

HALUK TANRIKULU

HERKES İÇİN

**python**  
**ile**  
**matematik**

WWW.PİTONCA.COM

## Yazar tarafından verilen eğitimler:

### Yapay Öğrenme ve Veri Analizi

Python ile Veri Analizine Giriş

Keras İle Derin Öğrenmeye Giriş

Makine Öğrenmesine Giriş

### Python Programlama Dili

Python ile Programlama (Giriş, Orta, İleri)

Python ile Nesne Yönelimli Programlama

Python ile Soket Programlama

Matematikçiler için Python Programlama

Network Mühendisleri için Python

Python ile Database Programlama

Python ile DSP'ye Giriş

Python ile Yapay Zeka Uygulamaları

Python ile Fourier Dönüşümleri

Python | Flask ile Programlama

Python | Django ile Programlama

Python ile Matematik

### HTML5, JavaScript, CSS ile Web Programlama

HTML5 ile Web Tasarımı

HTML5, JavaScript ve CSS ile Mobile Programlama

### PHP ve MySql İle Programlama

PHP'ye Giriş

PHP ve MySql ile Web Tasarımı

### Lua Programlama

Lua ile Mobil Oyun Geliştirme

Mikrokontroller için Lua Programlama

### Ağ Teknolojileri

MCNA Network | Yönlendirme | Ağ Anahtarlama

Huawei HCNA | HCNP

Cisco CCNA | CCNP

Network Fundamentals

Comptia Network+ | Cloud+

### Siber Güvenlik

Comptia Security+

Ethical Hacking Eğitimleri (CEH)

Python ile Güvenlik Uygulamaları Geliştirme

Sistem Güvenliğine Giriş

### İşletim Sistemleri

Linux (Mint, RedHat, Fedora, Kali, Debian, Ubuntu, SuSE)

Windows Servers

# Python ile Matematik

Ver. 1.05 – 2018

## Haluk Tanrikulu

[www.haluktanrikulu.com](http://www.haluktanrikulu.com)

[tanrikul@metu.edu.tr](mailto:tanrikul@metu.edu.tr)



**MCNA** | **Pitonca**

[info@pitonca.com](mailto:info@pitonca.com) | 312 256 16 16

[www.mcnatech.com](http://www.mcnatech.com) | [www.pitonca.com](http://www.pitonca.com)



**Ankara 2018**

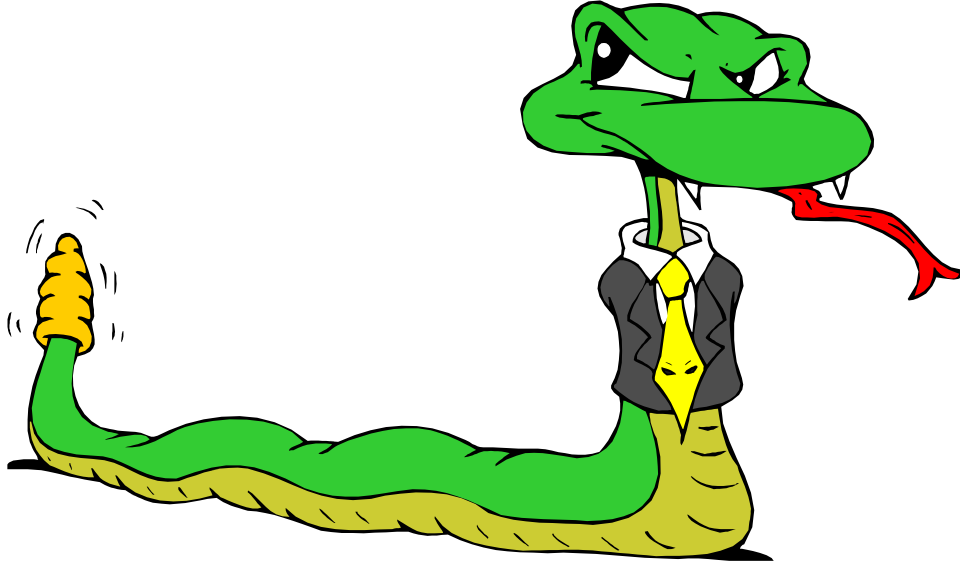
18 17 16 15 14 13 12 11

**Fazi ve Bıdıklarına,**

# Python ile Matematik

## 3. Bölüm

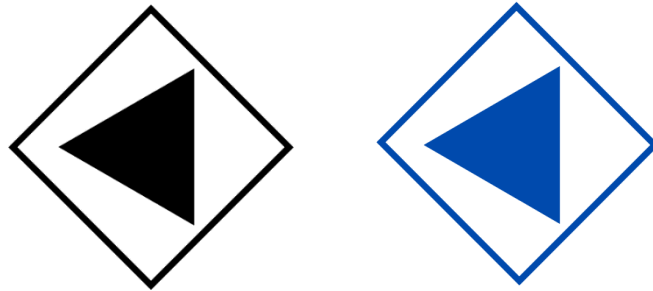
**Haluk Tanrıkulu**



## Proje Çalışması

Düzgün çokgenleri daire çizerek oluşturmayı bir önceki derste öğrenmiştik. Bu derste ise Pitonca Education'ın logosunu yine daire çizerek oluşturmaya çalışacağız.

Aşağıda Pitonca Education logoları görülmektedir. Logo bir kare içine eşkenar üçgen yerleştirilmiş ve içi siyah renkle boyanmıştır.

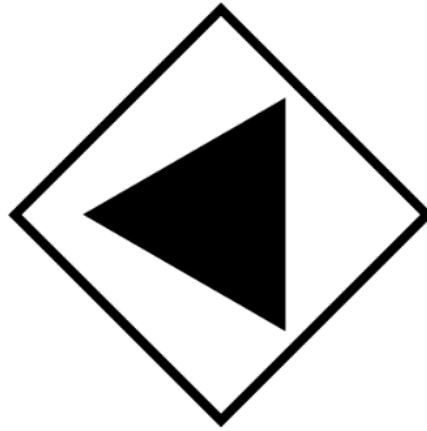


Şimdi bu logoyu Python kullanarak çizdirelim.

### Daireler Çizelim

Logo da iki daire oluşturalım.

- **Dış Daire** : Dış daire 100 piksel yarıçapında olsun. Daire karenin köşelerini üzerinde bulunduracaktır.
- **İç Daire** : İç daire 60 piksel yarıçapında olup, bir köşesi karenin sol köşesi ile aynı hizada olacaktır.
- **Üçgenin İçinin Boyanması** : Üçgenin için boyamak için `.fd()` fonksiyonu kullanmak bizim için çok daha kolay olacaktır. Ancak biz bu tür çizimlerde ne karenin nede üçgenin kenar boylarını bilmiyoruz. Bildiğimiz tek şey, iç içe olan bu dairelerin 100/60 oranında olmasıdır. Bu yüzden `.goto()` kullanacağız. İçin doldurmak için ise `begin_fill()` ve `end_fill()` fonksiyonlarının kullanacağız.



## 1. Adım - Karenin Çizimi :

Daireleri çizelim. Önce dıştaki kareyi çizelim. Önce kenar sayısını 4 olarak belirleyelim. Logoda karenin kenarlarının kalın çizildiği görüldüğünden `.pensize()` değerini 5 olarak belirledik

```
import turtle
t = turtle.Pen()
t.pensize(5)
t.speed(0)
kenar_sayısı= 4
```

Şimdi ise karenin çizimi yapalım. Çizime başladığındaki konumu  $(x_1, y_1)$  iken ilk döngüde yani ilk 90 derecelik açıda  $(x_2, y_2)$  konumuna gidiyor.  $(x_2, y_2)$  konumundan  $(x_1, y_1)$  konumuna bir çizgi çizip tekrar  $(x_2, y_2)$  konumuna dönen bir yapı kuruyoruz. Sonra  $(x_1, y_1)$  konumunun değerleri  $(x_2, y_2)$  konumu ile değiştiriyoruz. Burası çok önemli!


Bunu daha basit bir şekilde yapabilir misin?

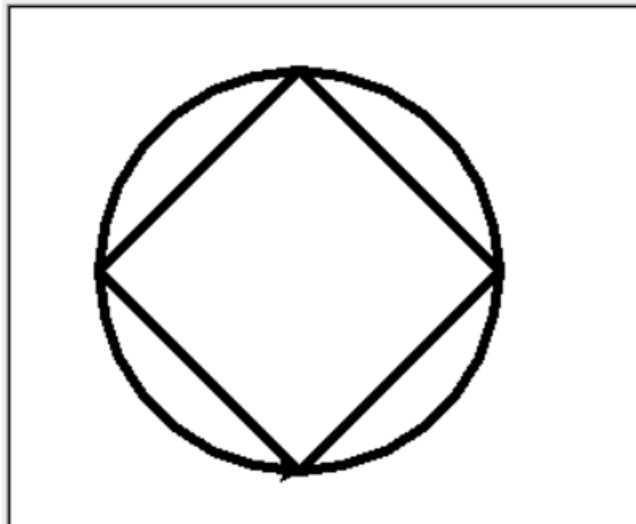
```
for i in range(kenar_sayısı):
    x1, y1 = t.pos()
    aci = 360 / kenar_sayısı

    t.circle(100, aci)

    x2, y2 = t.pos()

    t.goto(x1, y1)
    t.goto(x2, y2)
    x1, y1 = x2, y2
```

 Python Turtle Graphics



Şimdi daireyi kaldıracak bir hamle yapalım. `.penup()` bizim işimize yarayacaktır. Tüm kodları aşağıdaki tekrar yazıp çalıştıralım.

```
import turtle
t = turtle.Pen()

t.pensize(5)
t.speed(0)
kenar_sayısı = 4

for i in range(kenar_sayısı):
    x1, y1 = t.pos()
    aci = 360 / kenar_sayısı

    t.penup()
    t.circle(100, aci)
```



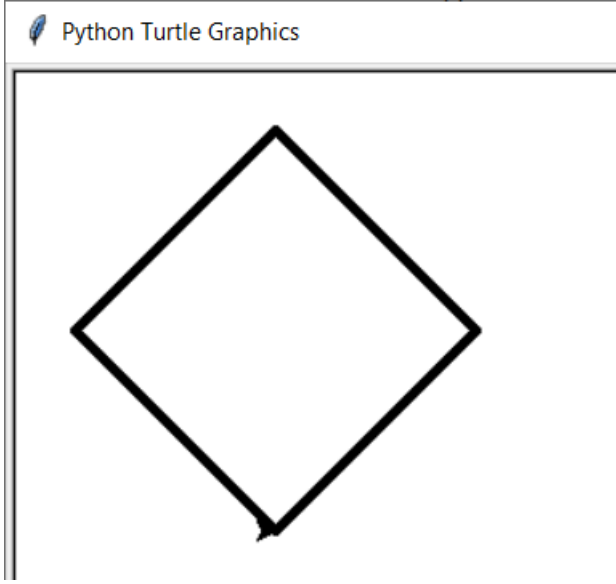
**t.pendown()**

```
x2, y2 = t.pos()
```

```
t.goto(x1, y1)
```

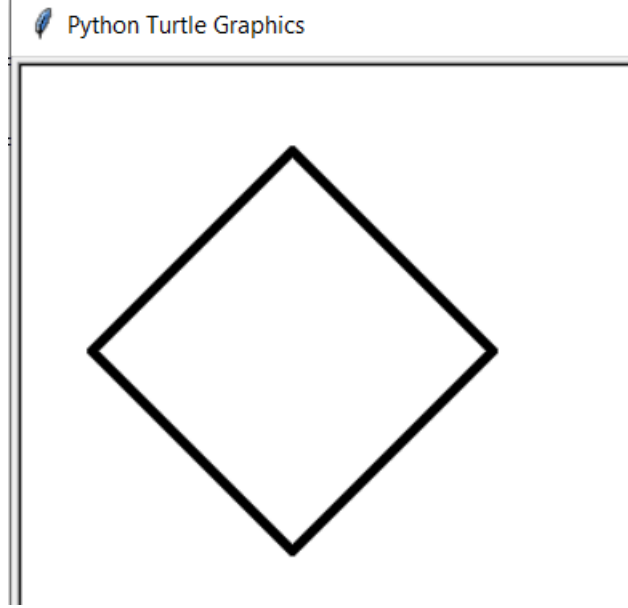
```
t.goto(x2,y2)
```

```
x1, y1=x2, y2
```



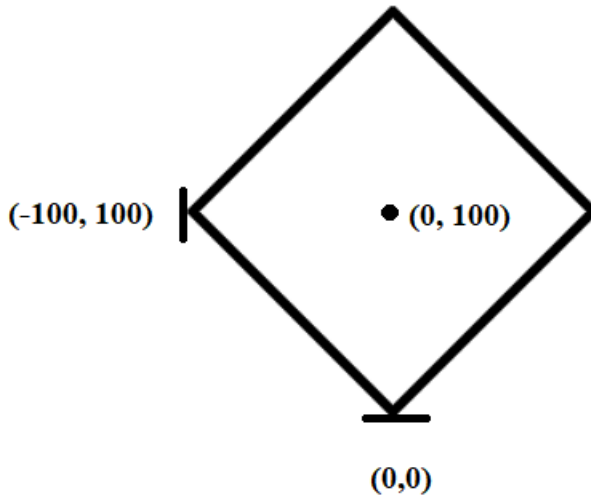
Görüntüde başladığı yere dönen ikonun karenin en alt kısmında görüyoruz. Bunun görünmesini istemeyiz. Bunun için sadece t.penup() kullanmak yeterli olur mu?

Şimdi oraya t.hideturtle() ekliyoruz. Bu ikonun sonda görünmemesini sağlasın.



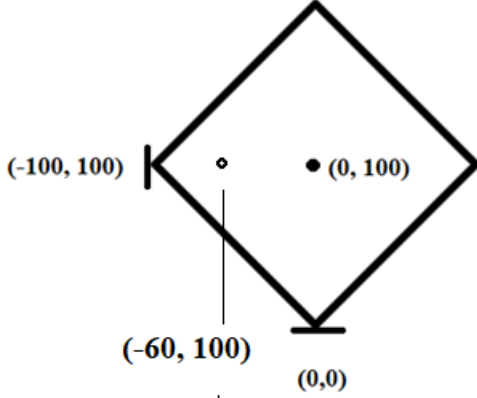
Evet bu daha iyi oldu.

## 2. Adım – Üçgenin Çizimi :



Resimde görüldüğü gibi çizime (0,0) noktasında başladık ve aynı noktada bitirdik. Dairemizin merkezi ise (0,100). Logomuzda karenin sol köşesi ile aynı hizada bir köşesi olan üçgenimiz var.


O zaman üçgenin bir köşesi  $(-60,100)$  noktasında olması gerekir. Unutmayın tek bildiğimiz iç içe iki dairenin yarıçapları, biri 100 piksel diğeri ise 60 piksel değerinde.

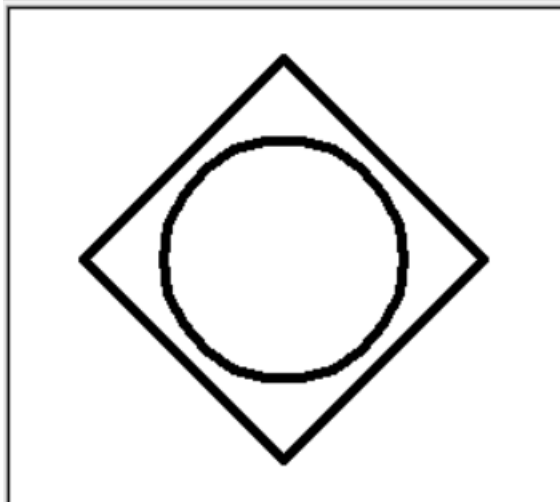


Şimdi  $(-60, 100)$  noktasından başlayan (tabii ki yönü aşağı doğru olacak) bir çember çizelim.

```
t.goto(-60, 100)
t.right(90)
a = 3
t.pendown()
x1, y1 = -60, 100
```

```
t.circle(60)
```

 Python Turtle Graphics



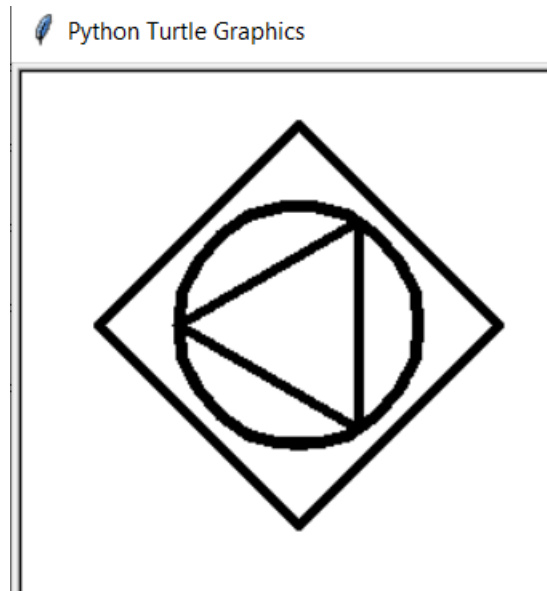
Bu çemberin üzerinde 3 köşesi olan bir üçgen çizelim.

```
for i in range(3):
```

```
    acı = 360 / a
    #t.color("white")
    #t.begin_fill()
```

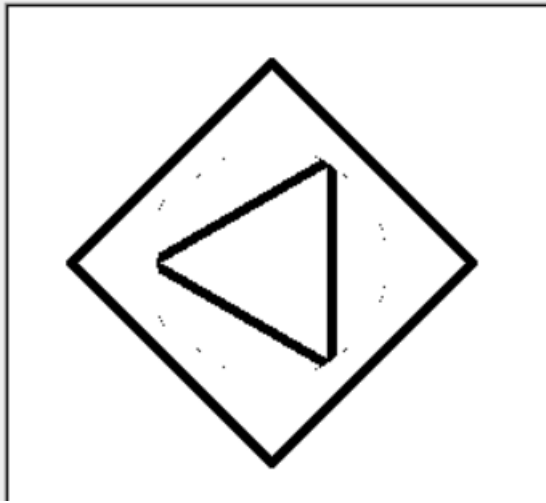
```
    t.circle(60, acı)
    #t.end_fill()
    t.pendown()
    x2, y2 = t.pos()
```

```
    t.goto(x1, y1)
    t.goto(x2,y2)
    x1, y1=x2, y2
```



Burada iç çemberden kurtulalım. Bunun için for döngüsü bitimine aşağıdaki kodları ekleyelim. Siyah çemberin üzerinden beyaz ile geçince kare ve üçgen ortaya çıkacaktır.

```
t.color("white")
t.pensize(6)
t.circle(60)
```



Görüldüğü üzere çemberin silmesine rağmen izler kaldı. 3. Adımda üçgenin içini boyalım ve bu izlerden kurtulalım.

### 3. Adım – Üçgenin İçin Boyanması:

Herşeyden önce `.goto()` kullandığımızda `begin_fill()` ve `end_fill()` kullandığımızda için dolduramıyoruz. Bunun için ters bir işlem yapacağız.

Önce üçgeni oluşturmak için kullandığımız çemberin içini siyah ile doldurup daha sonra üç adet yayın bulunduğu alanları beyaz ile dolduracağız. Bu işlemleri yapan tüm programı aşağıda bulabilirsiniz.

```
import turtle

t = turtle.Pen()

t.pensize(5)
t.speed(0)
a= 4
for i in range(a):
    x1, y1 = t.pos()
    aci = 360 / a

t.penup()
```

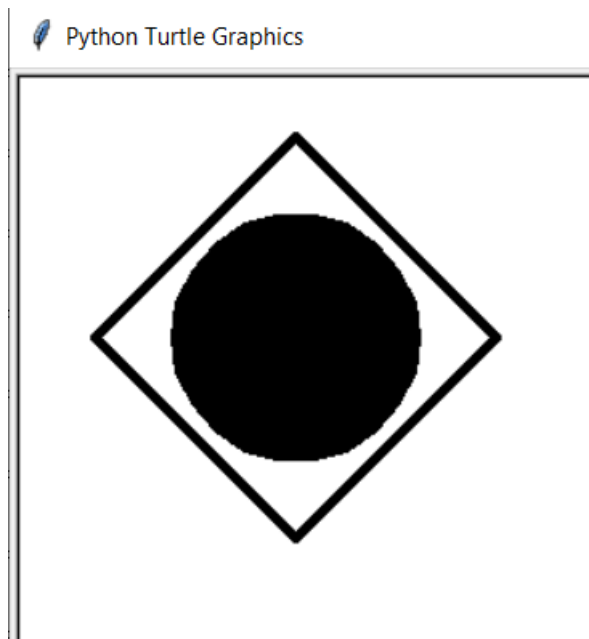
```
t.circle(100, acı)
t.pendown()

x2, y2 = t.pos()

t.goto(x1, y1)
t.goto(x2,y2)
x1, y1=x2, y2
t.hideturtle()
t.penup()

t.goto(-60, 100)
t.right(90)
a = 3
t.pendown()
x1, y1 = -60, 100

t.begin_fill()
t.circle(60)
t.end_fill()
```



Şimdi yapacağımız yayların için beyaz boyamak. Projenin son hali aşağıdadır.

```
import turtle
```

```
t = turtle.Pen()

t.pensize(5)
t.speed(0)
a= 4
for i in range(a):
    x1, y1 = t.pos()
    aci = 360 / a

    t.penup()
    t.circle(100, aci)
    t.pendown()

    x2, y2 = t.pos()

    t.goto(x1, y1)
    t.goto(x2,y2)
    x1, y1=x2, y2
t.hideturtle()
t.penup()

t.goto(-60, 100)
t.right(90)
a = 3
t.pendown()
x1, y1 = -60, 100

t.begin_fill()
t.circle(60)
t.end_fill()

for i in range(3):


    aci = 360 / a
    t.color("white")
    t.begin_fill()

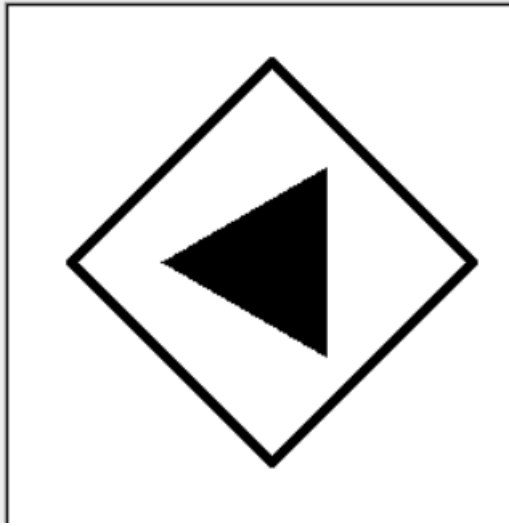
    t.circle(60, aci)
    t.end_fill()
    t.pendown()
    x2, y2 = t.pos()

    t.goto(x1, y1)
    t.goto(x2,y2)
```

$x_1, y_1 = x_2, y_2$

```
t.pensize(5)  
t.circle(60)
```

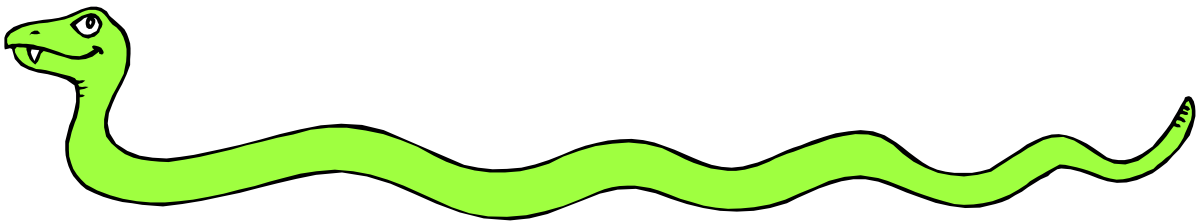
 Python Turtle Graphics



## ÖDEV :

1 - Logonun üçgenini farklı renklerde yapınız.

2 – (-60, 100) noktası yerine (-70, 100) noktasından başladığınızda görüntü nasıl olur?



Bu yayının tüm hakları saklıdır. Hiçbir şekilde kopyalanamaz, çoğaltılamaz.