölüm Adı: %s" %metin)
```

```
Bölüm Adı: Bilgisayar Mühendisliği
```

```
In [7]: print ("Actual Number = %d" %46)
        print ("Float of the number = %f" %27)
        print ("Float of the number = %.2f" %46)
        print ("Octal equivalent of the number = %o"%78 )
        print ("Hexadecimal equivalent of the number = %x" %46)
```

```
Actual Number = 46
Float of the number = 27.000000
Float of the number = 46.00
Octal equivalent of the number = 116
Hexadecimal equivalent of the number = 2e
```

```
In [8]: a=5 #int
        b=6.7 # float
        c="metin" # string

        print ("{} {} {}".format(a,b,c))
```

```
5 6.7 metin
```

```
In [9]: # istenilen sırada yazılabilir
        print ("{2} {1} {0}".format(a,b,c))
```

```
metin 6.7 5
```

```
In [10]: # Süslü parantezlerin içindeki kullanım ondalıklı kısmın sadece 2 basamağ
        print (" {:.2f} {:.2f} {:.3f}".format(3.1463,5.324,7.324324))
```

```
3.15 5.32 7.324
```

```
In [ ]:
```

2.0.1 Değişkenler ve Atama İşlemleri

Değişkenler, değerleri korumak için kullanılır. Bu değerler sayı, dizi gibi farklı biçimlerde olabilir.

- Python, büyük küçük harf duyarlıdır ve değişken isimleri için kesin kurallar kullanır.
- Bir değişken ismi belirteç için bir örnektir.
- Belirteç, öğeleri isimlendirmek için kullanılan kelimedir.
- Belirteçler fonksiyon, sınıf ve metod gibi parçaları isimlendirmek için de kullanılır.

Belirteç belirlerken dikkat edilmesi gereken kurallar

1. Bir belirteç en az bir karakter içermelidir.
2. Belirtecın ilk karakteri harf (küçük ya da büyük) ya da alt çizgi olmalıdır - (ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz_).

3. Devam eden karakterler, harf (küçük ya da büyük), alt çizgi ya da SAYI olabilir - (ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz_0123456789).
4. Belirteçlerde boşluk dâhil diğer özel karakterler KULLANILMAZ
5. Programlama diline ait ayrılmış kelimeler belirteç olarak kullanılamaz (Reserved Words).
6. Türkçe karakterler kullanılamaz.

- Atama işlemi bir değeri bir değişken ile eşleştirir. Bu ifadedeki en önemli ayrıntı, atama (=) sembolüdür.

In [35]: Image(filename="/Users/ibrahim/Programlama Dilleri Laboratuvarı/images/ata")

Out [35]:

Python için Ayrılmış Kelimeler			Python için Doğru Belirteçler	Python için Yanlış Belirteçler
and	del	from		<input type="checkbox"/> ara-toplam (- sembolü kullanılamaz)
as	elif	global	<input checked="" type="checkbox"/> x	<input type="checkbox"/> ilk değer (boşluk kullanılamaz)
assert	else	if	<input checked="" type="checkbox"/> a2	<input type="checkbox"/> 4ogrenci (sayı ile başlayamaz)
break	except	import	<input checked="" type="checkbox"/> Toplam	<input type="checkbox"/> *2 (* sembolü kullanılamaz)
class	False	in	<input checked="" type="checkbox"/> Toplam_Brut	<input type="checkbox"/> öğrenci (Türkçe karakter içeremez)
continue	finally	is	<input checked="" type="checkbox"/> Anahtar_10	<input type="checkbox"/> class (class ayrılmış kelime olduğu için kullanılamaz)
def	for	lambda		

Python'da çeşitli veri türü bulunmaktadır. Değişkenlere bu veri türünden biri atanır. Yorumlayıcı, bir değişkenin veri türüne bağlı olarak bellek ayırır ve ayrılmış bellekte nelerin saklanabileceğine karar verir.

Veri Türleri:

- Text Type: str
- Numeric Types: int, float
- Sequence Types: list, tuple, range
- Mapping Type: dict
- Set Types: set, frozenset
- Boolean Type: bool

Değişkenin veri türünü öğrenmek için type() komutu kullanılır

Örnek değişken atamaları (değişken atama işlemi → değişken veri türü)

- x = "Hello World" → str
- x1 = 20 → int
- x2 = 20.5 → float

- x3 = ["apple", "banana", "cherry"] -> list
- x4 = ("apple", "banana", "cherry") -> tuple
- x5 = range(6) -> range 0 1 2 3 4 5
- x6 = {"name": "John", "age": 36} -> dict
- x7 = {"apple", "banana", "cherry"} -> set
- x8 = frozenset({"apple", "banana", "cherry"}) -> frozenset
- x9 = True -> bool

```
In [11]: #Değişken tanımlama ve değer atama
        sayi_1=15                # integer atama (0 -52 7)
        sayi_2=10.0             # floating point atama (1.2345, -1.23 0.0 13.0 -
        sayi_3="ibrahim"        # string atama
        sayi_4 = '''Çok satırlı string
        değişken atama
        '''                     # çok satırlı string atama
        sayi_5=True              # Boolean True False
```

```
In [12]: print(sayi_1)
        print(type(sayi_1))
```

```
15
<class 'int'>
```

```
In [13]: print(sayi_2)
        print(type(sayi_2))
```

```
10.0
<class 'float'>
```

```
In [14]: print(sayi_3)
        print(type(sayi_3))
```

```
ibrahim
<class 'str'>
```

```
In [15]: print(sayi_4)
        print(type(sayi_4))
```

```
Çok satırlı string
değişken atama
```

```
<class 'str'>
```

```
In [16]: print(sayi_5)
         print(type(sayi_5))
```

```
True
<class 'bool'>
```

İşleçler Python aritmetik işlem notasyonu olarak infix notasyonunu(islem ortada) kullanır. Tablo’da Python dilinde kullanılan işleçler verilmiştir.

```
In [36]: Image(filename="/Users/ibrahim/Programlama Dilleri Laboratuvarı/images/islecler.png")
```

```
Out [36]:
```

Tablo 2: Python dilinde kullanılan işleçler

Tür	İşleçler	Örnek
Aritmetik	+, -, *, /, %, **, //	x+y
Karşılaştırma	==, !=, <>, >, <, >=, <=	a>b
Atama	=, +=, -=, *=, /=, %=, **=, //=	x += 2
Bit tabanlı	&, , ^, ~, <<, >>	a&b
Mantıksal	and, or, not	True and False
Üyelik	in, not in	x in liste
Kimlik	is, is not	a is None

İşleç Örnek Uygulamaları

```
In [37]: Image(filename="/Users/ibrahim/Programlama Dilleri Laboratuvarı/images/ari.png")
```

```
Out [37]:
```


İfade	Anlamı
$x+y$	Eğer x ve y rakamsa, x, y'ye eklenir. Eğer x ve y harf dizisi ise x ile y birleştirilir.
$x-y$	Eğer x ve y rakamsa, x'ten y çıkarılır.
$x*y$	Eğer x ve y rakamsa, x ile y çarpılır. Eğer x harf dizisi ise ve y rakamsa, x, y kez sıralanır. Eğer x rakamsa ve y harf dizisi ise y, x kez sıralanır.
x/y	Eğer x ve y rakamsa, x, y'ye bölünür.
$x//y$	Eğer x ve y rakamsa, x'in içinde y değişkenin katsayısını arar. Örn. $10 // 4 = 2$
$x\%y$	Eğer x ve y rakamsa, bölme işleminde x / y kalan kısmını verir.
$x**y$	Eğer x ve y rakamsa, x, y'nin kuvvetiyle çarpılır.

```
In [17]: x1=5
          y1=2
          s1="Pau"
          s2="Ceng"
```

Aritmetik işlemler

```
In [18]: x1+y1
```

```
Out[18]: 7
```

```
In [19]: s1+s2
```

```
Out[19]: 'PauCeng'
```

```
In [20]: x1-y1
```

```
Out[20]: 3
```

```
In [21]: 5*5
```

```
Out[21]: 25
```

```

In [22]: s1*5
Out[22]: 'PauPauPauPauPau'

In [24]: 5*s2
Out[24]: 'CengCengCengCengCeng'

In [26]: 27%5 # modüler bölme işlemi - Mod operatörü (%) sayı bölümünden kalanı h
Out[26]: 2

In [28]: 5*5*5*5
Out[28]: 625

In [29]: 5**4 # üs alma
Out[29]: 625

In [30]: 23/4 # bölme
Out[30]: 5.75

In [31]: 23//4 # floor division
Out[31]: 5

```

Karşılaştırma

```

In [ ]: x1=5
        y1=2
        s1="Pau"
        s2="Ceng"
        s3="Ceng"

In [ ]: print(x1==y1)

In [ ]: print(x1!=y1)

In [ ]: print(s1==s2)

In [ ]: print(s2==s2)

In [ ]: x1<y1 #x1=5 y1=2

In [ ]: sayi1=23
        sayi2=5

        sayi1%=sayi2 # sayi1 = sayi1 % sayi2

        print(sayi1)

```

2.0.2 Input Fonksiyonu

Kullanıcıdan girdi/bir değer almak için kullanılır. String değer döndürür. , - Alınan string değeri int, float fonksiyonlarıyla istenilen türe dönüştürülebilir.

```
In [ ]: girdi = input("")
        print(girdi)

In [ ]: girdi = input("Enter your input: ")
        print(girdi)
        print(type(girdi))

In [ ]: deger = int(input('Bir sayı girin: '))
        print(deger)
        print(type(deger))
```

2.0.3 Karar ve Kontrol Yapıları

Karar ve kontrol yapıları kodun akışını değiştirmemizi sağlayan yapılardır ve genellikle bir blok oluşturmamızı sağlarlar. -Python dilinde bloklar diğer dillerde olduğu gibi süslü parantezlerle değil girintilerle tanımlanırlar. Girinti miktarı 4 boşluk karakteri kadardır.

If Yapısı Verilen koşulun doğruluğuna göre işlem veya işlemler gerçekleştirmemizi sağlayan karar yapısıdır. - Else if yapısı için elif anahtar kelimesi kullanılır. - if, elif ve else yapılarından sonra blok tanımlamak için : sembolü kullanılır.

Üç farklı şekilde yazılabilir.

```
In [38]: Image(filename="/Users/ibrahim/Programlama Dilleri Laboratuvarı/images/ifs
```

Out[38]:

If	If-else	If-elif
<pre>if some_condition : algorithm/expression</pre>	<pre>if some_condition : algorithm/expression else : algorithm/expression</pre>	<pre>if some_condition : algorithm/expression elif some_condition: algorithm/expression else : algorithm/expression</pre>

```
In [ ]: x = 46
        if x > 40:
            print ("40 den büyüğüm")
```

```
In [ ]: x = 11
        if x > 10: # koşul/şart
            print ("Deger 10 dan büyük") # Eğer koşul doğruysa -True
        else:
            print ("Deger 10 dan küçük") # iEğer koşul yanlışsa-False
```

```
In [ ]: x = 10
        y = 30
        if x > y:
            print ("x sayısı y sayısından büyüktür")
        elif x < y:
            print ("x sayısı y sayısından küçüktür")
        else:
            print ("iki sayı eşittir")
```

```
In [ ]:
```

For Döngüsü Python dilindeki for döngüsü C ve türevi dillerden biraz farklıdır. - C'deki for döngüsünün kısımlarının aksine Python dilindeki for döngüsü ardışık değerler içeren veri yapılarının(liste, string vd.) elemanları üzerinde tekrarlı işlemler yapmak için kullanılır. - for döngüsü genellikle aralık belirten range fonksiyonu ile birlikte kullanılır. Asağıda bu kullanıma örnek asal sayı hesaplayan bir kod bulunmaktadır. - range fonksiyonu üst sınırda belirtilen değeri dahil etmez. - Örneğin; range(4) ifadesi 0,1,2,3 değerlerini içerir.

```
In [ ]: for i in range(4):
        print(i)
```

```
In [ ]: for i in range(2,3):
        print(i)
```

```
In [ ]: for i in range(2,10,3):
        print(i)
```

```
In [ ]: metin = input('Bir metin girin: ')
        for harf in metin:
            print(harf)
```

While Döngüsü while döngüsü ilgili koşul doğru olduğu sürece sürekli olarak çalışacaktır. - while döngüsünde de blok oluşturmak için : sembolü ve girintiler kullanılır

```
In [ ]: sayac = 1 # Başlangıç değeri kontrol değişkenine atılır
        while sayac <= 5: #İstenilen değere ulaşp ulaşmadığını kontrol eder
            print("sayac değeri :",sayac) # Sayaç değerini ekrana yazar.
            sayac+= 1 # Sayaç değerini 1 arttırır.
        print(sayac)
```

```
In [ ]: sayi = int(input('Toplanacak sayıları girin(-1 çıkış): '))
        toplam=0
        while sayi != -1: # -1 e eşit olmadığı her durum için çalışır.
            toplam+=sayi
            sayi = int(input('Sayıyı girin: '))
```

```

        toplam += sayi
        sayi = int(input('Toplanacak sayıları girin(-1 çıkış): '))
print('Toplam: '+str(toplam))

```

```

In [ ]: # negatif sayı girişi olana kadar girilen sayıları toplamı
        giris = 0
        toplam=0
        print("Lütfen bir sayı giriniz, negatif sayılar döngüyü sonlandırır:")
        while True:
            giris = int(input())
            if giris < 0:
                break # Döngüden çıkılıyor
            toplam += giris
        print("Toplam =", toplam)

```

```

In [ ]: # Girilen 3 sayının ortalamasını alan program
        # Negatif sayı girildiğinde program sonlandırılır
        sayac = toplam = 0
        print("Lütfen Ortalama hesaplamak için 3 pozitif sayı giriniz")
        while sayac < 3:
            sayi = float(input("Sayı giriniz: "))
            if sayi < 0:
                print("Negatif sayılar kabul edilmemektedir.")
                #break
                continue
            sayac += 1
            toplam += sayi
        else:
            print("Ortalama =", toplam/sayac)

```

Haftanın Deneyleri

1. Tas-kagıt-makas oyunu:

Bu uygulamada iki kişi ile oynanan klasik tas, kagıt, makas oyununu gerçekleştirmek

KAZANMA

- Taş, makası kırarak yener.
- Kağıt, taşı sararak yener.
- Makas, kağıdı keserek yener.

```

In [39]: Image(filename="/Users/ibrahim/Programlama Dilleri Laboratuvarı/images/tes

```

```

Out[39]:

```

Execution files

vpl_evaluate.cases

```
1 ▾ case = Test 1
2   input=taş
3   taş
4   output=berabere
5 ▾ case = Test 2
6   input=taş
7   kağıt
8   output=ikinci oyuncu kazandı
9 ▾ case = Test 3
10  input=taş
11  makas
12  output=birinci oyuncu kazandı
13 ▾ case = Test 4
14  input=kağıt
15  taş
16  output=birinci oyuncu kazandı
17 ▾ case = Test 5
18  input=kağıt
19  kağıt
20  output=berabere
21 ▾ case = Test 6
22  input=kağıt
23  makas
24  output=ikinci oyuncu kazandı
25 ▾ case = Test 7
26  input=makas
27  taş
28  output=ikinci oyuncu kazandı
29 ▾ case = Test 8
30  input=makas
31  kağıt
32  output=birinci oyuncu kazandı
33 ▾ case = Test 9
34  input=makas
35  makas
36  output=berabere
37
```

2. Mükemmel sayı

Tam bölenlerinin toplamı kendisine esit olan sayılara mükemmel sayı denir. 28 ve 96

3. Collatz sanısı uygulamaları

Henüz doğruluğu kanıtlanmamış olan Collatz sanısı yapılan iki işlem sonunda 1'e ino

In [40]: Image(filename="/Users/ibrahim/Programlama Dilleri Laboratuvarı/images/col

Out[40]:

$$f(n) = \begin{cases} \frac{n}{2}, & n \text{ çift ise} \\ 3n + 1, & n \text{ tek ise} \end{cases}$$
$$a_i = \begin{cases} n, & i=0 \\ f(a_{i-1}), & i>0 \end{cases}$$

Collatz sanısında bir seri sayının kendisi ile baslar ve 1 ile sonlanır.

Örneğin 13 sayısı için Collatz serisi 13, 40, 20, 10, 5, 16, 8, 4, 2,1 sayılarından

Kaynaklar [1] PEP 8 – Style Guide for Python Code. Url: <https://www.python.org/dev/peps/pep-0008/#code-lay-out> dev/peps/pep-0008/ (son erişim: 19.2.2021).

[2] Python web sitesi. url: <https://www.python.org> (son erişim: 19.2.2021).

[3] The Python Tutorial. url: <https://docs.python.org/3/tutorial/index.html> (son erişim: 18.2.2019).

[4] <https://www.tutorialspoint.com/python/index.htm>

In []: