

## Jupyter Notebook Kullanımı

Jupyter notebook daha önce değinildiği üzere Anaconda ile gelen bir uygulamadır. Özelliği ise hem hatırlatma notları yazılabilmesi hem de Python ile kod yazılabilmesidir. Başlangıç safhasında yani Python programlama dilini öğrenmeye yeni başlamışken hatırlatma notlarını tutmak pratik olacaktır.

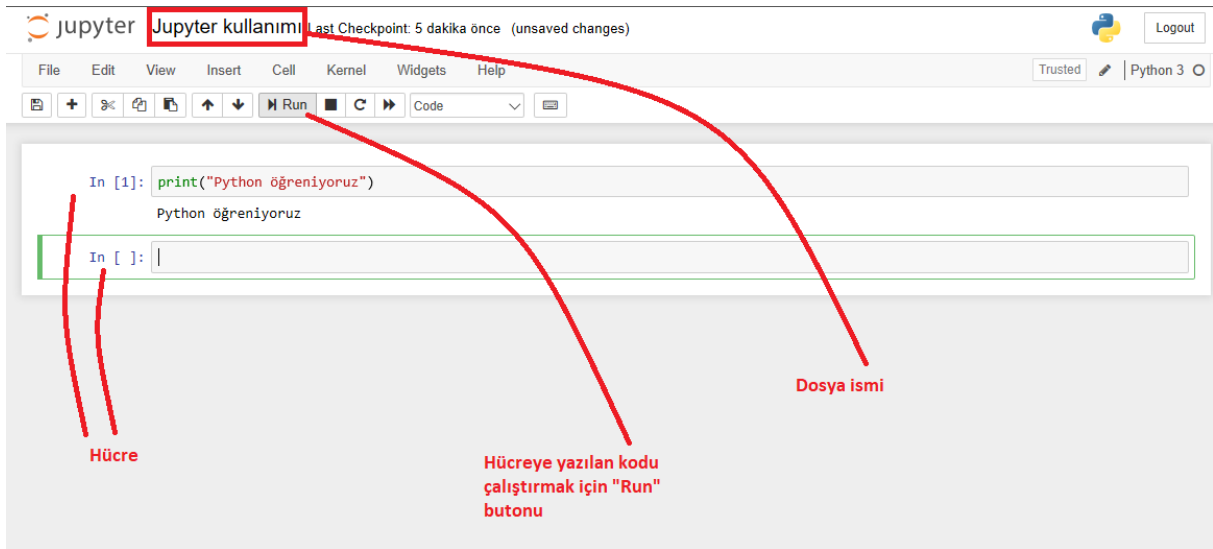
Jupyter Notebook' u Anaconda Navigator'dan Launch diyerek ya da (kurulum aşamasında sistem pathine eklenmesi seçilmişti) Windows terminalinden Jupyter Notebook yazarak çağırabiliriz. Anımsanacağı üzere Jupyter Notebook web arayüzü ile açılacaktır ve beraberinde de bir server açılacaktır. *Bu serverın kapatılmaması gerekmektedir!*

Jupyter Notebook açıldığında **C:\Users\kullanıcı ismi>** olarak açılır ve adres satırında da **http://localhost:8888/tree** olarak görünür. Eğer web arayüzü ile Jupyter notebook açılmazsa, Jupyter notebooku açmanın bir üçüncü yolu da, serverda görünen adresi **Ctrl+C** ile kopyalayıp web arayüzündeki adres satırına bu adres yazılırsa Jupyter notebook yine açılacaktır.

Jupyter Notebook açıldıktan sonra (kendi bilgisayarındaki) **C:\Users\kullanıcı ismi>Desktop>Python çalışmaları>Jupyter>** klasörüne gidiyorum. Jupyter Notebook' un bir özelliğide yapılan her işlemi otomatik kaydetmesidir. Yeni çalışma dosyası açmak için sağ üst köşeden New seçeneği ile Python 3 dosyası seçilmeli ve Untitled bölümüne giderek uygun bir isim verilmelidir.

Jupyter notebookun birçok özeliği vardır ve bunları akılda tutmak zor olabilir. Bu amaçla bir manual yani el kitabı hazırlanmıştır. Bu el kitabına, <https://jupyter.brynmawr.edu/services/public/dblank/Jupyter%20Notebook%20Users%20Manual.ipynb> adresinden ulaşılabilir. Böylece detaylı kullanım için gereken bilgiler buradan elde edilebilir.

Basit bir örnek aşağıda verilmiştir.

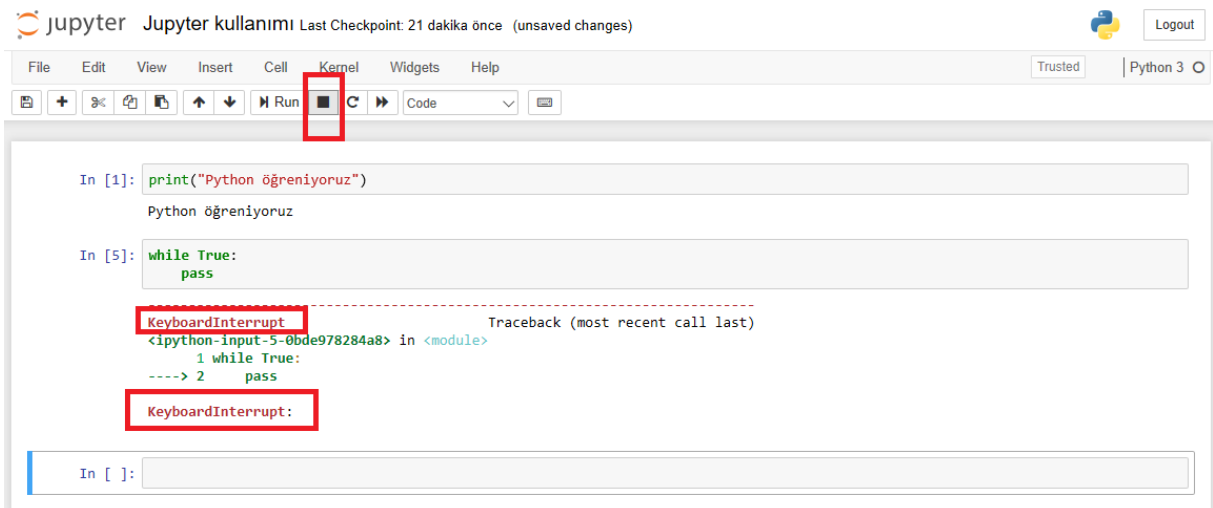


**In [ ]** yazan bölümler hücre olarak adlandırılmaktadır. Uygun formatta yazılan komutu çalıştırmak için Run Cell (Hücreyi çalıştır) anlamında şekilde gösterilen butona basılır. Bu basit uygulamada

**print()** komutu ekrana programın çıktılarını yazdırmak amacıyla kullanılan komuttur. Yazdırılacak bilgi tırnak içerisinde yazılır. Örneğin yukarıdaki uygulamada **Python öğreniyoruz** çıktısı programın çalıştırılmasıyla yazdırılmıştır.

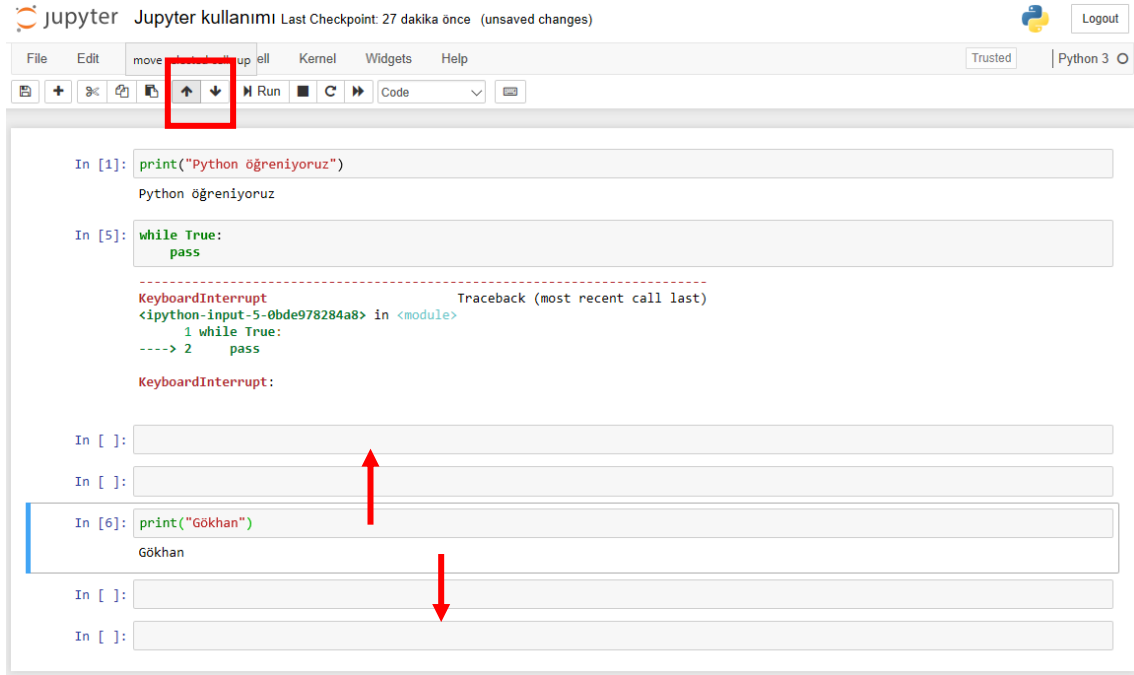
Hücre çalıştırdıktan sonra yeni bir hücre oluşturulur. Dikkat edilirse, hücre çalıştırılmadan önce köşeli parantez içerisinde herhangi bir rakam yokken, hücre çalıştırdıktan sonra köşeli parantez için 1 rakamı yazılmıştır. Bu, çalıştırılmış ve işlemi yapılmış hücre sayısını göstermektedir. Run düğmesine basıldıktan sonra çalışan bir hücrede köşeli parantez içerisinde \* görünür. \*, hücre üzerinde hala işlem yapıldığı anlamına gelmektedir. Hücreyi çalıştırmak için Run butonu kullanılabileceği gibi klavyeden **Shift+Enter** tuşlarına da basılabilir.

Bazı durumlarda program kısır döngüye girebilir veya programın sonuç vermesi uzun sürebilir. Bu tip durumlarda Run düğmesinin yanındaki **Interrupt** (*kesmek, yarıda kesmek*) düğmesine basılabilir.

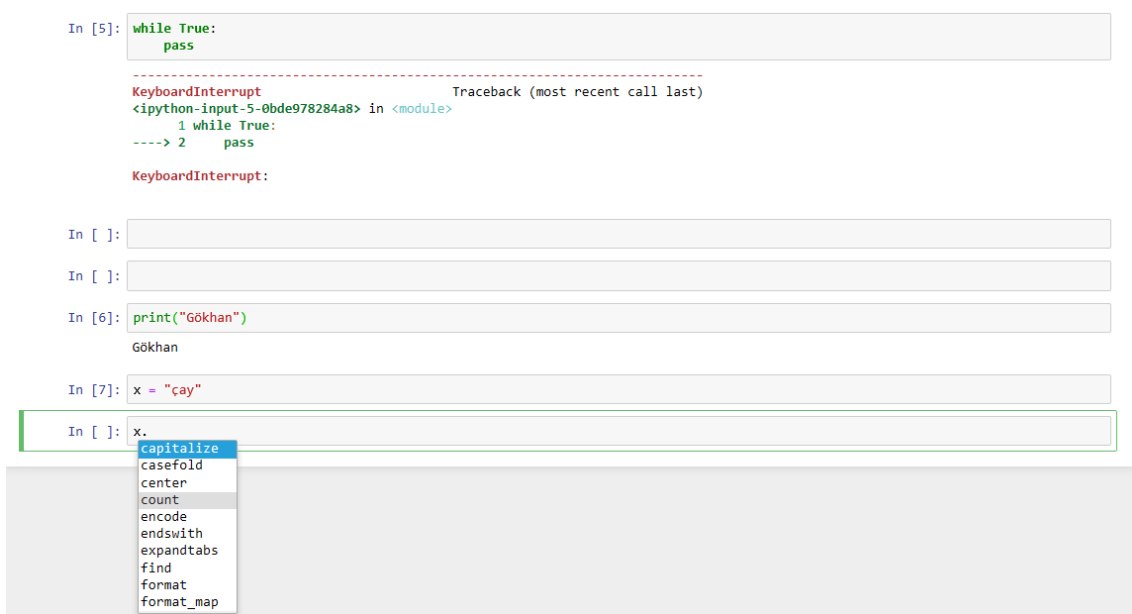


Interrupt seçeneği ile program durdurulduğunda ya da yarıda kesildiğinde dikkat edilirse hücrenin altında **KeyboardInterrupt** ve diğer bazı bilgiler gösterilmektedir. Aynı işlem **Kernel** sekmesinden **Interrupt** denilerek te yapılabilir.

Bir hücrenin altına sol üstteki + düğmesine basarak gerektiği kadar hücre eklenebilir. Hücrelerin yerlerini değiştirmek için de Run tuşunun sol tarafındaki yukarı ve aşağı ok düğmeleri kullanılabilir.

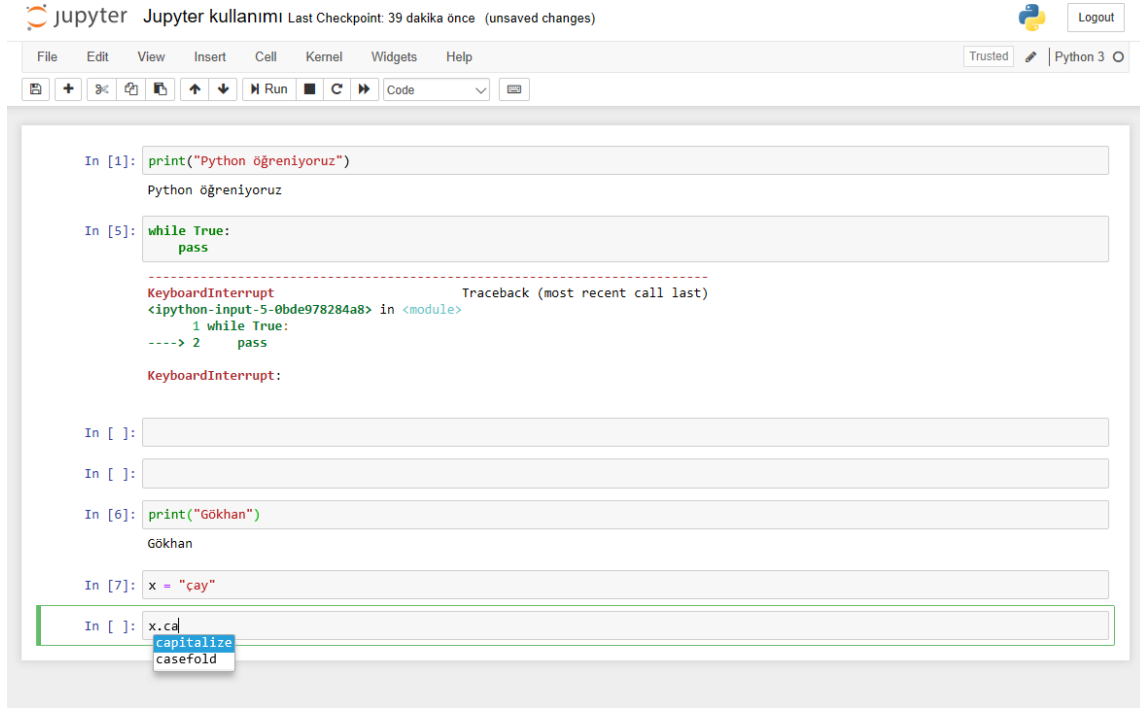


Jupyter notebookun bir diğer özelliği de aşağıdaki örnekte gösterilmiştir.

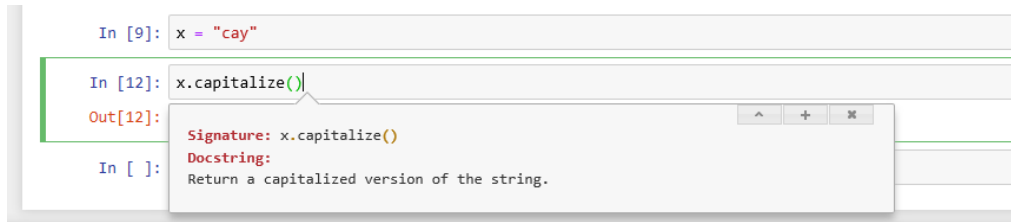


Bu örnekte bir **string (kelime)** değişkeni olarak **x** değişkeni tanımlanmıştır. Bu değişken ile işlem yapmak için bir sonraki hücrede **x.** yazdıktan sonra **TAB** tuşuna basılırsa **x** değişkeni üzerinde yapabileceğimiz işlemler küçük pencere açılarak görüntülenir. Değişken üzerinde nasıl bir işlem yapılmak isteniyorsa fare yardımıyla bu işlem açılan pencereden seçilebilir.

**x.ca** yazılıp **TAB** a basıldığında **ca** ile başlayan tüm işlem seçenekleri gösterilebilir.



Dahası **Shift+TAB** tuşlarına basıldığında da **capatilize** komutunun ne işe yaradığı görüntülenebilir.

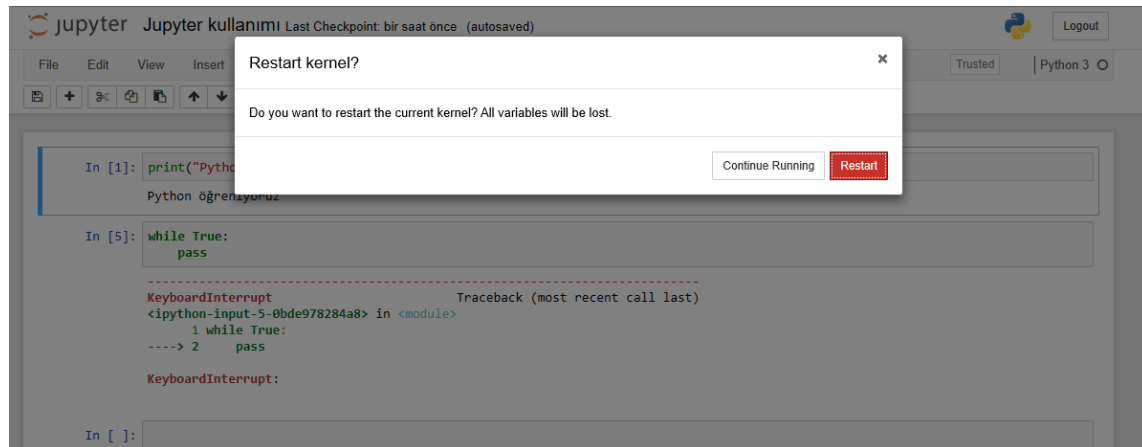
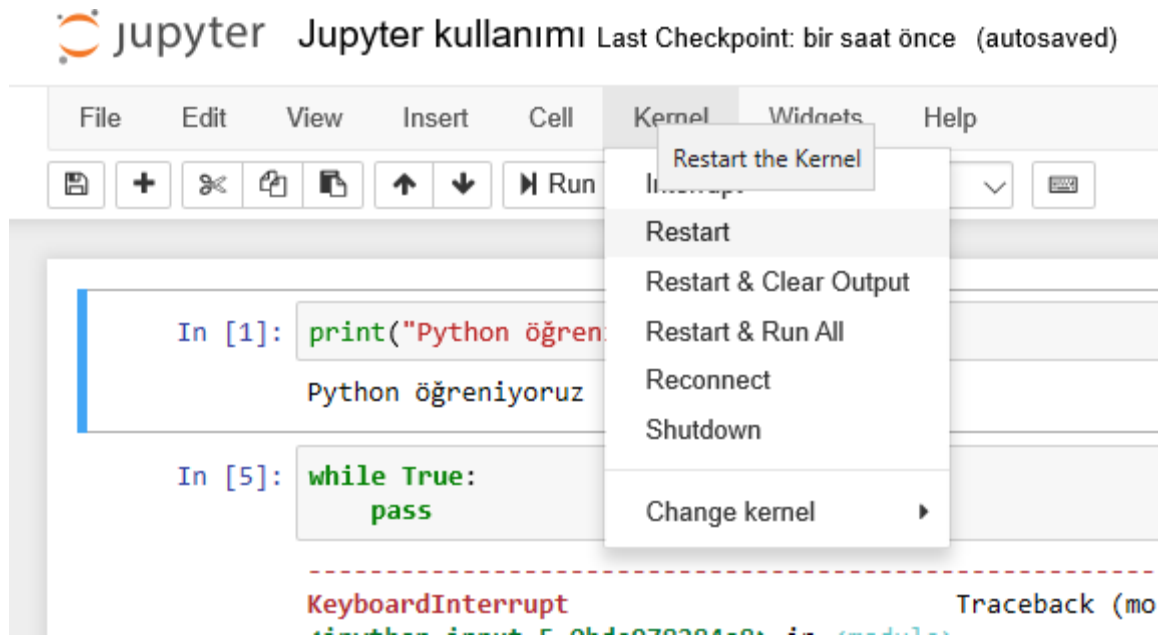


Anlaşılabacağı üzere, `capitalize` komutu, `x` string değişkeninin ilk harfini büyük yapmıştır.



Ayrıca, Python’da help şeklinde bir fonksiyon bulunmaktadır. Hücreye **help(x)** yazıldığında x olarak oluşturduğumuz **string** değişkenin bütün özellikleri dökülür. Bu işlem herhangi bir veri tipi içinde yapılabilir. **help(x.count)** denilirse x string değişkenimizin count metodunun özelliklerini öğrenmek istersen **Shift+TAB** yapabiliriz. Aynı işlemi, hücreye **help(x.count)** yazarakta program çıktısı olarak alabiliriz. Böylece count fonksiyonun ne iş yaptığını görebiliriz.

Çalıştığımız dosyadaki komutları değil ancak diğer şeyleri temizlemek istersek bu durumda **Kernel** sekmesinde **restart** seçeneği seçilmelidir.



Bu işlem yapılırsa, her kod yani her hücre yeniden çalıştırılmalıdır, çünkü tüm değişkenler kaybedilecektir.