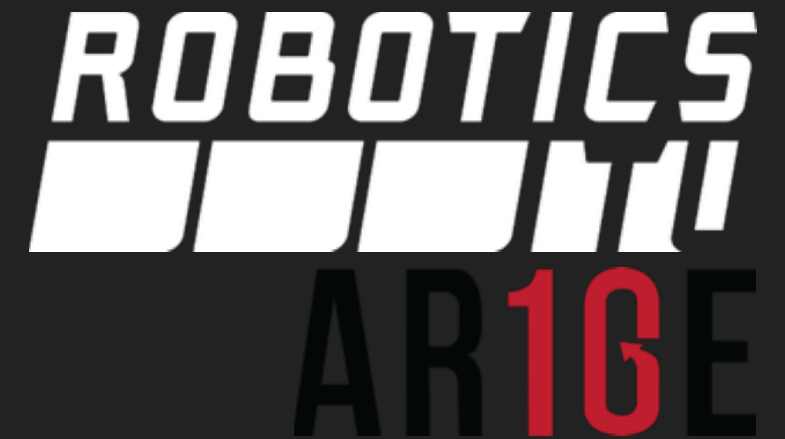


İTÜ ROBOTİK KULÜBÜ (ARIGE) EĞİTİMİ

3. HAFTA

PYTHON İLE PROGRAMLAMAYA GİRİŞ

GUIDO VAN ROSSUM (31 JANUARY 1956)



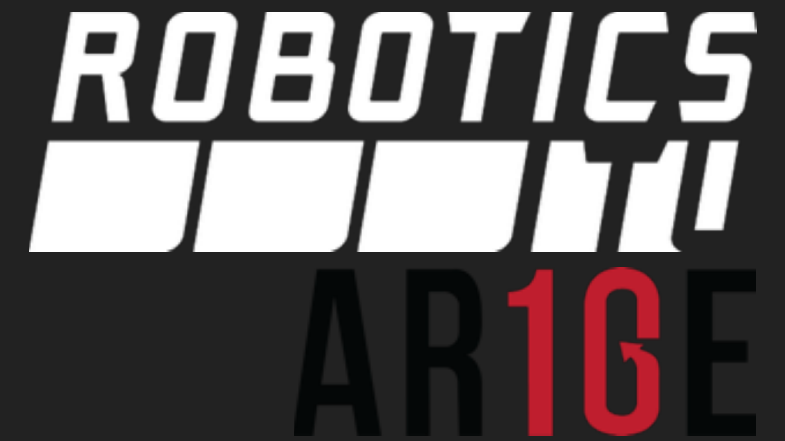
"Over six years ago, in December 1989, I was looking for a "hobby" programming project that would keep me occupied during the week around Christmas. My office ... would be closed, but I had a home computer, and not much else on my hands. I decided to write an interpreter for the new scripting language I had been thinking about lately: a descendant of ABC that would appeal to Unix/C hackers. I chose Python as a working title for the project, being in a slightly irreverent mood (and a big fan of Monty Python's Flying Circus)."



▶ DropBox

▶ **IMPORT THIS**

PYTHON NEDİR? NEDEN PYTHON?

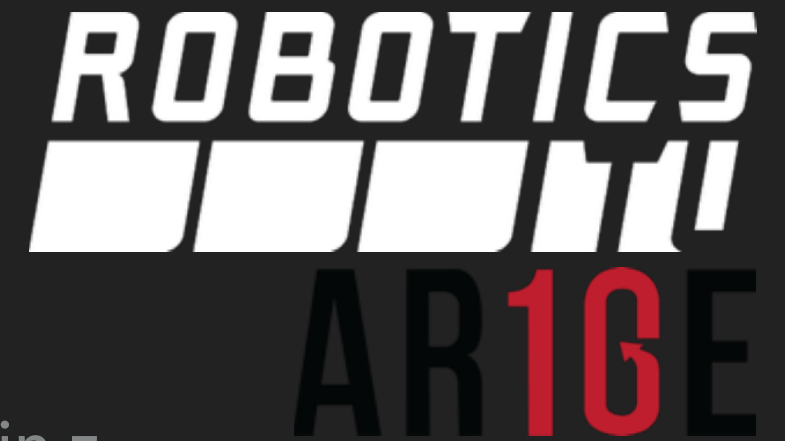


- ▶ Python is a widely used high-level, general-purpose, interpreted, dynamic programming language. Its design philosophy emphasizes code readability, and its syntax allows programmers to express concepts in fewer lines of code than possible in languages such as C++ or Java. The language provides constructs intended to enable writing clear programs on both a small and large scale.
- ▶ Yukarda geçen *High-level, General-purpose, Interpreted, Dynamic* kavramları ne demektir.
- ▶ Numpy, PyQt, Django, PyGame...

PYTHON NASIL TELAFFUZ EDİLİR?



GEÇTİĞİMİZ HAFTADAN ÖRNEKLER



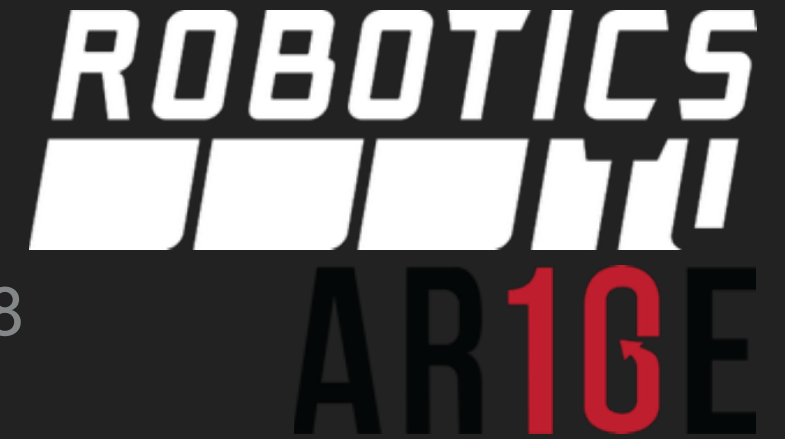
- ▶ Daire alan bulma fonksiyonu yaz.
- ▶ "Hoşgeldin x, en sevdiğin meyvenin y, en sevdiğin rengin z olduğunu biliyorum." çıktısı.
- ▶ $f(x) = ax^2 + bx - c$ şeklindeki fonksiyonun farklı değerler için köklerini hesapla.



- ▶ Şekildeki beyaz alanın 9, 7.74, 7.73 ve 7.72 cm² olduğu durumlar için pi sayısını hesaplayın. (kare çevresi 48cm)

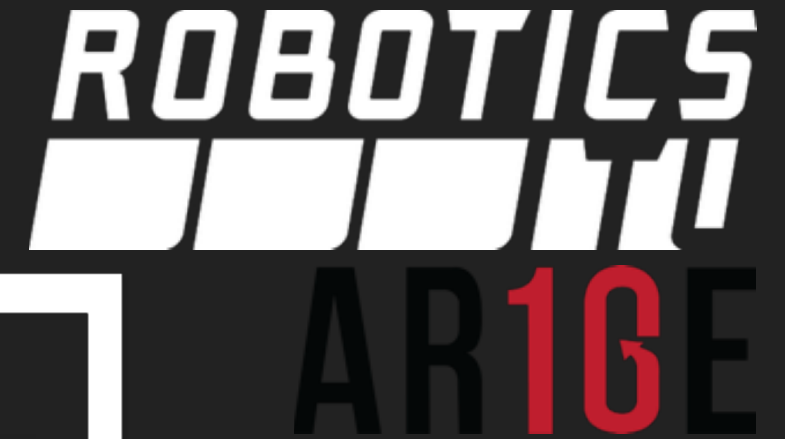
- ▶ Farklı kodların başına A ekleyen bir fonksiyon yazın. (örnek kod 1234567)

GEÇTİĞİMİZ HAFTADAN ÖRNEKLER



- ▶ 18 yaş sınırı ve hoş geldin mesajı. (örnek:"merhaba x, 18 yaşından küçük olduğun için sigara/alkol içemezsin.")
- ▶ 3 ürünlü market alışverişi. Stoklar belirtilecek, alış verişi fonksiyonu yazılacak. Çıktı verilecek.
- ▶ Data anlamlandırma. (N400, W300, S100)
- ▶ Sınav notları.(AA,BA,BB,CB,CC)

2016 ARIGE PYTHON EĞİTİMİNDEN



ARIGE - PYTHON'A GİRİŞ

2. HAFTA TEKRAR



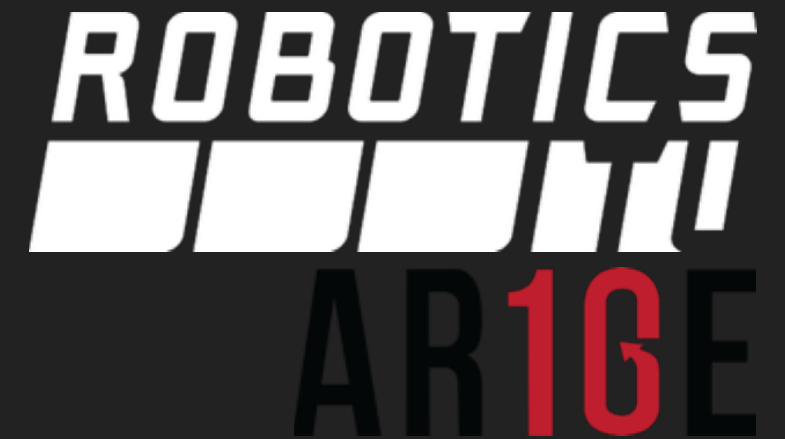
► Geçen hafta konuştuğumuz problemin çözümü

```
1 Arkadaşlar Merhaba son toplandığımızda şöyle bir problemden bahsetmiştik:
2
3 3 farklı hesap numarasında farklı bakiyeler vardır.
4 Kullanıcıdan kendi hesap numarasını girmesi istenir.
5 Girilen hesap numarası kayıtlı olanlara uymazsa program uyarı verir.
6 Girilen hesap numarasına göre kişinin bakiyesi ekrana yazdırılır.
7
8 Daha sonra 5 yiyecekli bir menü tanımlanır.
9 Kullanıcıdan bir sipariş girmesi istenir.
10 Siparişin fiyatı kişinin adisyonuna eklenir.
11
12 9. ve 10. satırda yazılı olanlar iki kere daha tekrarlanır.
13
14 Daha sonra kullanıcının adisyonu ekrana yazdırılır.
15 Eğer kullanıcının hesabındaki para adisyonu karşılamaya yeterliyse ekrana "afiyet olsun" yazdırılır.
16 Değilse ekrana "bulaşıkları yıka" yazdırılır.
17
18
19
20 Bir kaç ipucu vermek gerekirse.
21
22 Hesap numaraları tanımlanırken x1000, x1001, x1002 şeklinde tanımlanabilir.
23 Kullanıcıdan istenen hesap numarasının başına x eklenerek karşılaştırma yapılabilir.
24
25 if("x"+kullanıcı_h_no == "x1000"):
26     bakiye=x1000
27
28 Sipariş edilen yemeklerin adisyona eklenmesi başta adisyonun 0'a eşitlenmesi ile yapılır.
29
30 adisyon=0
31 tavuk = 12
32 adisyon += tavuk
33
34 gibi.
```

LİSTELER

- ▶ `empty = []`
- ▶ `numbers = [1, 3, 5, 7, 9]`
- ▶ `list1 = ['one', 2, "three", 4.2]`
- ▶ `colors = ['yellow', 'red', 'blue', 'green', 'black']`
- ▶ `list2 = ['one, two, three, four, five']`

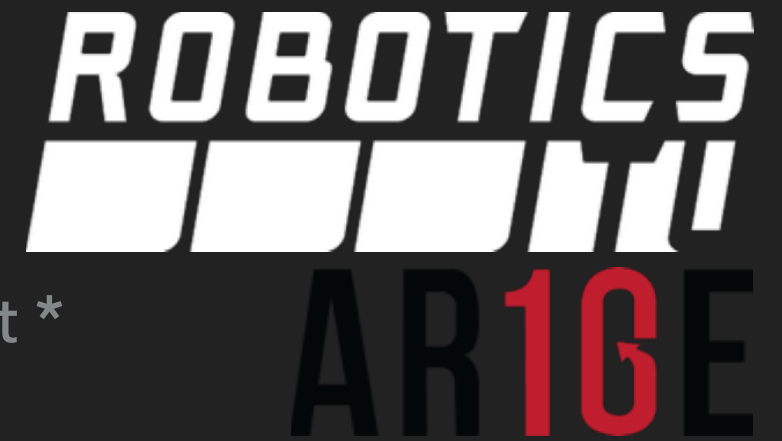




LİSTELER

BUILT-IN FUNCTIONS

- ▶ `list.append(x)`
 - ▶ `len(list)`
 - ▶ `list.remove(x)`
 - ▶ `list.pop(index of x)`
 - ▶ `list.insert(index, y)`
 - ▶ `list1.extend(list2)`
 - ▶ `del list[1]`
 - ▶ koşullar; ("red" in list), ("white" not in list)
 - ▶ List Reverse (do it with for loop)
 - ▶ `sorted(list, reverse=True)` , `list.sort()`
 - ▶ `s.count('o')`
 - ▶ `s = range(0,10,2)`
- ▶ `a[start:end]` # items start through end-1
 - ▶ `a[start:]` # items start through the rest of the array
 - ▶ `a[:end]` # items from the beginning through end-1
 - ▶ `a[:]` # a copy of the whole array
 - ▶ `a[start:end:step]` # start through not past end, by step
- ▶ `max(variable)`
 - ▶ `min(variable)`
 - ▶ `abs(variable)`
 - ▶ `type(variable)`



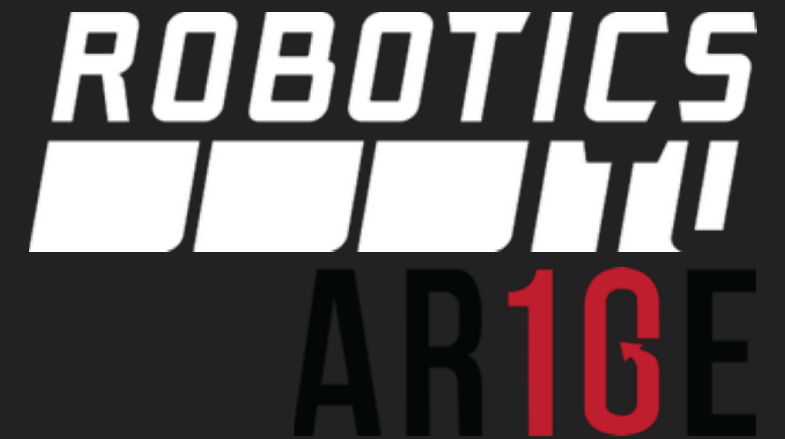
IMPORTING MODULES

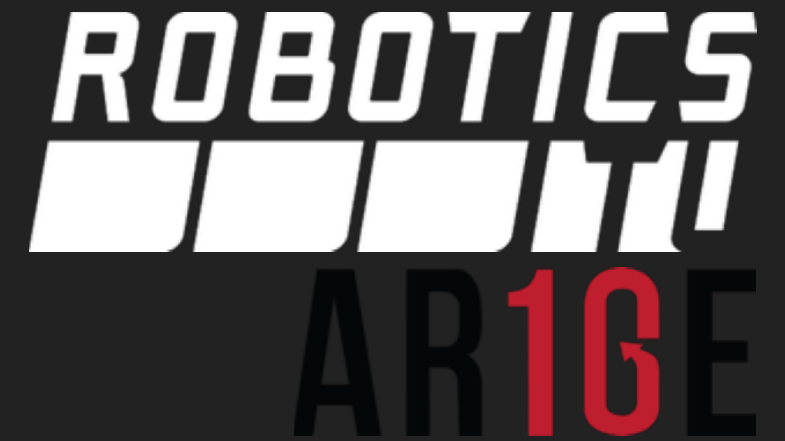
- ▶ `import math / from math import sqrt / from math import *`
- ▶ `print math.sqrt(25)`

- ▶ `import math` `# Imports the math module`
- ▶ `everything = dir(math)` `# Sets everything to a list of things from math`
- ▶ `print everything` `# Prints 'em all!`

ZAMAN

- ▶ `from datetime import datetime`
- ▶ `now = datetime.now()`
- ▶ `print ('%s/%s/%s %s:%s:%s' % (now.month, now.day, now.year, now.hour, now.minute, now.second))`





DICTIONARIES

- ▶ `{'key': value,
'key': value,
'key': value}`
- ▶ `dict["key"] = value`
- ▶ `adict={
'a':{'b':10,'c':20},
'd':{'b':30,'c':40}
}`

DICTIONARIES

- ▶ `print dict["key"]`
- ▶ `del dict["key"]`
- ▶ `koşul (key in dict)`
- ▶ `print dict.keys()`
- ▶ `dict.get("key", default = None)`

