# ÜNİVERSİTE SEVİYESİNDE PYTHON NOTLARI:

**Grafikler:**

Python grafikleri bize ekran üzerine görsel imgeler oluşturmamıza yarayan kodlardır.

*Graphics* konusunda kod yazarken dikkat etmemiz gereken birkaç püf nokta ve sıkça kullanacağımız bazı komutlar mevcut.

1. **Kütüphaneyi Çağırmak**

* Daha önce Karel’da yazdığımız gibi kodlamaya başlamadan önce ihtiyacımız olan kütüphaneleri çağırmamız (import etmemiz) gerekiyor.



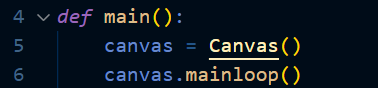
* Bu şekilde Canvas kütüphanesini graphics dosyasının içerisinden çağırmamız lazım.
* Bunu yazmayı unutursak karşımıza şu hata kodu çıkacaktır.



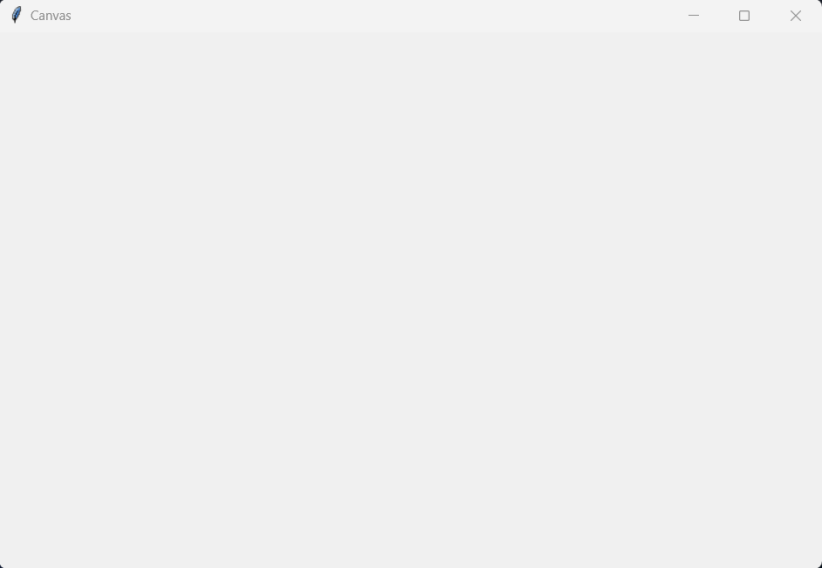
* Gördüğünüz gibi “*Canvas*” yerine “*Canva*” yazdım ve *Python* bunu anlayamadı, tıpkı onun benim duygularımı anlamadığı gibi...

1. **Canvas’ı Oluşturmak ve Ekranda Okutmak**

* Canvas kullanarak bir görüntü oluşturabilmek için öncelikle Canvas’ı oluşturmamız ve ekranda sürekli görünmesi için okutmamız gerekiyor.



* **main()** fonksiyonunun içerisine yazıyoruz kodlarımızı her zamanki gibi
* **canvas = Canvas()** ile Canvas’ımızı oluşturuyoruz
* **canvas.mainloop()** komutu ile de ekranda görünmesini sağlıyoruz. Bu sayede şöyle bir görüntü ortaya çıkıyor:



* Gördüğünüz gibi boş bir pencere oluşturduk
* Peki **canvas.mainloop()** komutunu kullanmasaydık ne olurdu?

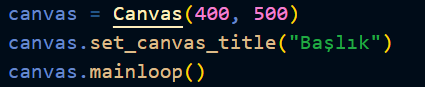
=> Ekranda bir anda bir canvas oluşur ve **mainloop()** olmadığı için ortadan kaybolur. Bunu kendiniz deneyerek görebilirsiniz.

1. **Canvas’ın Boyutlarınıı Ayarlamak**



* Canvas komutunun içerisine yazdığımız birinci değer **genişlik**, ikinci değer **yükseklik** oluyor.

1. **Canvas’a Başlık Eklemek**



* Canvas.set\_canvas\_title() komutu sayesinde başlık ekliyoruz. Çıktı bu şekilde oluyor:

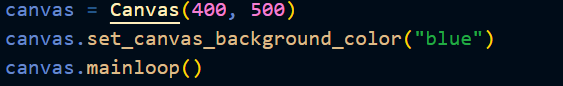


**!!! ÖNEMLİ NOT: Canvas kütüphanesinde bir şeyleri eklemek/ayarlamak için genelde canvas.set\_... şeklinde komutlar kullanırız.**

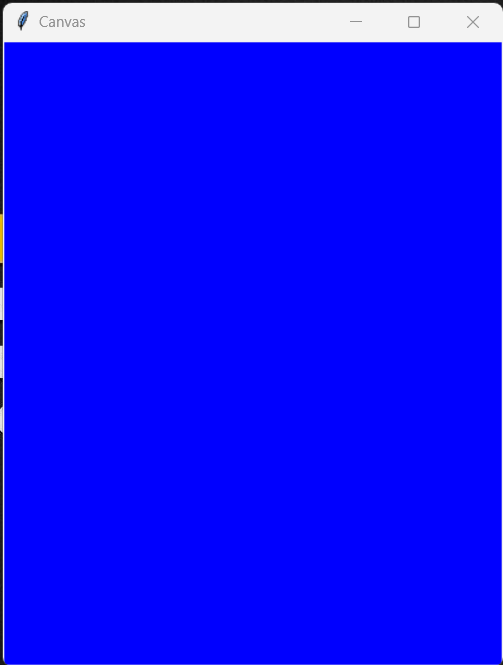
**!!! Komutlarımızın büyük bir çoğunluğu canvas.set\_... veya canvas.get\_... şeklindedir.**

1. **Canvas’ın Arka Plan Rengini Değiştirmek**

* Gelelim Canvas’ımızı renklendirmeye

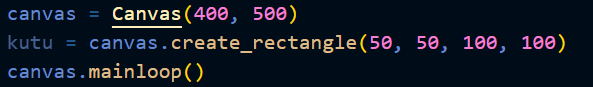


* **canvas.set\_background\_color()** komutu sayesinde arka planın rengini değiştirebiliyoruz ve ortaya böyle bir görüntü çıkıyor:

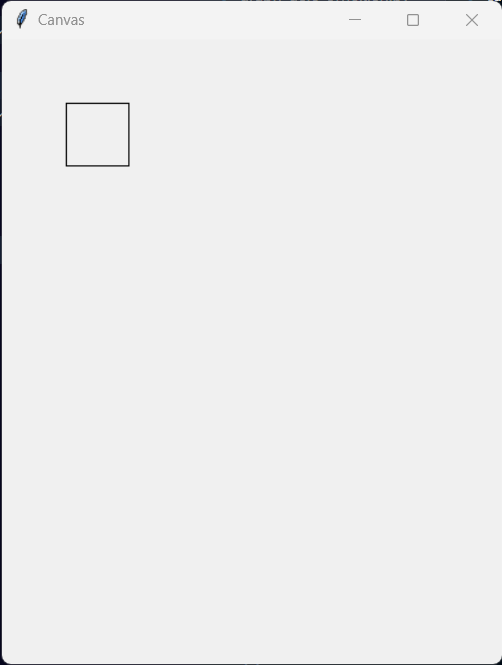


1. **Canvas’a Şekil Eklemek**

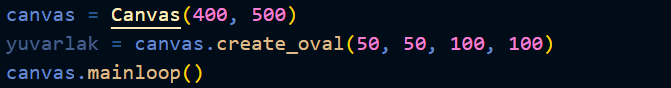
* Canvas’ınıza dörtgen, yuvarlak ve çizgi şekilllerini çizmemize olanak tanıyor.
  1. **Dörtgen Eklemek**



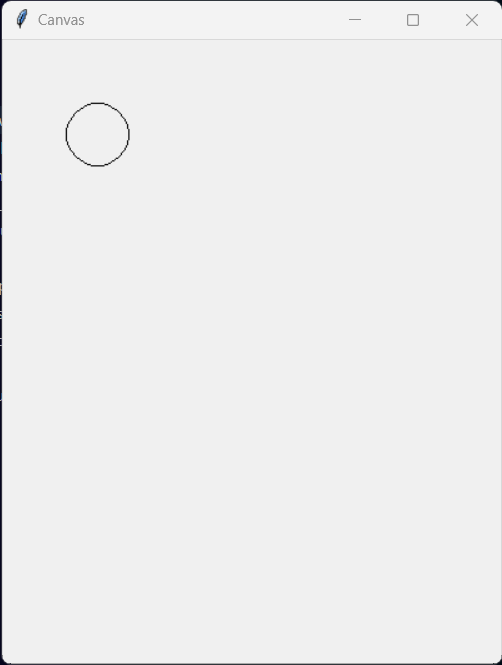
* **canvas.create\_rectangle()** komutu anlaşıldığı üzere bize bir dörtgen oluşturmak için yardım ediyor.
* **canvas.create\_rectangle(x1, y1, x2, y2)** parantez içerisindeki değerleri dörgenin **sol üst** ve **sağ alt** konumununu belirlemeye yarıyor.



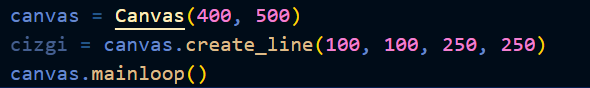
* Sonucunda da boş Canvas’ımızın üzerinde sol üst ve sağ alt koordinatlarını girdiğimiz bir dörtgen oluşuyor.
  1. **Yuvarlak Eklemek**
* Aynı şekilde bir yuvarlak oluşturmak istediğimizde de benzer komutları kullanıyoruz, hiçbir fark yok.



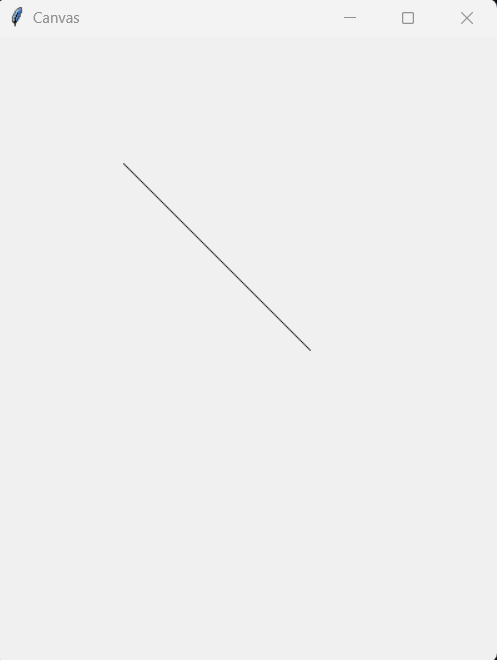
* **canvas.create\_oval()** komutu sayesinde bir yuvarlak oluşturabiliyoruz.
* **canvas.create\_oval(x1, y1, x2, y2)** parantez içerisindeki değerler ise Python’un bizim için yuvarlağımızı içine alan bir dörtgen hayal etmesine yardımcı olmak içindir.

,

* Bu şekilde Canvas’ımızın üzerinde bir yuvarlağımız oluştu.
  1. **Çizgi Eklemek**
* Canvas’ımızın üzerine çizgi eklemek de tıpkı dörtgen ve yuvarlak gibi kolay bir işlemdir.

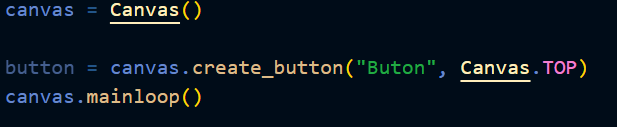


* Çıktı olarak da ekranın (100, 100) noktasından (250, 250) noktasına uzanan bir çizgi oluşmuş oldu.

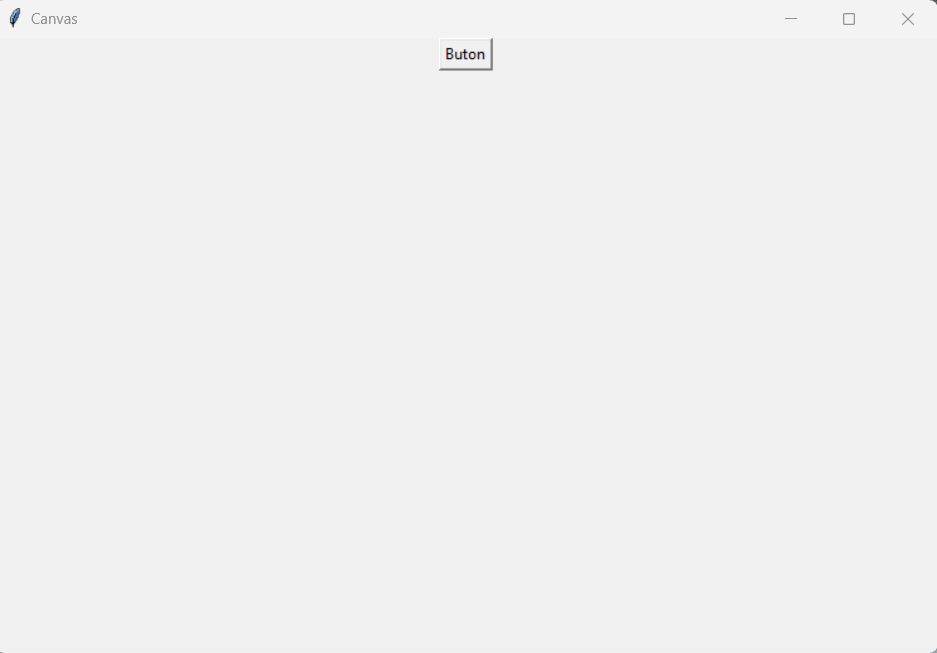


1. **Canvas’a Buton Eklemek**

* Canvas’a bir şeyler eklerken her zaman kullandığımız canvas.create\_....() komutunu kullanıyoruz.
* **canvas.crete\_button(“Başlık”, LOKASYON)** şeklinde yazarak ekleriz.



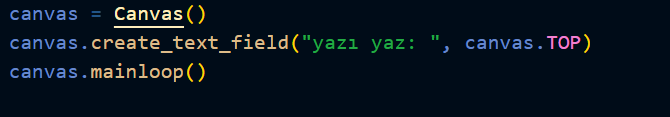
* Çıktı olarak da karşımıza şu çıkıyor:



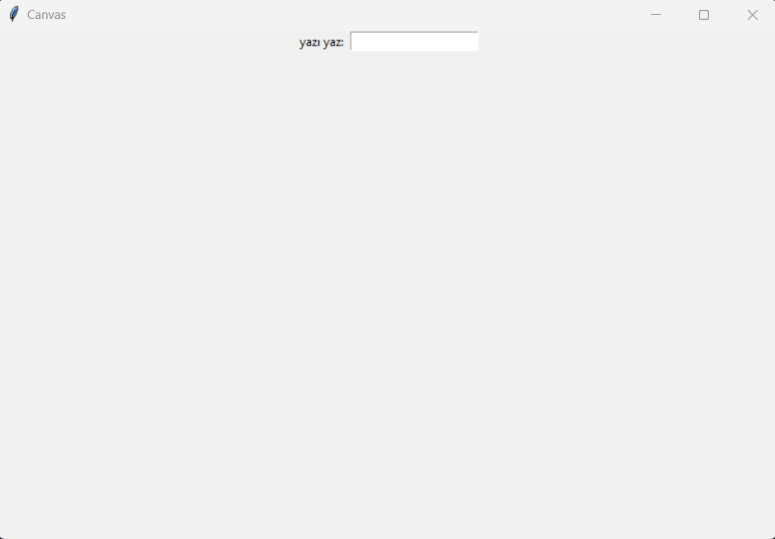
* Gördüğünüz gibi Canvas’ın “TOP” lokasyonunda bir buton oluşturmuş olduk.
* Lokasyonu eklemezsek Canvas butonu nereye koyacağını bilemeyecek ve çalışmayacaktır!

1. **Canvas’a Text Field Eklemek**

* Buraya kadar okuyan ve Türkçe bilen birisi için canvas’a nasıl bir komponent ekleneceğini açıklamayacağım :)
* Kodlardan anlayabileceğine inanıyorum.



* Çıktı:



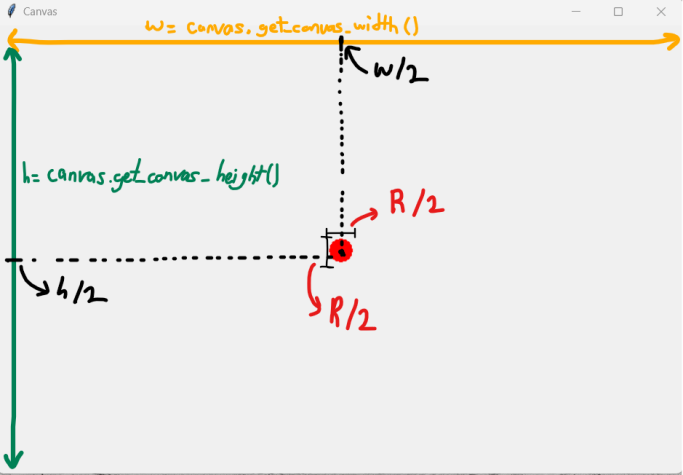
**!!! Buraya kadar olan kısımda Canvas ve graphics’in temel özelliklerine değindik. Buradan sonrasında daha geniş kapsamlı örneklerin çözümlemesini yapacağız.**

1. **Ekranın Tam Merkezine bir şekil yerleştirmek**

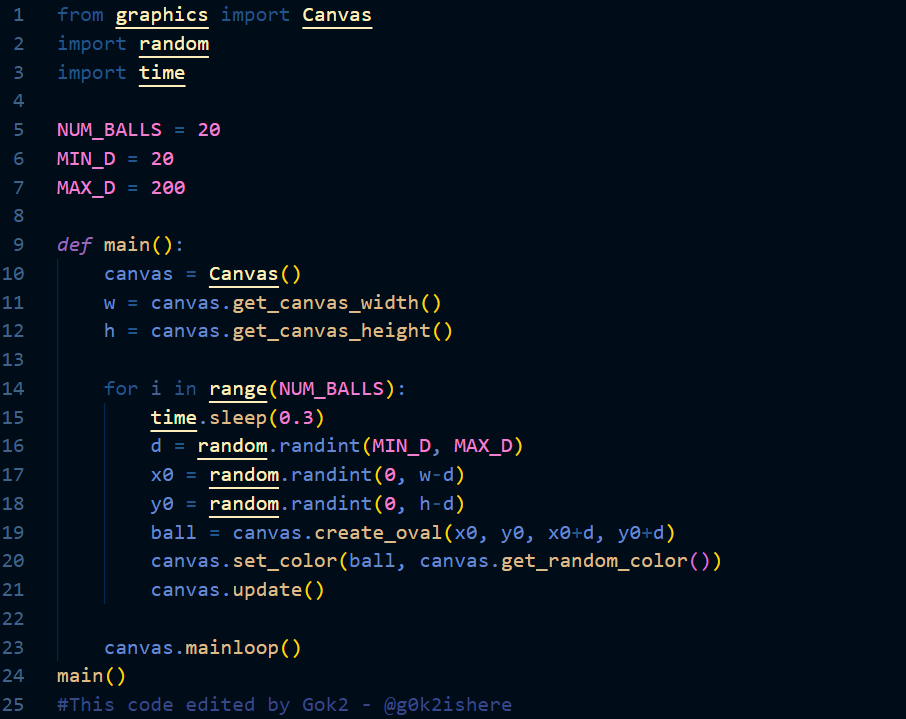


* Bu projede amacımız Canvas’ımızın boyutları ne olursa olsun tam merkez noktasını bulup oraya bir daire şekli oluşturmak.

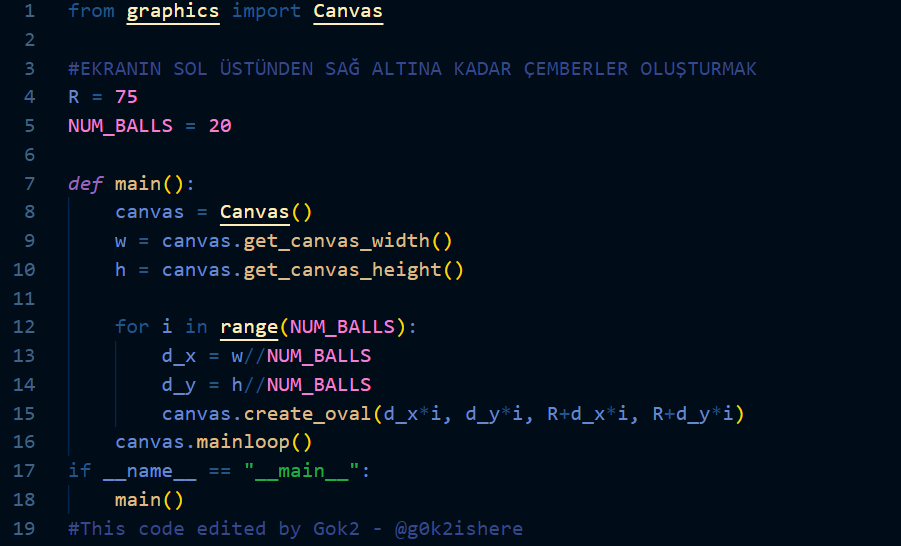
1. “R” isminde bir yarıçap değeri oluşturduk.
2. “w” ve “h” değerlerine canvas’ımızın genişlik ve yükseklik değerlerini atadık. (canvas.get\_canvas\_width/height() fonksiyonları sayesinde Canvas’ın genişliğini/yüksekliğini her zaman bize verecektir).
3. Yerleştireceğimiz şeklin merkez koordinatlarını belirlemek için merkez\_x ve merkez\_y değişkenleri belirledik. Bu değişkenler matematiksel olarak yazmış olduğumuz işlem ile oluşturacağımız şeklin merkezi ile canvas’ın merkezini çakışacak şekilde hesaplıyor.



1. Bu işlemde canvas’ın genişlik ve yüksekliğinin yarısından şeklimizin çapının yarısını çıkartıyoruz ki tam merkeze denk gelsin.
2. Canvas.create\_oval(merkez\_x, merkez\_y, merkez\_x+R, merkez\_y+R) kodu ile şeklimizi her seferinde merkeze yerleştirmiş olacağız.
3. Canvas.set\_color() ile şeklimizin rengini ayarladık.
4. Canvas.mainloop() ile Canvas’ın ekranda sürekli görünemsini sağladık ve en sonundada kodları yazdığımız main fonksiyonunu çalıştırıyoruz.
5. **Ekrana Rastgele Toplar çıkartmak**



1. Kullanacağımız kütüphaneleri birinci satırdan itibaren import ediyoruz.
2. Kullacağımız global değişkenleri (açılmış olan dosyanın herhangi bir yerinde birden fazla kez kullanılabilir halde) belirliyoruz.[“MIN\_D” değeri oluşturalacak topların minimum çapını, “MAX\_D” değeri de maksimum çapını verir.]
3. “w” ve “h” değerlerine canvas’ımızın genişlik ve yükseklik değerlerini atadık. (canvas.get\_canvas\_width/height() fonksiyonları sayesinde Canvas’ın genişliğini/yüksekliğini her zaman bize verecektir).
4. For döngüsü ile NUM\_BALLS kadar (yani 20) top üretmesini sağlıyoruz.
5. [15.satırdaki] time.sleep(0.3) komutu ile topların kaç saniye aralıklarla ekrana yazdırılacağını belirliyoruz.
6. [16. satırdaki] d = random.randint(MIN\_D, MAX\_D) komutu ile çapı MIN\_D ve MAX\_D arasında olan toplar oluşturulur.
7. [17-18. satırdaki] x0 ve y0 değerleri topların oluşturacağı rastgele koordinatların x ve y eksenlerini verir. ! w-d ve h-d yapma sebebimiz: topların ekrandan dışarı taşmasını istemiyoruz.
8. [20. satırdaki] Oluşturulan toplara random renk atıyoruz.
9. [21. satırdaki] canvas.update() komutu ekranı yenilemeye yarıyor. Yazmayı unutursanız düzgün çalışmayacaktır.
10. **Ekrana Ardışık Şekiller Çizdirmek**



1. Kütüphaneleri import ediyoruz
2. Global değişkenlerimizi belirliyoruz
3. [12. satırdaki] İşin matematiği 12. satırda başlıyor. Bir for döngüsü ile d\_x ve d\_y değişkenlerini oluşturuz ve sürekli artan i değerini, her defasında oluşturduğumuz yuvarlakların koordinatlarını belirlemede kullanıyoruz.
4. [15. satırdaki] Yuvarlakların başlangıç ve bitiş noktalarını her seferinde arttırıyoruz. Hem x hem y eksenlerinde ötelendiği için çapraz bir görüntü oluşur.
5. 