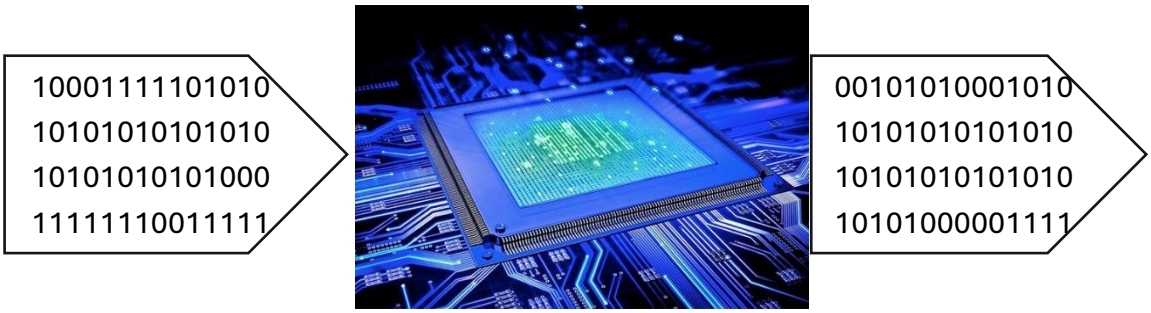
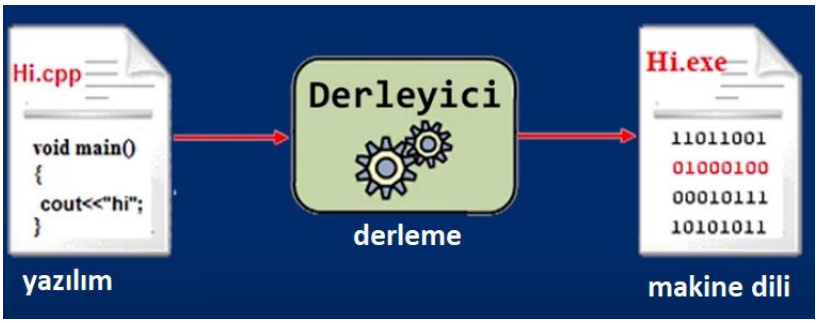
Giriş

Mikroişlemci (CPU – Central Processing Unit) bir mikrodenetleyici ya da bilgisayar sisteminin beyni olarak düşünülebilir. Buna karşın bir mikrodenetleyici giriş çıkış elemanları, hafıza gibi elemanlar içeren özel bir amaç için tasarlanmış küçük bir bilgisayardır.

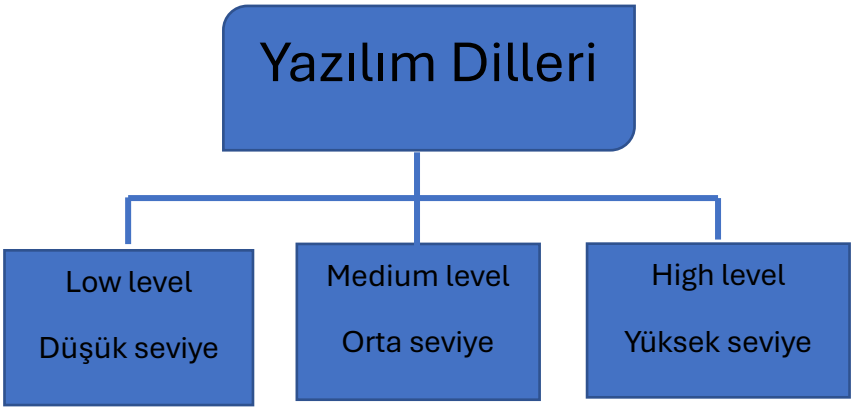
Bu sistemler transistörlerden oluşur ve transistörün iletimde olduğu durum mantık 1 durumudur ve kesimde olduğu durum da mantık 0 durumudur. Dolayısıyla mikrodenetleyici ve bilgisayar sistemleri mantık 0 ve mantık 1 durumları ile programlanır. Mikroişlemci de bu 0 ve 1' ler ile sistemi yönetir.



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



Yazılım dilleri ***Low level*** (düşük seviye), ***Medium level*** (orta seviye), ***High level*** (yüksek seviye) olmak üzere üçe ayrılır.



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 çok zordur. Bu nedenle de Orta seviye yazılım dilleri geliştirilmiştir.

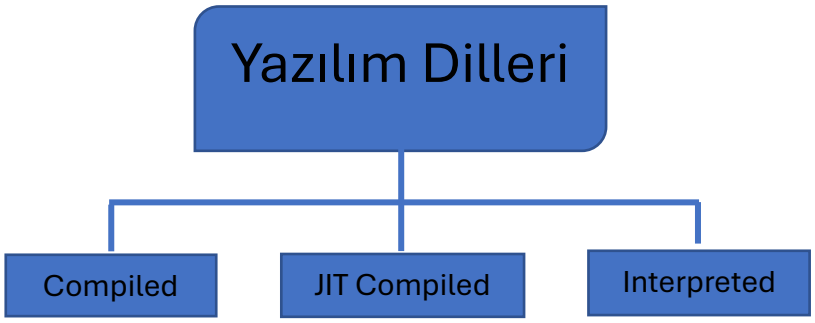
Orta seviye yazılım dilleri, insan dil yapısına daha yakın olup, düşük seviye dilinin de özelliklerini bünyesinde barındırır. Bu nedenle hem insan diline hem de makine diline yakındır. Orta seviye dillerin derlenmesi ile makine diline dönüştürülen kodlar, düşük seviye diller kadar etkili olmasa da yine de performansları iyidir. Orta seviye dillere örnek olarak C dili veya C++ verilebilir. Orta seviye diller, düşük seviye dillinin özelliklerini de içerdiğinden, eğer yazılımcı mikrodenetleyicinin mimarisini bilmiyorsa geliştirme sırasında kodlamada ciddi hatalar oluşabilir. Bu nedenle orta seviye dilleri öğrenmek zor olduğu gibi kodlama süresi de uzun olur. Bu sorunun üstesinden gelebilmek için de yüksek seviye diller geliştirilmiştir.

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 kullanamadan bu kütüphanelerden faydalanır. Dahası farklı kütüphanelerdeki özellikleri amaca uygun olarak farklı kombinasyonlarda kullanarak etkin kodlar oluşturulabilir.

Yüksek seviye yazılım dillerinde, programcının makine mimarisini bilmesine gerek yoktur, kütüphaneler bu süreci kendisi halleder. Gerek kütüphane mekanizması gerek insan diline yakın olması nedeniyle yüksek seviye yazılım dillerinde program yazmak son derece hızlı olarak gerçekleştirilebilir.

Ancak yüksek seviye yazılım dillerinde, yazılım derlendikten sonra kullandığı kütüphanelerden dolayı çıkan makine dili kodlarının sayısı fazla olur ve bu nedenle yüksek seviye yazılım dillerinin performansı düşüktür. Ancak günümüzde mikroişlemcilerin, işlem gücü yüksek olduğundan yazılımın düşük performansını mikroişlemcinin işlem gücü tolere edebilmektedir. Yüksek seviye yazılım dillerine örnek olarak **C#**, **Java** ve **Python** verilebilir.

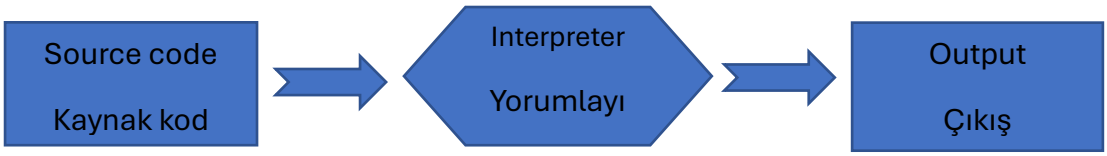
Yazılım dilleri, çalışmaları açısından **Compiled** (Derlenebilen), **JIT – Just in Time Compiled** (Zamanında derlenen) ve **Interpreted** (Yorumlanan) yazılım dilleri olarak da üçe ayrılır.



Compiled yani derlenebilen yazılım dillerinde yazılan kod, makine diline dönüştürülür. Bu nedenle en performanslı dillerdir. Ancak kodlanan bir kod, farklı bir mikroişlemcide kullanılacaksa bu durumda farklı derleyiciler kullanılması gerekir. Bu sebeple yazılan kodda değişiklikler yapılması gerekebilir. Derlenen dillere örnek olarak ***C***, ***C++*** ve ***Pascal*** verilebilir.

Zamanında derlenen yazılım dillerinde bir veya daha fazla sanal makine vardır. Yazılan kod, önce **Byte Code** adı verilen bir ara formata dönüştürülür. Bu Byte Code' lar sanal makine üzerinde çalışır yani Byte Code, bir sanal makinenin makine dilidir. Dolayısıyla programcı makine dili yerine bu sanal makinede çalışacak kodu yazar. Sanal makineler Byte Code' u anlayabildiği için gerekli gördüğü kod bloğunu ihtiyaç duyduğu anda mikroişlemcinin makine diline derleyerek çalıştırırlar. Böylece programcı, kodu, sanal makineye yazdığından mikroişlemci değişse bile kod veya derleyiciyi değiştirmesine gerek kalmaz. Zamanında derlenen yazılım dillerine en iyi örnekler ***C#*** ve ***Java***' dır.

Yorumlanan yazılım dillerinde, yazılımcının yazdığı kod, satır satır okunup, yorumlanıp çalıştırılır. Bu nedenle bu dil grubu, bir yorumlayıcıya ihtiyaç duyar.



Yorumlayıcılarda, makine dilini kullanan hazır kütüphaneler mevcuttur. Yorumlanan yazılım dillerinin en önemli özelliği derleyiciye ihtiyaç olmamasıdır. Yazılımcı yazdığı kodun sonucunu anında görebilir ve bir problem varsa anında müdahale edebilir. 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.

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 yılandan 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.

Python' un Kullanıldığı bazı uygulamalar

Django (**jango** olarak okunur) Python Programlama Dili için hazırlanmış ve BSD lisansı ile lisanslanmış yüksek seviyeli bir web çatısıdır. Basit kurulumu ve kullanımı, detaylı hata raporu sayfaları ve sunduğu yeni arayüz kodlama yöntemleriyle diğer sunucu yazılımı ve çatılardan kendini ayırmaktadır. İsmi, caz gitaristi Django Reinhardt'tan gelmektedir, **Zope** uygulama sunucuları, Python programlama dili ile yazılmış nesne yönelimli ve açık kaynak ağ uygulama sunucusudur. Açılımı Z Object Publishing Environment'tır. YouTube ve orijinal BitTorrent istemcisi, Python kullanan önemli projelerden bazılarıdır. Ayrıca Google, NASA ve CERN gibi büyük kurumlar da Python kullanmaktadır.

OpenOffice.org (*MS Office yazılımının alternatifi ve açık kaynak kodlu ücretsiz bir yazılımdır*),

GIMP (*GNU Image Manipulation Program - GNU Resim İşleme Programı, GNU Tasarısı dahilinde geliştirilen piksel tabanlı özgür ve ücretsiz bir görüntü işleme yazılımı. GIMP, Adobe Photoshop ve benzeri kapalı kaynak resim işleme araçlarına eşdeğer bir işlevler bütünü sunar. Linux, Windows, Mac OS gibi pek çok platformu destekler*),

Inkscape (*GNU Genel Kamu Lisansı ile dağıtılan, özgür ve ücretsiz bir vektörel grafik düzenleme yazılımıdır. Inkscape, logo ve afiş tasarımı ile çeşitli grafik işlemlerine olanak sağlamaktadır. Ayrıca yazılım ile var olan resimleri de düzenlemek mümkündür.),*

Blender (*Özgür bir üç boyutlu modelleme ve canlandırma uygulamasıdır. Blender tasarısı, 1998'de Hollanda'da kurulan NeoGeo adlı canlandırma stüdyosunun, kendi kadrosuyla yazılım geliştirerek kendi üretim araçlarına sahip olmayı hedeflemesiyle başlamıştır.)*

Scribus (*Özgür ve ücretsiz bir masaüstü yayıncılık yazılımı. PDF ve Postscript biçiminde dosyalar elde etmeye yönelik olarak ticari seviyede kullanılmaktadır. QuarkXPress ve InDesign gibi masaüstü yayıncılık araçlarına özgür ve ücretsiz bir seçenek sunar. Windows, Linux, Mac OS X başta olmak üzere birçok işletim sistemini destekler.)* ve

Paint Shop Pro (*Jasc Corporation'un çıkardığı resim düzenleme yazılımıdır. Bu yazılım Adobe şirketinin sahip olduğu Photoshop yazılımının rakibidir. Yazılım Corel şirketine transfer edilmiş olup 9. sürüm ve sonrasındaki sürümler bu şirket tarafından çıkarılmaktadır. Paint Shop Pro pek çok web master tarafından resim işleme yazılımı olarak kullanılmaktadır. Profesyonel ve yarı profesyonel çalışmalara uygundur.)* gibi bazı programlarda betik dili olarak kullanılır. Pek çok Linux dağıtımında ve Apple macOS işletim sisteminde Python ön tanımlı bir bileşen olarak gelmektedir.

- İlgilenen öğrenciler için ek Türkçe kaynak: Python Programlama Dili - Python 3 Türkçe Kaynak
- Python resmi 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
- Exception 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 soket 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 kullanılabilir.
- SciPy ve SymPy ile sembolik hesaplama yapabilme gücüne de sahiptir.

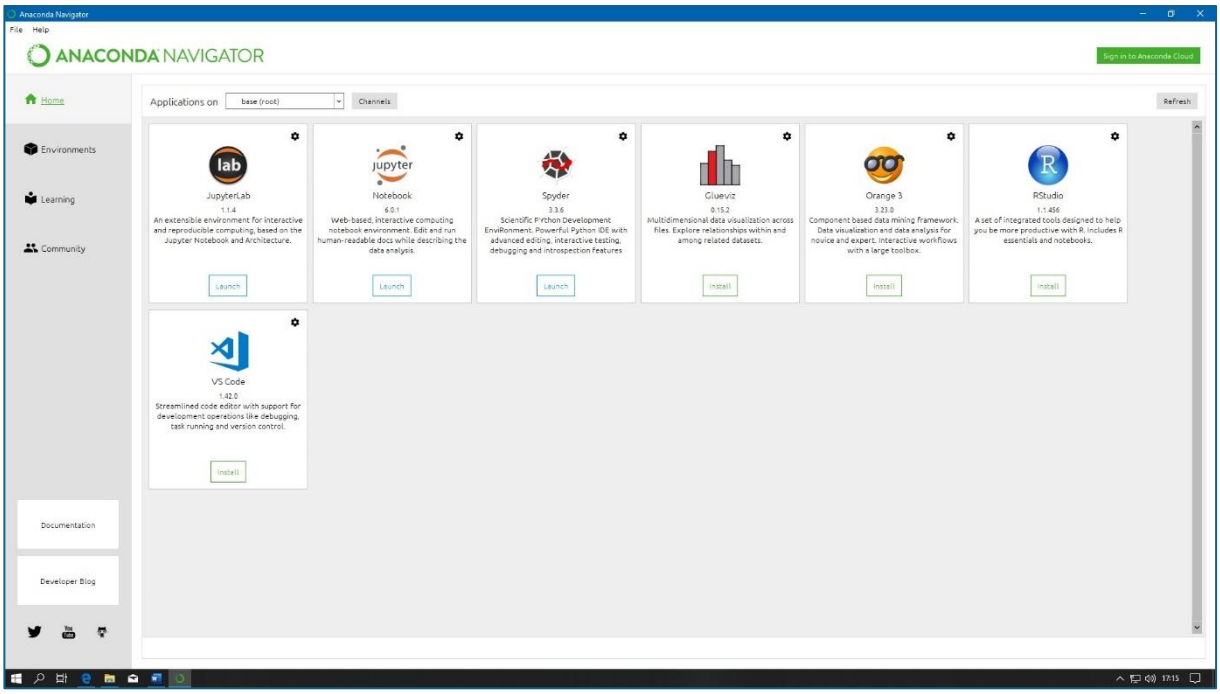
Python için Gerekli Ortamların Kurulması

Anaconda (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.

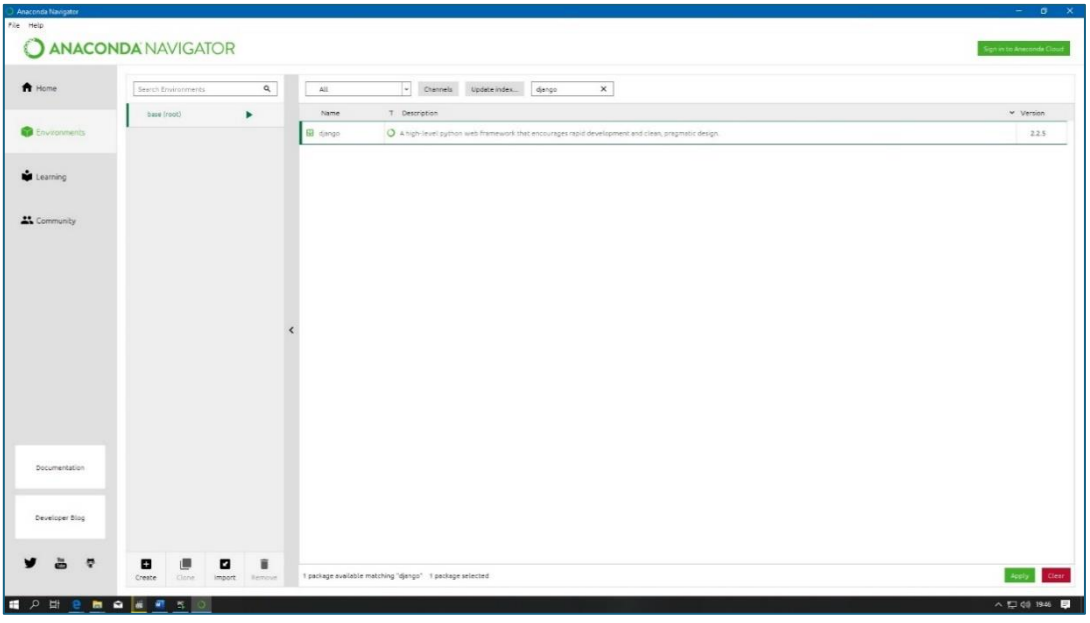
Anaconda yüklenirken ***Next***, ***I Agree***, ***Just me*** *ya da* ***All users*** ***(****bu seçim kullanıcıya aittir)* seçenekleri ile Anaconda sisteme yüklenir. Bu kurulum ile **Jupyter Notebook**, **Phyton ara yüzü** ve **Python yorumlayıcısı** bilgisayarınıza yüklenmiş olacaktır.

Windows' un arama bölümüne Anaconda Navigator yazılarak Anaconda başlatılır.



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

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 notebooouk'tur. ***Jupyter notebook çalıştırıldığında bir de terminal açılır ve bu terminal çalışma tamamlanana kadar kapatılmamalıdır!***

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

Jupyter Notebook Kullanımı

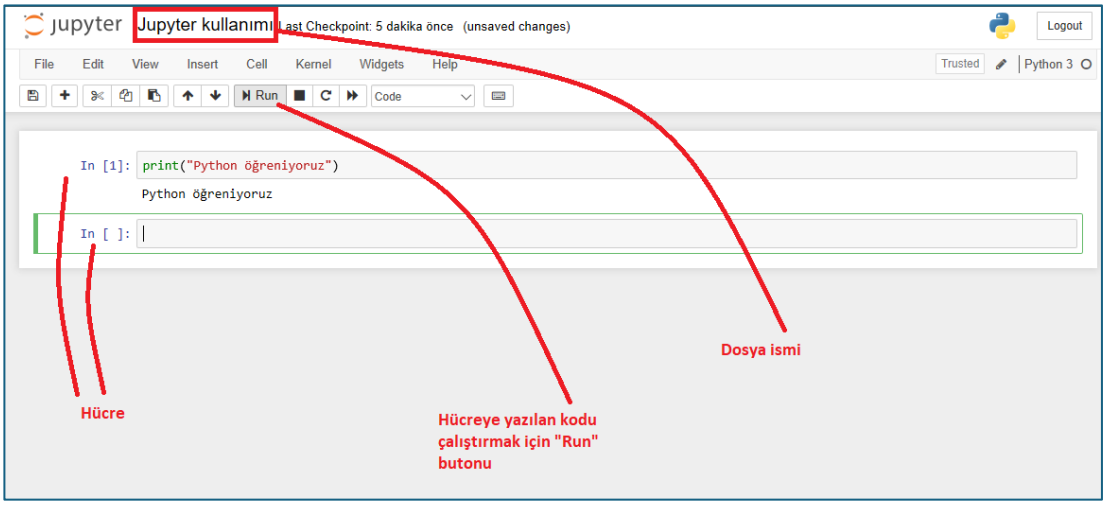
Jupyter notebook daha önce değinildiği üzere Anaconda ile gelen bir uygulamadır. Özelliği ise hem hatırlatma notları yazılabilmesi hem de Python ile kod yazılabilmesidir.

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 terminalde görünen adresi **Ctrl+C** ile kopyalayıp web arayüzündeki adres satırına bu adres yazılırsa Jupyter notebook yine açılacaktır.

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.

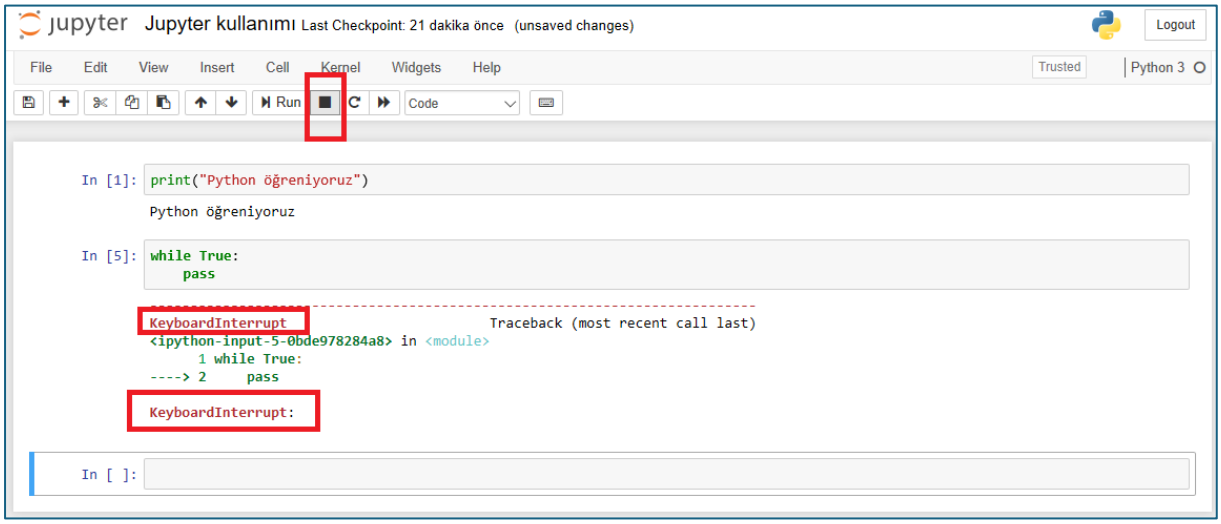


In [ ] yazan bölümler hücre olarak adlandırılmaktadır. Uygun formatta yazılan komutu çalıştırmak için ***Run Cell*** (Hücreyi çalıştır) anlamında şekilde gösterilen butona basılır. Bu basit uygulamada

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

Hücre çalıştırıldıktan 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. 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.

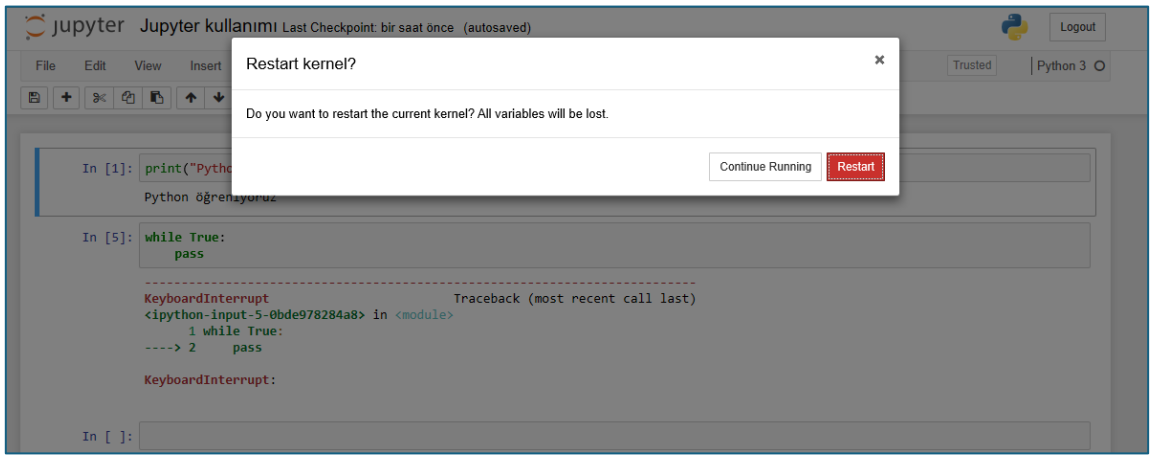
Jupyter notebookun bir diğer özelliği de aşağıdaki örnekte gösterilmiştir.

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.

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 istersek **Shift+TAB** yapabiliriz. Aynı işlemi, hücreye **help(x.count)** yazarak ta 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.