

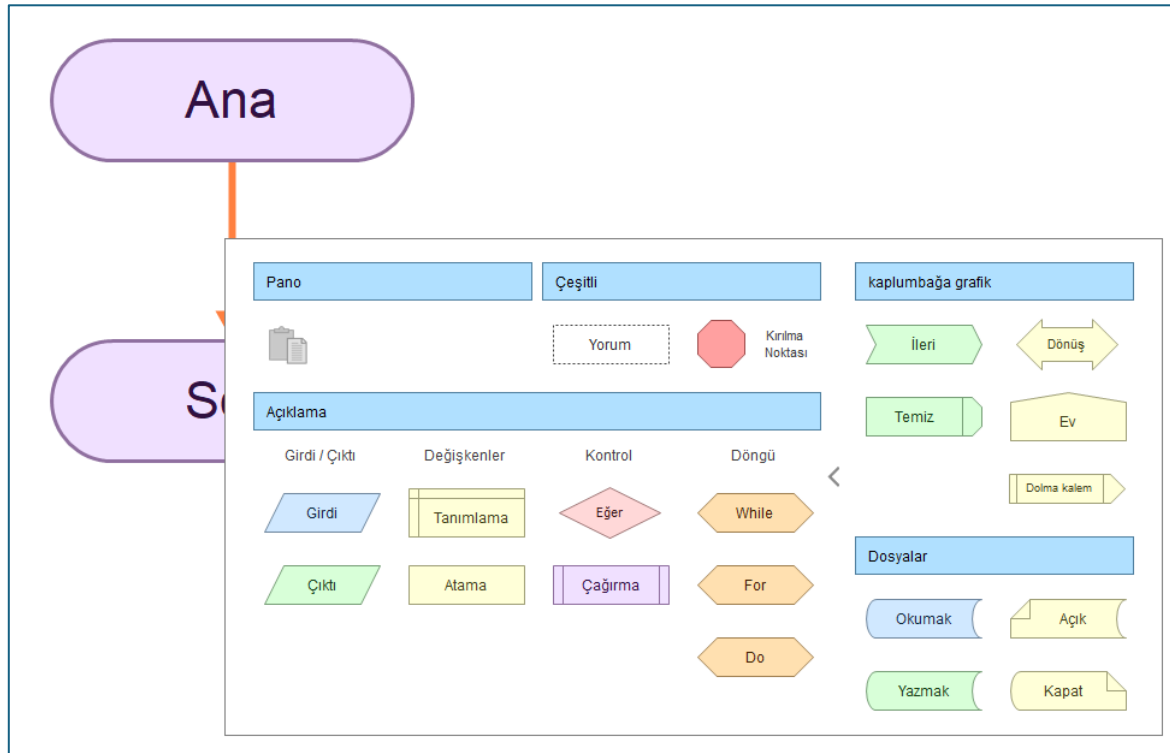
Bölüm 1. Programlamaya Giriş ve Anaconda Navigator Kurulumu

Algoritma

Herhangi bir sorunun çözümü için izlenecek yolun tarifi algoritma olarak adlandırılır. Çözüme ulaşmak amacıyla yapılması gereken aşamalar adım adım ve net şekilde belirtilir. Algoritma **Başla** ile başlatılır ve **Bitir** ile bitirilir.

Akış diyagramı

Algoritmanın görsel olarak simge ya da sembollerle ifade edilmesidir. Akış diyagramı hazırlamak için [flowgrothim](#) isimli ücretsiz uygulama kullanılabilir.



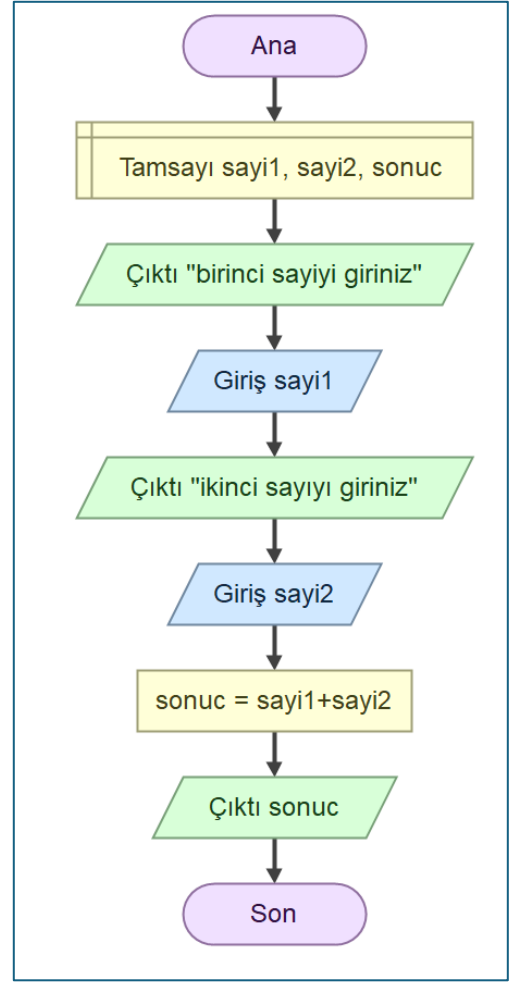
Şekil 1.1. *Flowgrothim* uygulamasının ekran görüntüsü

Her algoritma Başla ile başlanıp Bitir ile bitirildiğinden akış diyagramı da bu ifadelerle başlayıp biter. Flowgrothim uygulamasında Başla ve Bitir yerine **Ana** ve **Son** ifadeleri kullanılmıştır, bu ifadeler Başla ve Bitir ile aynı anlamdadır. Programın Türkçe dil desteği bulunmaktadır ancak bazı programlama dilleri Türkçe dilini desteklemediğinden değişken tanımlarında Türkçe karakter kullanılmaması tercih edilebilir.

Bölüm 1. Programlamaya Giriş ve Anaconda Navigator Kurulumu

Örnek: İki sayının toplamını bulan programın algoritmasını ve akış diyagramını hazırlayınız.

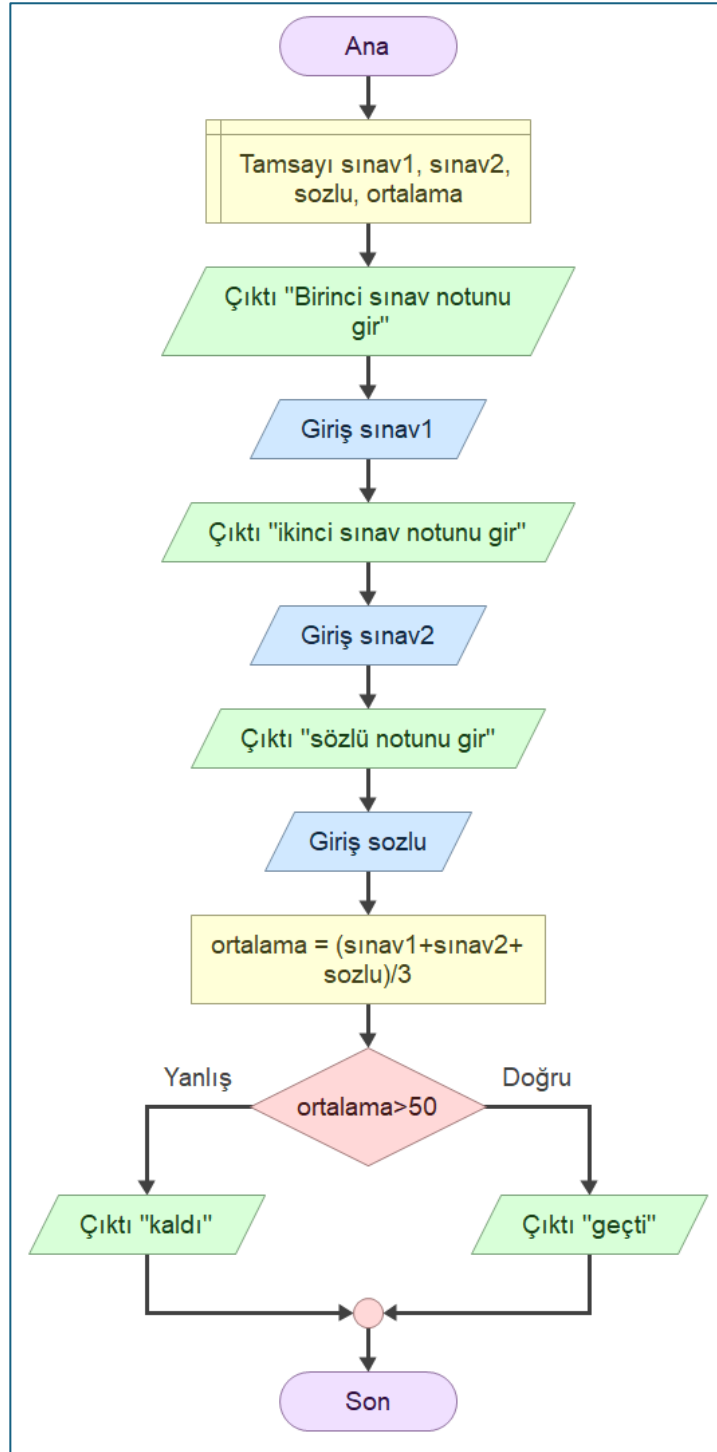
1. Başla
2. Birinci sayıyı giriniz
3. İkinci sayıyı giriniz
4. 1. ve 2. sayının toplamını hesapla
5. Sonucu yazdır
6. Bitir



Bölüm 1. Programlamaya Giriş ve Anaconda Navigator Kurulumu

Örnek: 2 adet yazılı 1 adet sözlü sınavın ortalamasını hesaplayarak ortalama değer 50 değerinden büyükse ekrana “geçer” yazdıran küçükse “kaldı” yazan programın algoritmasını hazırlayınız.

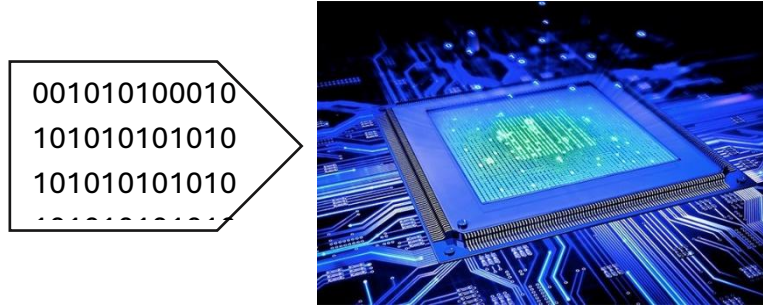
1. Başla
2. Birinci sınav notunu gir
3. İkinci sınav notunu gir
4. Sözlü notunu gir
5. Ortalamayı hesapla
6. Eğer ortalama 50’den büyükse 7. adıma git değilse 8. adıma git
7. Ekrana “geçti” yaz
8. Ekrana “kaldı” yaz
9. Bitir



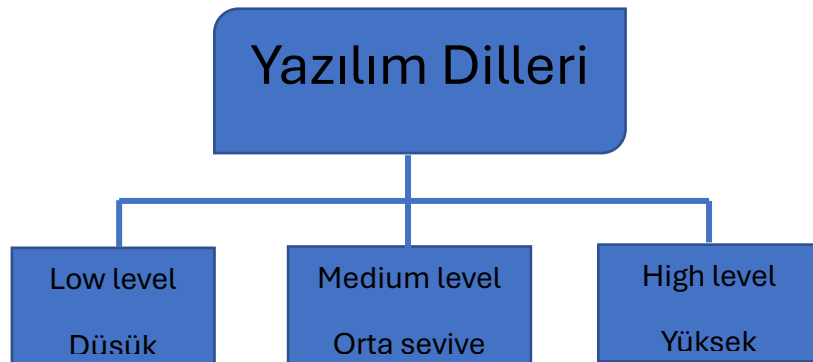
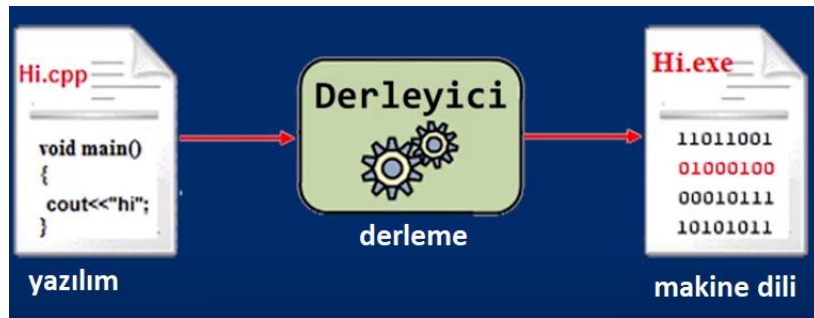
Bölüm 1. Programlamaya Giriş ve Anaconda Navigator Kurulumu

Yazılım Dillerine Genel Bakış

Mikroişlemcisi olan cep telefonu, bilgisayar gibi aygıtlar sadece 0 ve 1' ler ile yani Makine dili ile çalışır. Mikroişlemcinin bu komutları anlayabilmesi için komutlar kodlanmalıdır.



Yazılım dilleri, makine dili kodlarını kolaylık olsun diye semboller ve yazılım diline özgü gramer ile ifade etme aracıdır. Bu sembol ve gramerin, mikroişlemcinin anlayacağı 0 ve 1' lere dönüştürülmesi gerekir. Bu dönüştürme işlemine **derleme** adı verilir.



Düşük seviye yazılım dillerinde, makine dili sembollerle ifade edilir ve Assembly dili olarak adlandırılır. Bu dilde yazılan kodlar uzundur ve derlendikten sonra daha kısa bir kod bütününe dönüşür. Assembly dili etkin bir dildir fakat insan dil yapısına benzemediğinden hem bu dili öğrenmek hem de bu dilde kodlama yapmak zordur. Bu nedenle de Orta seviye yazılım dilleri oluşturulmuştur.

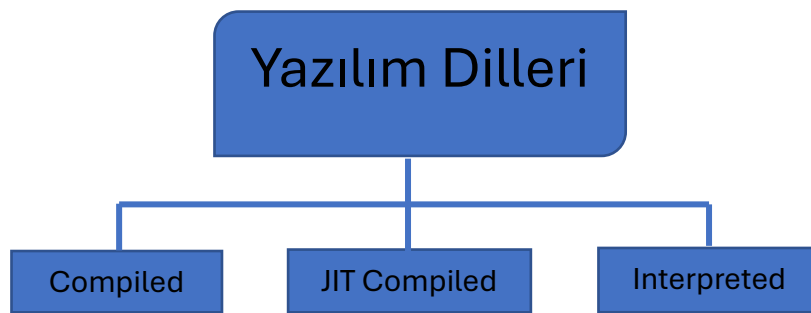
Orta seviye yazılım dilleri, insan dil yapısına daha yakın olup, düşük seviye dilinin de özelliklerini bünyesinde barındırır. Bu nedenle hem insan diline hem de makine diline yakındır.

Bölüm 1. Programlamaya Giriş ve Anaconda Navigator Kurulumu

Orta seviye dillerin derlenmesi ile makine diline dönüştürülen kodlar, düşük seviye diller kadar etkili olmasa da yine de performansları iyidir. Orta seviye dillere örnek olarak C dili veya C++ verilebilir.

Yüksek seviye yazılım dilleri, insan diline yakındır. Bu nedenle öğrenmesi ve kodlama süreci çok kısadır. Bu yazılım dillerinde çok sayıda yazılım kütüphanesi bulunur. Bu kütüphaneler sayesinde yazılımcı uzun uzun kodlar kullanmadan bu kütüphanelerden faydalanır. Dahası farklı kütüphanelerdeki özellikleri amaca uygun olarak farklı kombinasyonlarda kullanarak etkin kodlar oluşturulabilir. Örnek olarak **C#**, **Python** ve **Java** verilebilir.

Yazılım dilleri, çalışmaları açısından **Compiled** (derlenebilen), **JIT-Just in Time Compiled** (Zamanında derlenen) ve **Interpreted** (Yorumlanan) yazılım dilleri olarak da üçe ayrılır.



Derlenen dillere örnek olarak **C**, **C++** ve **Pascal** verilebilir.

Zamanında derlenen yazılım dillerinde bir veya daha fazla sanal makine vardır. Bu dillere örnek olarak **C#** ve **Java** verilebilir.

Yorumlanan yazılım dillerinde, yazılımcının yazdığı kod, satır satır okunup, yorumlanıp çalıştırılır. Bu nedenle bu dil grubu, bir yorumlayıcıya ihtiyaç duyar. Yorumlanan dillere en iyi örnek **Python**’ dur.



NOT: “En iyi yazılım dili” diye bir durum yoktur. Her yazılım dili yerinde kullanıldığı zaman en iyi yazılımdır.

Bölüm 1. Programlamaya Giriş ve Anaconda Navigator Kurulumu

Python Programlama Dili

Python (*paytın* olarak okunur), nesne yönelimli, yorumlamalı, modüler ve etkileşimli yüksek seviye programlama dilidir. Diğer programlama dillerinden en önemli farkı sözdiziminin basit ve bu nedenle de akılda kalıcı ve daha kısa kodlar ile aynı işlemin yapılabilir olmasıdır.

Geliştirilmeye 1990 yılında **Guido van Rossum** tarafından Amsterdam'da başlanmıştır. Adını sanılanın aksine bir yılından değil Guido van Rossum'un çok sevdiği, **Monty Python** adlı altı kişilik bir İngiliz komedi grubunun **Monty Python's Flying Circus** adlı gösterisinden almıştır. Günümüzde Python Yazılım Vakfı çevresinde toplanan gönüllülerin çabalarıyla sürdürülmektedir. Python 1.0 sürümüne Ocak 1994'te ulaşmıştır. Son kararlı sürümü, 2.x serisinde Python 2.7 ve 3.x serisinde Python 3.5.2'dir. 3 Aralık 2008 tarihinden itibaren 3.x serisi yayınlanmaya başlamıştır; ancak 3.x serisi ile 2.x serisi uyumlu değildir.

- İlgilenen öğrenciler için ek Türkçe kaynak: [Python Programlama Dili- Python 3 Türkçe Kaynak](#)
- [Python resmî sitesi](#)

Python Programlama Dili

Python dili, öğrenilmesi kolay ve güçlü bir programlama dili olması yanında;

- Kolay ve net okunabilen yazımı
- Dinamik veri yapısı
- Güçlü ifade yeteneği
- Modüler yapısı ve geniş kütüphanesi
- Nesne tabanlı programlamaya uygun olması
- Exeption tabanlı hata yönetimi
- C ve C++ ile ek modüller yazma kolaylığı
- Diğer programlara kodlama arayüzü olarak dahil edilebilmesi

özelliklerine sahiptir. Python dili;

- Flask ve Django framework' leri ile web geliştirme
- Ağ ve socket programcılığı
- Arayüz geliştirme
- Requests, Beautiful Soup ya da Scrapy ile örümcek türü yazılımlar geliştirmede
- Scikit-learn ve Numpy modülleri ile Machine Learning (Makine öğrenmesi) ve Yapay Zeka çalışmalarında kullanılır.

Bölüm 1. Programlamaya Giriş ve Anaconda Navigator Kurulumu

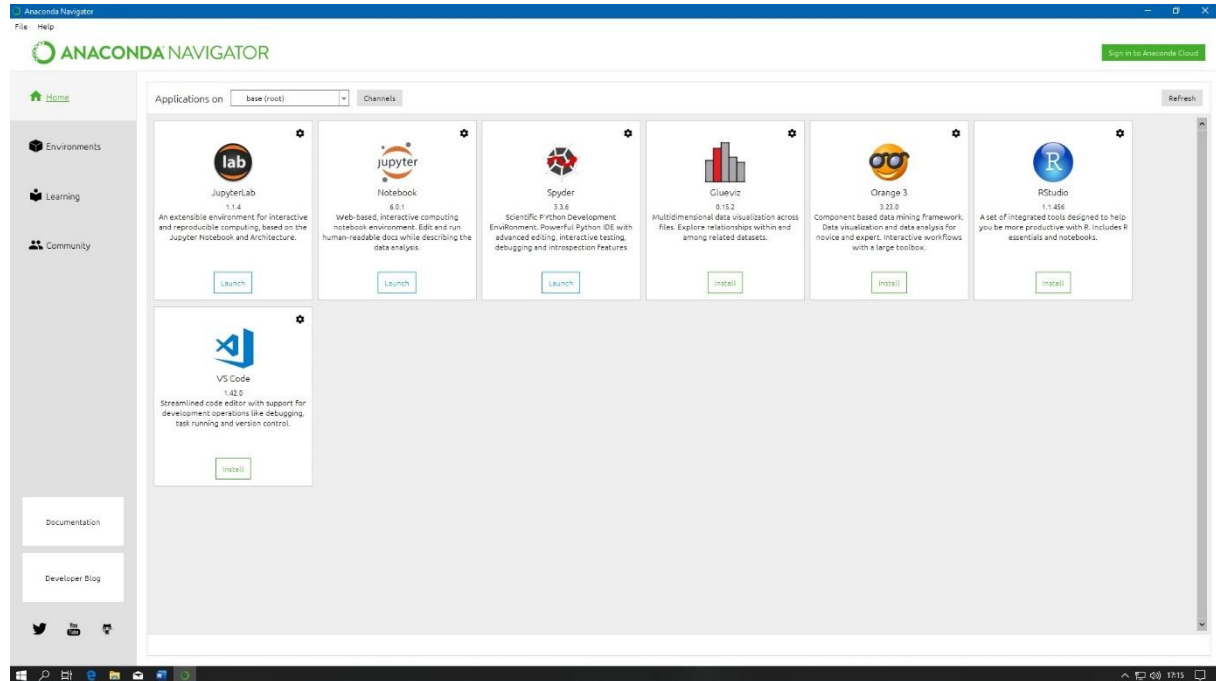
Python için Gerekli Ortamların Kurulması

[Anaconda](https://www.anaconda.com/distribution/) (<https://www.anaconda.com/distribution/>) adresinden **Anaconda Distribution** programı indirilip kurulmalıdır. Anaconda dağıtımının özelliği, bilimsel uygulamalar ve makine öğrenmesi için gerekli olan numpy, pandas, matplotlib gibi kütüphaneleri içermesidir.

Windows için **Ayarlar → Sistem → Hakkında** kısmına girerek ya da masaüstünde bulunan **Bu Bilgisayar'** a sağ tıklayıp, özellikler seçilerek bilgisayarınızın bit sayısını öğrenebilirsiniz. Bilgisayarınızın bit sayısına göre 64-bit ya da 32-bit olarak indirilmelidir.

Anaconda yüklenirken *Next, I Agree, Just me ya da All users* (bu seçim kullanıcıya aittir) seçenekleri ile Anaconda sisteme yüklenir. Bu kurulum ile **Jupyter Notebook**, **Phyton ara yüzü** ve **Python yorumlayıcısı** bilgisayarınıza yüklenmiş olacaktır.

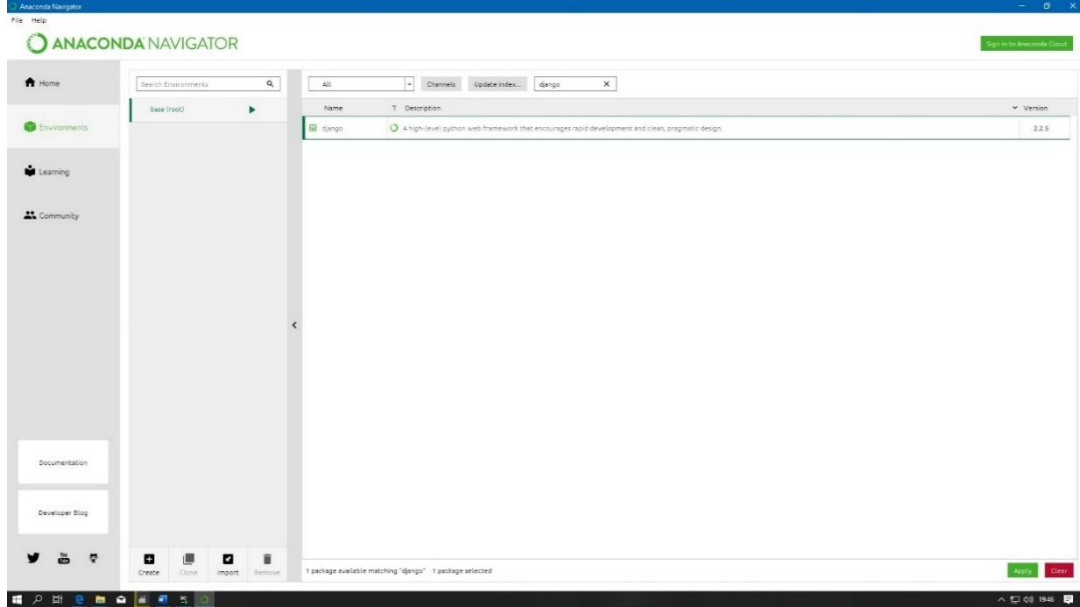
Windows' un arama bölümüne Anaconda Navigator yazılarak Anaconda başlatılır.



Anaconda açıldığında, Anaconda ile birlikte kurulan paketlerin görüntülediği yukarıdaki pencere açılır. Anaconda'yı açmanın bir diğer yolu da Anaconda'nın kurulduğu klasöre giderek Scripts klasöründen **Anaconda Navigator.exe** dosyasını çift tıklamaktır.

Bölüm 1. Programlamaya Giriş ve Anaconda Navigator Kurulumu

Anaconda Navigator'ün Environments sekmesine girilirse, Anaconda ile birlikte gelen Python paketleri görüntülenebilir. Bu bölümde daha sonra kullanılacak birçok uygulama halihazırda yüklenmiş olur. Ancak Django yüklü değildir. Django'yu yüklemek için, Anaconda Navigator'de Environments'a girilir. Sağ taraftaki bölümde **All** seçilerek arama (search) bölümüne Django yazılarak arama yapılır. Django'nun yüklü olmadığı görülür. Yüklemek için tike yazılarak sağ alt bölümde yer alan Apply (uygula) butonuna basılır ve Django'nun yüklenerek environments'a eklenmesi sağlanır.



Jupyter Notebook'u çalıştırmak için Anaconda Navigator Home penceresi açıkken Jupyter Notebook üzerindeki Launch butonuna basılarak Jupyter Notebook açılmış olur.

Jupyter Notebook hem çalışma esnasında hatırlatma amaçlı not alınabilen hem de Python kodlarımızı yazacağımız bir notebook olup varsayılan web ara yüzü ile çalışan bir notebook'tur. ***Jupyter notebook çalıştırıldığında bir de terminal açılır ve bu terminal kapatılmamalıdır!***

Jupyter notebook' u açmanın bir diğer yolu da Windows aramasında **cmd** yazılarak açılan ms-dos terminalinde **Jupyter Notebook** yazmaktır.

Bölüm 1. Programlamaya Giriş ve Anaconda Navigator Kurulumu

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 terminal açılır. *Bu terminalin, Jupyter notebook ile yapılan çalışma tamamlanıncaya kadar kapatılmaması gerekmektedir!*

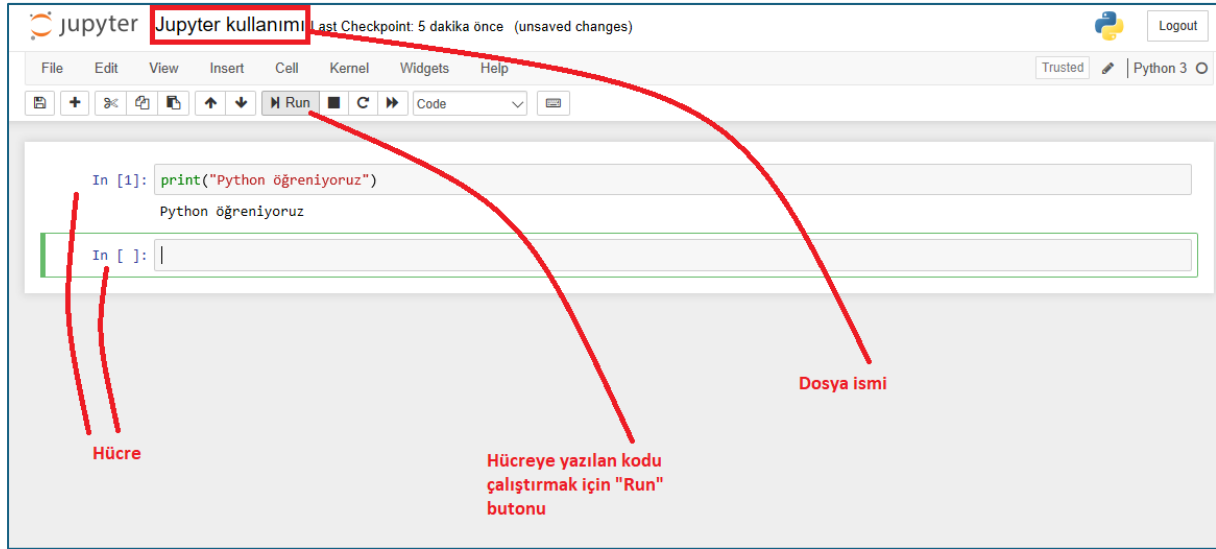
Jupyter Notebook açıldığında **C:\Users\kullanıcı ismi>** olarak açılır ve adres satırında da **http://localhost:8888/tree** olarak görünür. Eğer web arayüzü ile Jupyter notebook açılmazsa, Jupyter notebooku açmanın bir üçüncü yolu da, serverda görünen adresi **Ctrl+C** ile kopyalayıp web arayüzündeki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

Bölüm 1. Programlamaya Giriş ve Anaconda Navigator Kurulumu

Basit bir örnek aşağıda verilmiştir.

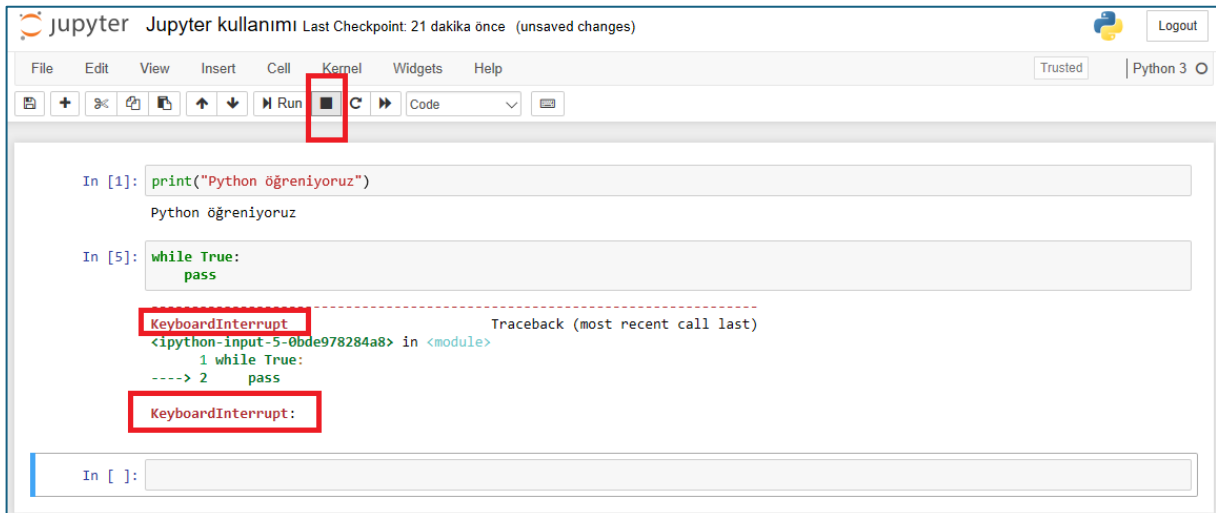


In [] yazan bölümler hücre olarak adlandırılmaktadır. Uygun formatta yazılan komutu çalıştırmak için Run Cell (Hücreyi çalıştır) anlamında şekilde gösterilen butona basılır. Bu basit uygulamada

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 çalıştırıldıktan 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.

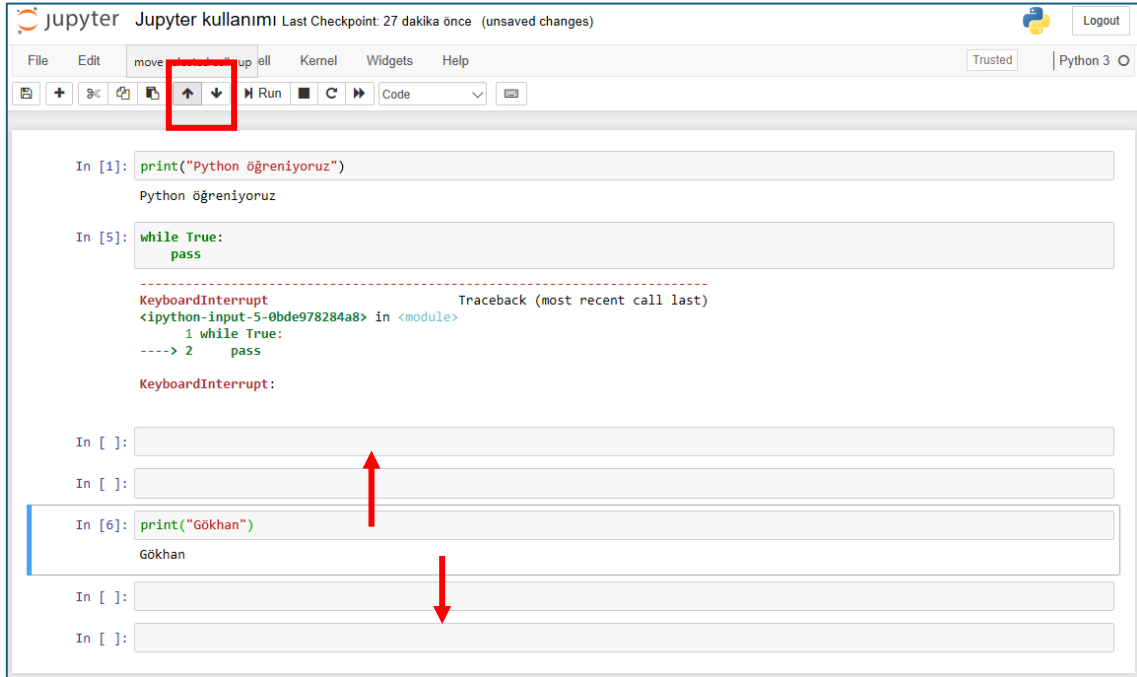
Bazı durumlarda program kısır döngüye girebilir veya programın sonuç vermesi uzun sürebilir. Bu tip durumlarda Run düğmesinin yanındaki **Interrupt** (*kesmek, yarıda kesmek*) düğmesine basılabilir.



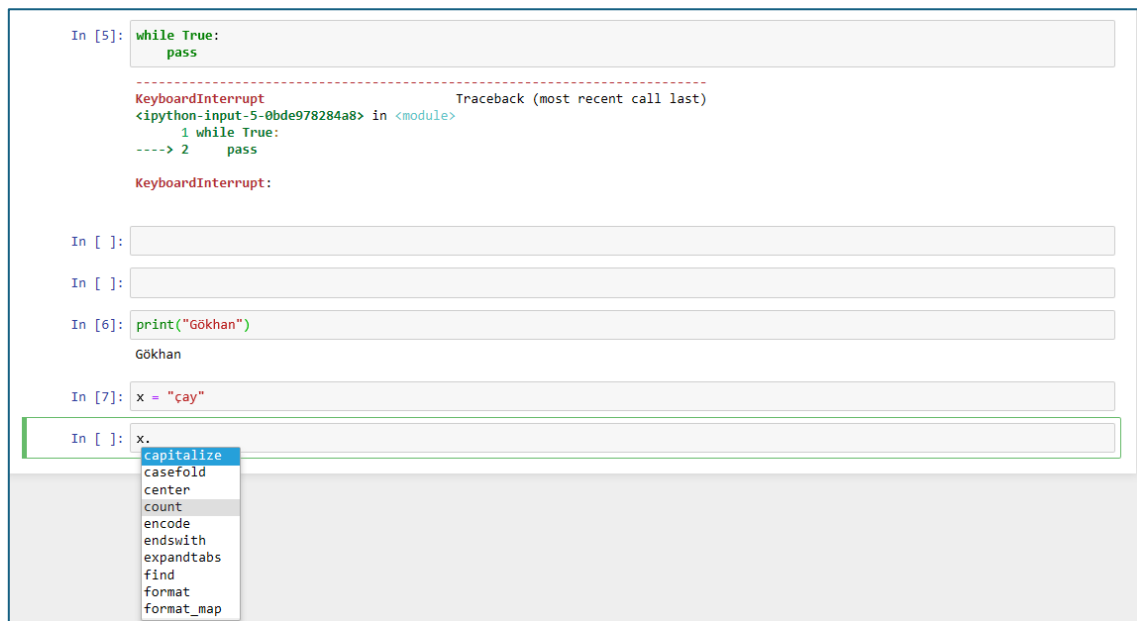
Bölüm 1. Programlamaya Giriş ve Anaconda Navigator Kurulumu

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 de 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.



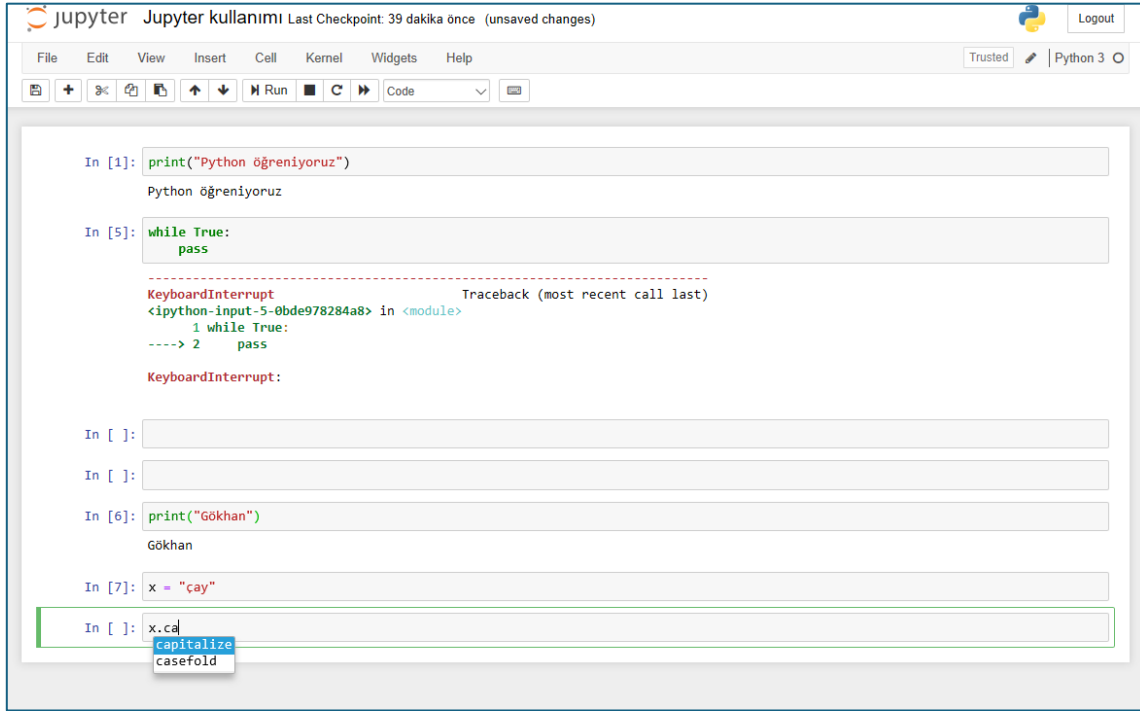
Jupyter notebookun bir diğer özelliği de aşağıdaki örnekte gösterilmiştir.



Bölüm 1. Programlamaya Giriş ve Anaconda Navigator Kurulumu

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.



```
In [1]: print("Python öğreniyoruz")
Python öğreniyoruz

In [5]: while True:
        pass

KeyboardInterrupt                                Traceback (most recent call last)
<ipython-input-5-0bde978284a8> in <module>
      1 while True:
----> 2     pass

KeyboardInterrupt:

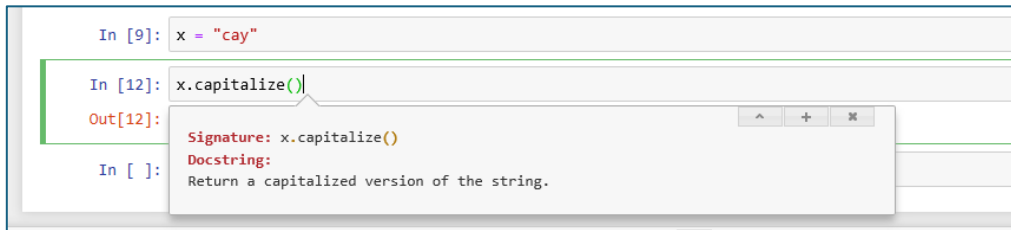
In [ ]:

In [ ]:

In [6]: print("Gökhan")
Gökhan

In [7]: x = "çay"

In [ ]: x.ca
        capitalize
        casefold
```

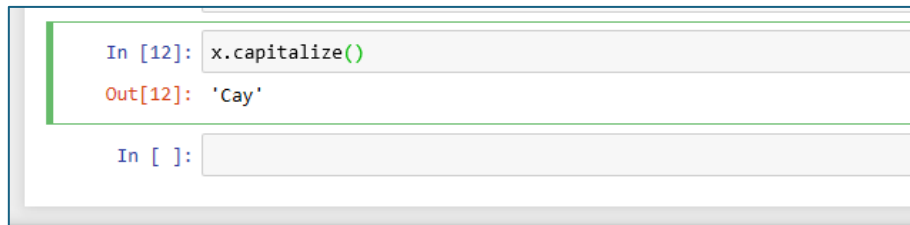


```
In [9]: x = "çay"

In [12]: x.capitalize()
Out[12]:

Signature: x.capitalize()
Docstring:
Return a capitalized version of the string.
```

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



```
In [12]: x.capitalize()
Out[12]: 'Cay'

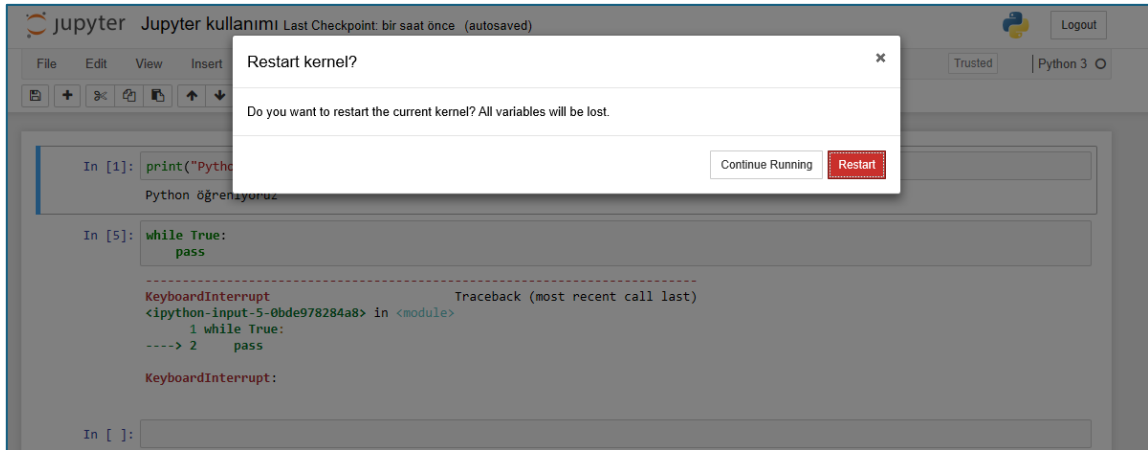
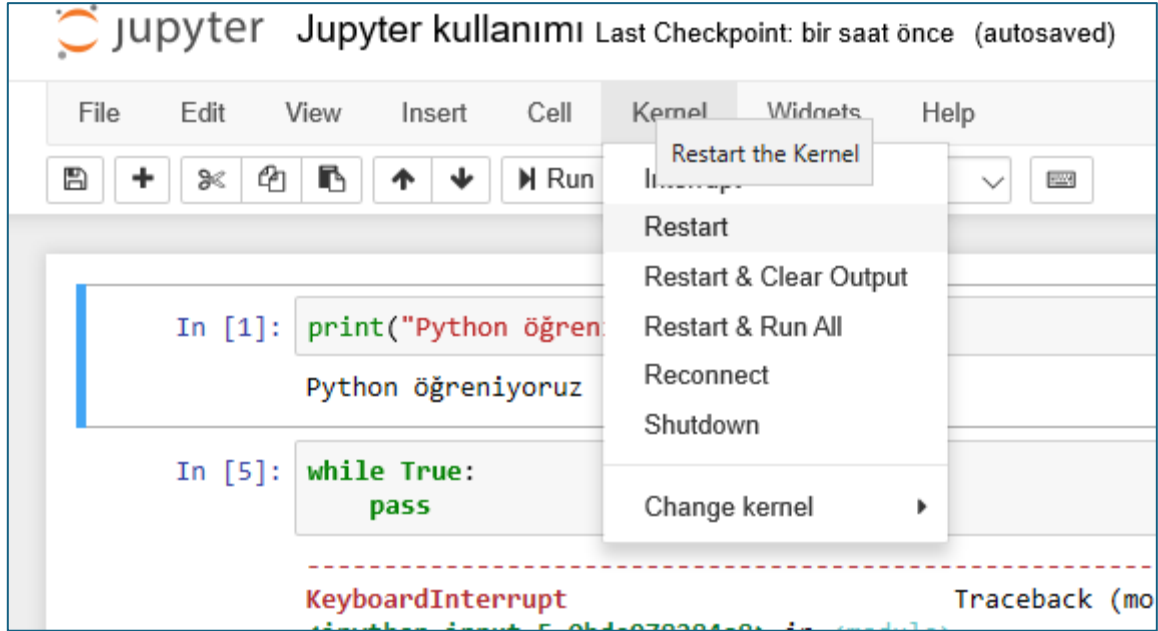
In [ ]:
```

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)**

Bölüm 1. Programlamaya Giriş ve Anaconda Navigator Kurulumu

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.



Bölüm 1. Programlamaya Giriş ve Anaconda Navigator Kurulumu

Python Temel Kavramlar

Sayılar, Değişken Tanımlama, Temel Matematik Operatörler

Python’ da sayılar: Tam sayılar (integer) ve ondalıklı sayılar (float) Python’ da bir veri tipidir.

Temel Matematik operatörler: Toplama (+), Çıkarma (-), Çarpma (*) ve Bölme (/)

Değişken tanımlama: Değişken, bir veri tipinden (sayısal (numeric) ya da sözcük (string)) değer tutan birimlerdir ya da objelerdir. Sayısal bir değişken, **değişkenin ismi = sayı değeri** olarak tanımlanır.

Değişken ismi verilirken aşağıdaki durumlara dikkat edilmelidir.

- Değişken ismi sayı ile başlayamaz.
- Değişken ismi, isim tamlaması gibi ise kelimeler arasında boşluk bulunamaz.
- :", <, >, /, ? \, !, @, #, \$, %, ^, &, *, ~, -, + simgeleri değişken isminde kullanılamaz. Sadece alt çizgi _ kullanılabilir.
- Python’ da tanımlı **while**, **not** gibi özel kelimeler değişken ismi olarak kullanılamaz.

Aslında bu kurallar tüm bilgisayar programlarında geçerlidir.

Python’da veri tipi dinamiktir. ***Yani tamsayı değeri verilmiş bir değişken programın işleyişinde örneğin ondalıklı bir sayı olarak dinamik olarak güncellenebilir.***

Not: Bölme işlemi sonucunda Python’un özelliği olarak sonuçlar float yani ondalıklı olarak görüntülenir. Bu özellik Python 3 ile gelen ve sonuçların daha kesin gösterilmesi için konulmuş bir özelliktir.

Python’ da değişken isminde büyük küçük harf duyarlılığı bulunmaktadır. Yani x ve X farklı değişken isimleri olarak kullanılabilir.

Bir değişkenin değerini bir artırmak için (ki bu işlem döngülerde çok önemlidir) +1 yapılabilir. Ancak Python’ da daha kolay bir yöntem +=1 tanımıdır. Bu tanım ile değişkenin değeri bir artırılır ve yeni değişken değeri olarak atanır. Sayı iki artırılmak istenirse de +=2 yazılabilir vb.

Bir değişkenin bir sayı ile çarpılıp çarpımın yeni değişken değeri atanması için de *=sayı komutu kullanılabilir. Örneğin x=5 ise ve yeni değeri **x=x*4** ise bu, **x*=4** olarak yazılabilir.

Ya da bir x değişkenin kendi değerinden örneğin 3 sayısı çıkarılacaksa, **x-=3** komut satırı kullanılabilir.

Yorum satırları

işaretinden sonra (Alt Gr+3) tek satırlık yorum satırı eklenebilir.

tekli yorum satırı

Çoklu yorum satırları için de

“““

Çoklu yorum satırı

”””

özellik kullanılır. Yorum satırları Python tarafından işleme alınmaz.

Bölüm 1. Programlamaya Giriş ve Anaconda Navigator Kurulumu

Matematik Operatörleri

// sembolü tamsayı bölmesidir.

% sembolü de bölümden kalanı göstermektedir.

Üs bulma işlemi için iki tane ** işareti peş peşe kullanılır.

Köklü sayılar için de örneğin karekök için $**(0.5)$ yazılarak sayının karekökü belirlenir.

Python'da veri yapıları aşağıda verildiği gibidir ve amaca uygun olarak bu veri tiplerinden biri ya da bir kaçı ile çalışmak gerekir.

