

MODÜL 1

PYTHON PROGRAMLAMA DİLİ

1.1. Bilgisayar Programları Nasıl Çalışır?

Bilgisayarlar **yazılım** ve **donanım** olmak üzere iki bileşenden oluşur. Donanımlar, bilgisayarların fiziksel parçalarıdır. Klavye, fare, ekran gibi bileşenler bilgisayarın donanım birimleridir. Yazılım ise bilgisayarda donanıma hayat veren ve bilgi işleminde kullanılan programlar, yordamlar, programlama dilleri ve belgelerin tümüdür. (TDK, 2020)

Bilgisayarlarda bulunan oyunlar, müzik programları, ofis programları hatta işletim sistemleri dâhil tüm yazılımlar bir programlama dili ile yazılmıştır.

Bilgisayarlar kendilerine verilen görevleri yerine getirirken bu işi doğası gereği değil, sahip oldukları işlemci komut setine ve işletim sisteminin yürütebileceği özellikte hazırlanmış yönergeleri yerine getirebilirler. Örnek olarak bilgisayardan bir gün içerisindeki ortalama sıcaklığı hesaplamasını istersek gerçekte burada kullanılan sıcaklık, günün kaç saat olduğu gibi veriler bilgisayar tarafından bilinmemektedir. Bu veri, bilgisayar açısından pek bir şey ifade etmeyeceği gibi bilgisayara yönergeler verilerek bir dizi işlem yaptırılabilir.

- Sıcaklığı temsil eden bir sayıyı kabul etmek,
- Günün kaç saat olduğunu temsil eden bir sayıyı (24) kabul etmek,
- Gün içindeki saatlik sıcaklıkları toplamak,
- Toplam sıcaklık değerini 24'e bölmek
- Sonucu görüntülemek.

Bu şekilde basit bir akış oluşturulabilir. Ancak bu şekilde bir işlem dizisi bilgisayar için anlamsız olabilir.

1.2. Doğal Diller ve Programlama Dilleri

Dil, düşünceleri ifade etmek ve kaydetmek için bir araçtır. Her toplumun kullandığı dil, birbirinden farklı olduğu gibi toplum içinde de dil bölgelere göre farklı biçimlerde kullanılabilir.

Bilgisayarın anladığı gerçek dil, 0 ve 1'lerden ibarettir. Bilgisayar bizim kendisine verdiğimiz komutları makine diline çevirerek icra eder.

Gerçekte bilgisayarları insanlar programlamaktadır. Programcıların direkt makine dilinde program yazmaları çok zahmetli ve uzun bir süreçtir. Ayrıca bilgisayar donanımındaki hızlı gelişim, yazılım geliştirme sürecinde daha pratik yöntemlerin gelişmesini gerekli kılmıştır. Bu gelişmeler yazılım geliştiren bireyler için programlama dillerinin ortaya çıkmasına neden olmuştur. Böylece yazılım geliştiricilere, kendi dillerine daha yakın ve hızlı geliştirme yapmaları için imkân tanınmıştır. Programlama dillerinde yazılan komutlar, direkt olarak bilgisayarın anlayabileceği komut yazma işlemlerini derleyici veya yorumlayıcılara bırakmışlardır.

C, Java, Python ve Pascal gibi programlama dilleri çıkıp, insan diline benzer yapıda yazılım geliştirme olanağı sağlayınca, yazılım geliştirme işi kolaylaşmış ve yazılımcı sayısında da artış görülmüştür. Python dili de İngilizce düşünen ve konuşan insanlar için neredeyse biriyle konuşuyormuşçasına yazılım geliştirme imkânı tanımıştır.

Makine dilinde 0 ve 1'lerin herhangi bir işleme tabi tutulmadan işlem yapılması, muhtemelen çok zor bir süreç olurdu. Makine dili; **merkezi işlem birimi (CPU)** tarafından anlaşılıp çalıştırılabilen komutlardan oluşan program yazma aracıdır. Makine dili üzerinden herhangi bir işlem ya da dönüştürmeye gerek kalmadan bilgisayar tarafından doğrudan anlaşılır. Bu dil, sadece 0 ve 1 sayılarından oluşan, ikili kodların anlamlı kombinasyonlarından meydana gelmektedir.

Makine dili insanlar tarafından oluşturulmuştur. Ancak 0 ve 1'lere bakarak bir çıkarımda bulunmak zordur. Bilgisayara verilen komutların bilgisayar tarafından anlaşılması için bir dizi işlem den geçmesi gerekmektedir:

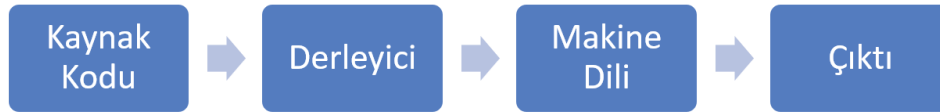
1. **Fetch** Bir sonraki görev hafızadan getirilir.
2. **Decode** Görevin ne demek istediği deşifre edilir.
3. **Execute** Görev yerine getirilir ve işlem gerçekleştirilir.
4. **Stop** Sonuç saklanır.

1.3. Derleyici ve Yorumlayıcılar

Makine dili, düşük seviyeli bir programlama dilidir ve bu programlama dili ile bilgisayara komutlar vermek, işlem yaptırmak zordur. Kullanıcıların kolaylıkla program yazmaları için yüksek seviyeli programlama dillerini kullanmaları gerekir. Yüksek seviyeli dil ile yazılmış olan programa **kaynak kodu** denir. Kaynak kodunun bilgisayar tarafından anlamlandırılabilmesi için makine koduna dönüştürülmesi gerekir. İşte, bu iş ya **derleyici (compiler)** ya da **yorumlayıcı (interpreter)** tarafından gerçekleştirilir.

Derleyici (compiler), belirli bir programlama dilinde yazılmış ifadeleri işleyen ve bunları bir bilgisayar işlemcisinin kullandığı makine diline veya “koda” dönüştüren özel bir programdır.

Derleyici, kaynak kodu derlemeden önce kodları kontrol eder. Olası yazım hataları veya dilin yapısına uygun olmayan kalıplar varsa kullanıcıya çıktı olarak bildirir. Bu sayede kullanıcı, **compile** işlemi esnasında hataları ayıklar.



Şekil 1.1: Derleyici işlem adımları

Yorumlayıcılar (interpreter), programı okuma ve yürütme yeteneğine sahip sistem yazılımı olarak adlandırılabilir. Yorumlayıcı programı satır satır yorumlar, bu işlem yukarıdan aşağı doğru yapılır ve kullanıcıya bildirilir. Daha alt satırlarda hata varsa bu hatalar bulunamaz çünkü satır satır işlem yapılmaktadır.



Şekil 1.2: Yorumlayıcı işlem adımları

MODÜL 1

1.4. Derleyici – Yorumlayıcı Farkı

Derleyici	Yorumlayıcı
<ul style="list-style-type: none">■ Tüm programı bir bütün olarak makine diline çevirir.■ Makine diline çevrilmiş bir programın çalışması daha hızlıdır.■ Daha fazla bellek gerektirir.■ Tüm hatalar belirlenerek bir seferde çıktı verir. Hata ayıklamak zordur.■ C, C++ gibi diller derleyici kullanır.■ Exe dosyaları derlenmiş uygulamaya örnek olarak verilebilir.	<ul style="list-style-type: none">■ Programın bazen tek satırını bazen tek ifadesini alıp işler, satır satır işlem görür.■ Daha yavaştır.■ Hafızada fazla yer kaplamaz.■ Gördüğü ilk hatada çalışmayı durdurur. Hata ayıklamak daha kolaydır.■ Python, Ruby, Java dilleri yorumlayıcı kullanır.

1.5. Python Nedir?

Python, 90'lı yılların başında Amsterdam'da Guido Van Rossum tarafından geliştirilmeye başlanan bir programlama dilidir. Python yazılım geliştirme ve veri analizinde ön plana çıkmıştır. Python'un standart kütüphanesi; geliştirme araçları diğer birçok kütüphanesi açık kaynak kod olarak ücretsiz şekilde indirilebilmektedir.

Python nesne yönelimli, yorumlanabilen ve yüksek seviyeli bir programlama dilidir.

1.6. Python Sürümleri

Python 2 ve Python 3 olarak adlandırılan iki ana Python dağıtımı bulunmaktadır. Python 2, Python'un eski sürümüdür. En son 2011 yılında 2.72 sürümü yayınlanmış ve gelişimi durmuştur.

Kitap yazım sürecinde Python Vakfının kendi sitesi olarak <https://www.python.org/> sitesinde son sürüm olarak 3.82 versiyonu yayınlanmaktadır. Ayrıca Vakıf düzenli aralıklarla güncelleme yapmaktadır.

Python 2 ile yazılan komut dosyaları Python 3 yorumlayıcısı tarafından çalıştırılmaz. Kodlardaki pek çok uyumsuzluk üzerinde düzenleme yapılması gerekmektedir

Python 3, sadece Python 2'den daha iyi bir versiyon değil, öncesine çok benzemesine rağmen tamamen farklı bir dildir.

Bu kitaptaki uygulamalar ve kodlar Python 3 diline göre hazırlanmıştır. Python 3.4, python 3.5, Python 3.6 ve Python 3.7 sürümlerinde yazılmış kodların tamamı Python 3.82 sürümünde çalışmaktadır.

1.7. Neden Python?

Python, popülerliğini hızlı bir şekilde yükseltmektedir. İlk defa programlamaya başlangıç yapanlar ya da farklı dillerde uzmanlaşanlar “Python öğrenmeli miyim?” sorusunu sormaktadır.

Uzun yıllar önce programlama öğrenmek isteyenler için bir programlama dilini öğrenmek veya aşına olmak şimdikinden çok daha zordu. Ancak zamanla insan diline yakın denilebilecek **yüksek seviyeli** programlama dilleri ortaya çıkmıştır.

Python kodlarını yorumlamak ve öğrenmek diğer dillere göre daha kolaydır. Diğer dillerde bulunan noktalama işareti zorunlulukları, parantezler veya kurallar programlamaya yeni başlayan kullanıcı için zaman zaman zorluklar çıkarmaktadır. Ancak Python'da bu tür zorunluluklar olmadığı gibi yapısı itibarıyla diğer dillere göre daha sadedir. Ayrıca Python, yıllar içerisinde belirli bir olgunluğa, geliştirici topluluğuna ve öğretici dokümana sahip olmuştur. Python dili ile program geliştirirken karşılaşmanızın muhtemel olduğu birçok sorunun cevabı stackoverflow gibi sitelerde bulunmaktadır. Bu da Python'u öğrenirken hızlı ilerlemenize olanak sağlamaktadır. Python, web geliştirme uygulamalarında kullanılabileceği gibi, kullanıcı ara yüzüne sahip (GUI) PyQt gibi kütüphanelere de sahiptir.

Python'un veri bilimi, veri analizi ve yapay zekâ ile ilgili gelişmiş kütüphaneleri olduğu için bu alanda ilerlemek isteyen kişilerin en çok kullandığı programlama dillerinin başında Python gelmektedir.

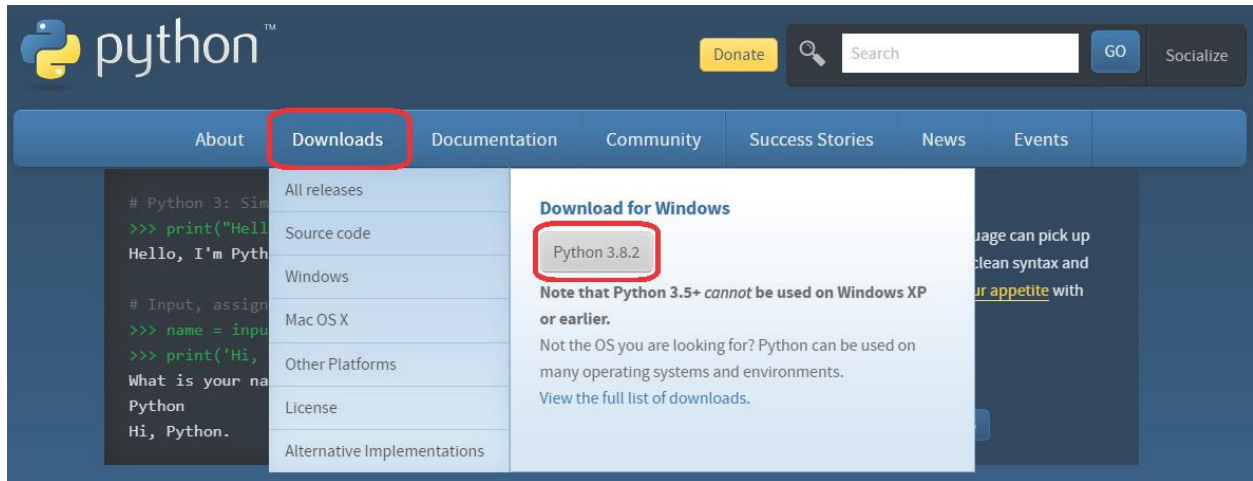
1.8. Python'u İndirme ve Yükleme

Python ile yazılım geliştirmek için kullanabileceğiniz pek çok miktarda araç mevcuttur:

- anaconda
- jupyter notebook
- ipython
- orange
- spyder
- visual studio code
- pycharm

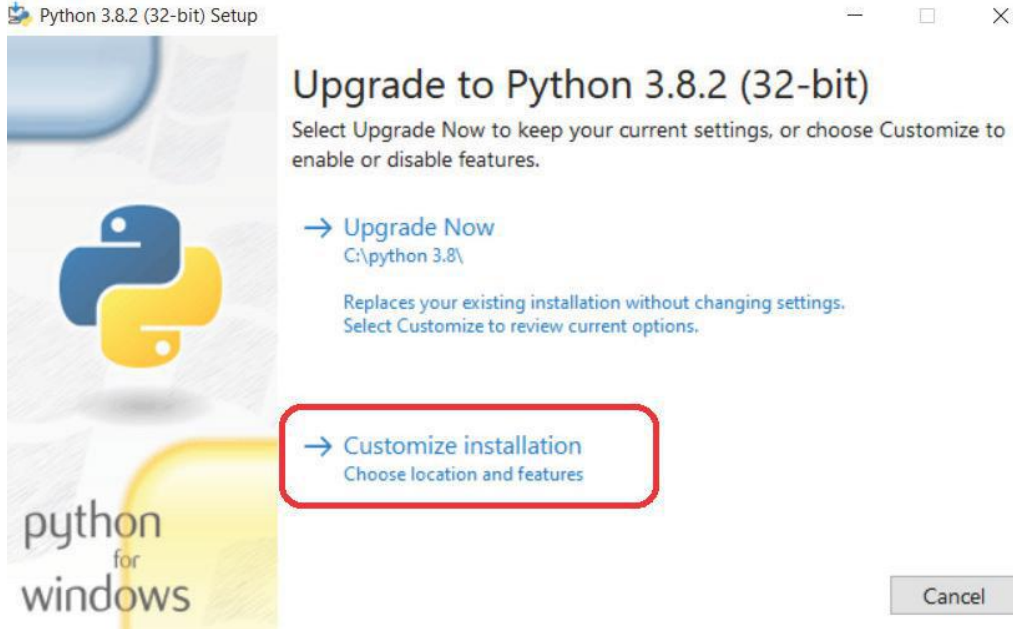
Kitapta Python'un resmî geliştirme ortamı kullanılacaktır. Bunun nedeni, kendi geliştirme ortamının (Idle) sade ve kolay olmasından dolayı ve güçlü özellikte bilgisayar gerektirmemesidir.

Python programını <https://www.python.org/> resmî sitesinden en son sürümünü indirip kurulması gerekmektedir. Kitap yazım aşamasında son versiyon olarak 3.82 sürümü olduğu için bu sürüm üzerinden Python programlama dili anlatılacaktır. Python programı güncellendikçe yeni sürümler indirilip kurulabilir. Site üzerinde farklı işletim sistemleri için kurulum paketi bulunmaktadır. Kullanıcı, sistemine en uygun paketi indirerek kişisel bilgisayarına kurulumu gerçekleştirebilir.



Şekil 1.3: Python resmi sitesi

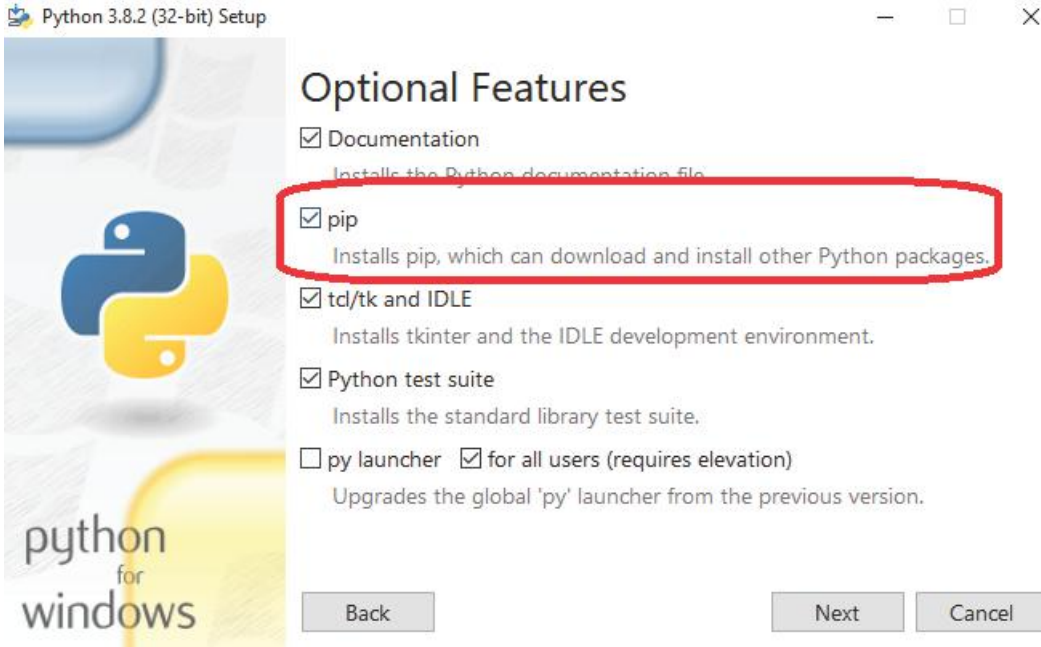
Kurulum ekranında **Customize Installation** seçeneği seçilmelidir. Bu sayede Python'un kurulacağı klasör seçilebilir.



Şekil 1.4: İlk kurulum ekranı

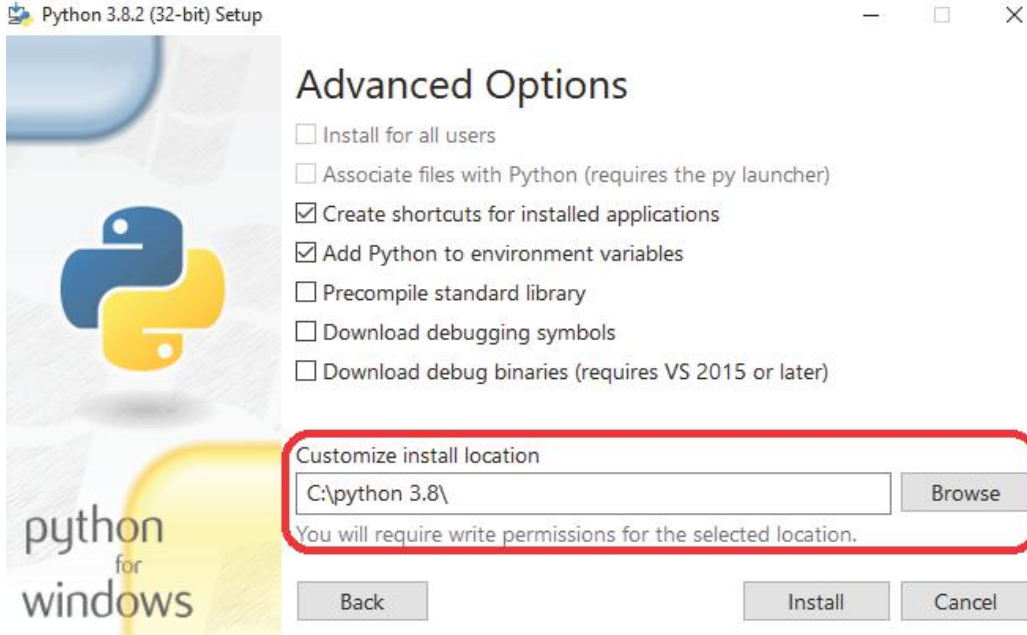
Kurulum ekranının ikinci sayfasında **pip** (paket yükleyici) seçeneği mutlaka seçilmeli, bu sayede ileride Python'a yeni kütüphanelerin eklenmesi sağlanabilir.

MODÜL 1



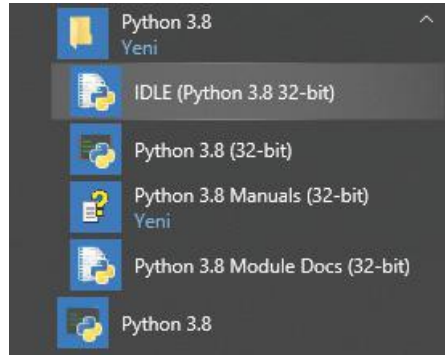
Şekil 1.5: pip paket yükleyicisinin seçilmesi

Kurulum ekranının 3. sayfasında, Python'un kurulacağı konum seçilmelidir. Eğer kurulum sihirbazı farklı bir konum gösteriyorsa değiştirilerek **c:\python 3.8** şeklinde erişilebilir bir konuma kaydedilmesinde fayda vardır.



Şekil 1.6: Kurulum klasörünün seçilmesi

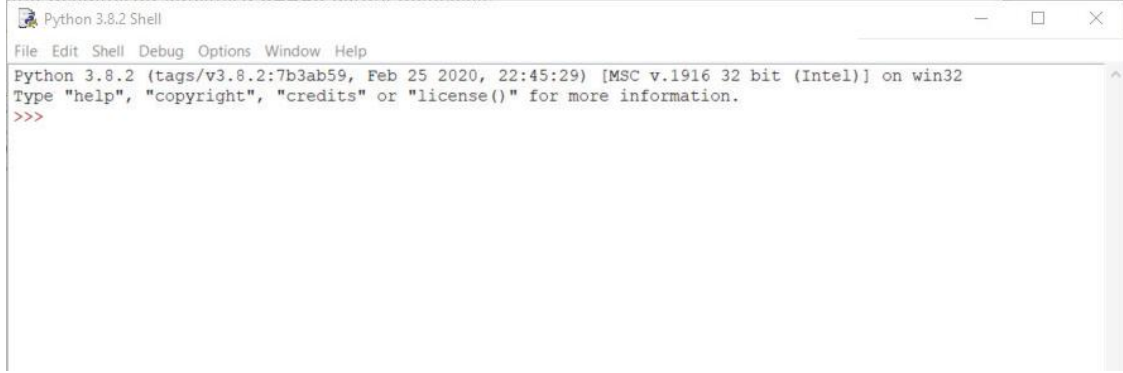
Kurulum tamamlandığında Python editörünü açmak için başlat menüsünden **IDLE** uygulaması tıklanarak Python editörü açılmalıdır.



Şekil 1.7: Windows başlat menüsü Python uygulaması

MODÜL 1

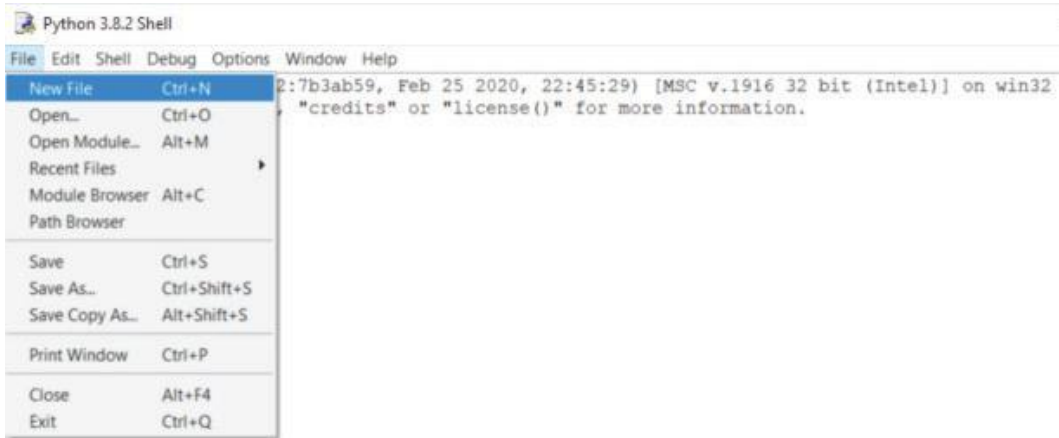
IDLE editörünü açtığımızda karşımıza **kabuk** ekranı gelecektir.



Şekil 1.8: Etkileşimli kabuk

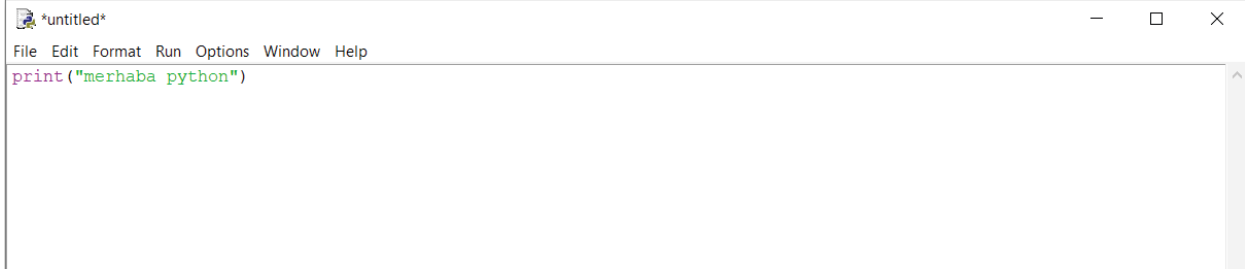
Kabuk ekranı üzerinde çeşitli aritmetiksel ve mantıksal işlemler yapılabileceği gibi, basit ve birkaç satırlık kodlarımızı burada çalıştırarak uygulamanın vereceği olası çıktılar da gözlemlenebilir.

Yeni bir Python dosyası oluşturmak ve kod yazıp çalıştırmak için, **File** menüsünden **New File** seçeneği seçilmelidir.



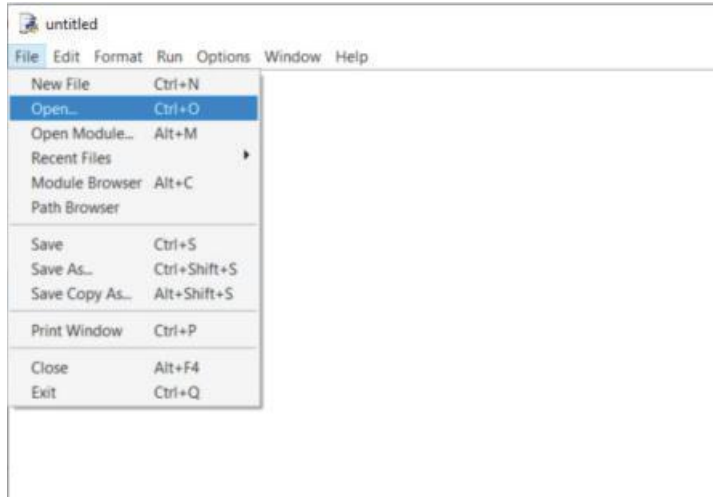
Şekil 1.9: Yeni Python dosyasının oluşturulması

Açılan **Not Defteri** benzeri pencerede kodlar yazılarak çalıştırılabilir.



Şekil 1.10: İlk kodlarımızı yazalım

Yazılan Python kodlarının çalıştırılması için kaydedilmesi gereklidir. Yeni bir pencere açılarak kodlar yazılabileceği gibi daha önceden oluşturulan programlar da çağrılabilir.



Şekil 1.11: Daha önceden oluşturulan dosyaların çağırılması

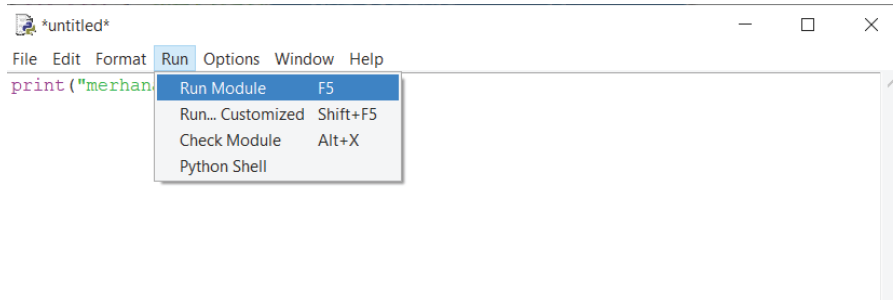
File menüsünde bulunan **Open** seçeneği ile daha önceden oluşturulmuş bir dosya yüklenebilir. Dosyayı kaydetmek için Save – Save As.. seçeneği seçilerek kaydedilecek dosyanın konumu seçilmeli, sonrasında dosyaya bir isim verilmelidir.

MODÜL 1

1.9. İlk Kodlarımızı Yazalım

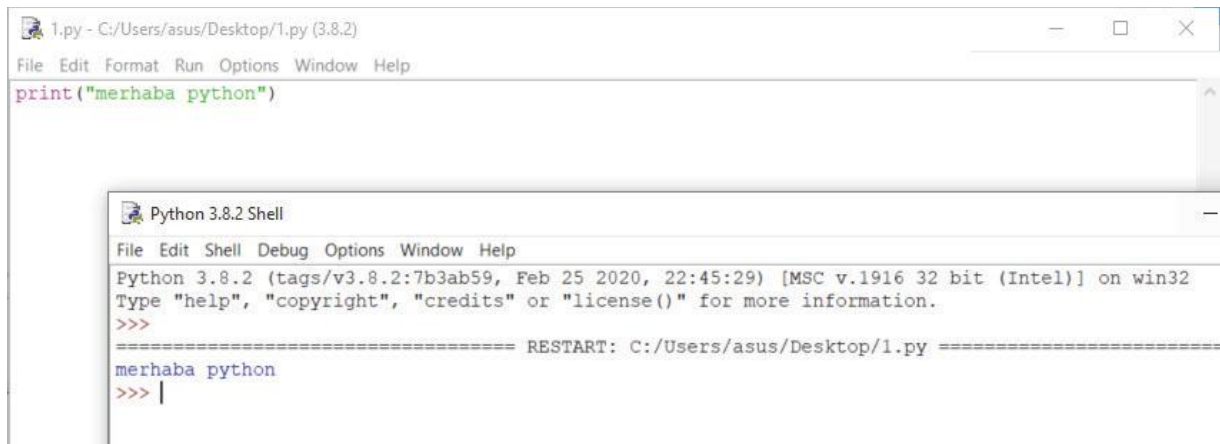
Bu aşamada ilk kodları yazılarak Python'un nasıl çalıştığı gözlemlenebilir.

Python editörü üzerinde `print("merhaba python")` şeklinde kodlar yazılmalıdır. Dosyayı çalıştırmak için **Run** menüsünde bulunan **Run Module** seçeneği seçilmeli ya da klavyeden **F5** tuşuna basılmalıdır. Eğer dosya kaydedilmemişse dosyanın kaydedilmesi için **Save – Save As** penceresi açılacak ve dosyayı kaydedebilmek için konum ve dosya adı girilmesi istenecektir.



Şekil 1.12: Uygulamanın çalıştırılması

Uygulamayı çalıştırıldığında **Etkileşimli Kabuk** penceresi açılacak ve yazılan kodların çıktısı etkileşimli kabuk üzerinde görüntülenecektir.

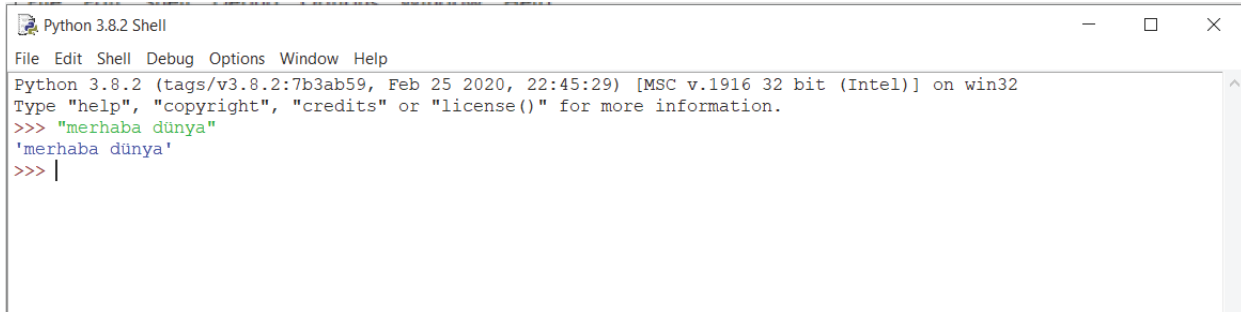


Şekil 1.13: Uygulamanın çıktısı

1.10. Etkileşimli Python Kabuğu

Etkileşimli kabuk kullanılarak, Python kodları kaydedilmeye gerek kalmadan, doğrudan konsol benzeri bir yazılım üzerinde denenebilir. **Etkileşimli kabuk (Interactive Shell)** yazılan komutun sonucunu çıktı olarak vererek kullanıcının dili öğrenmesini kolaylaştırır, uygulamaların denenerek çıktı alınmasına olanak sağlar.

Bunun yanında etkileşimli kabuk üzerinde çalışırken print fonksiyonu kullanmaya gerek kalmadan anlık olarak yapılan işlemlerin çıktısı görüntülenebilir. Kitabın ilk bölümlerinde yapılan uygulamalar, etkileşimli kabuk üzerinde yapılarak, Python programlama diline daha kolay bir şekilde adapte olunması sağlanacaktır.



```
Python 3.8.2 Shell
File Edit Shell Debug Options Window Help
Python 3.8.2 (tags/v3.8.2:7b3ab59, Feb 25 2020, 22:45:29) [MSC v.1916 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> "merhaba dünya"
'merhaba dünya'
>>> |
```

Şekil 1.14: Etkileşimli Python kabuğu

1.11. Google Colab Kullanımı ve Kitaptaki Uygulamalara Erişim

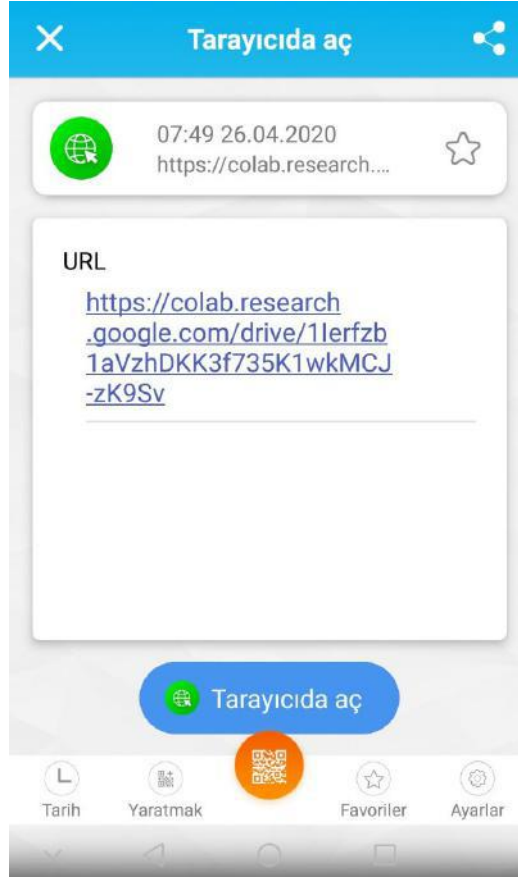
Kitaptaki kod örneklerine kare kod okutarak erişilebilir. Örnekler internet üzerinde sanal bir Python makinesi oluşturan (aslında Jupyter Notebook) Google Colab yardımıyla (bir Google hesabınız olduğunuzu varsayarak) kolay bir şekilde oluşturulabilir. Bilgisayar veya telefonda herhangi bir program kurulmadan Google Colab ile Python kodları yazılabilir. Google Colaboratory'e <https://colab.research.google.com/> adresinden erişilebilir.

Örnek kodların nasıl çalıştırılacağı hakkında bilgi edinmek için aşağıdaki yönergeler izlenmelidir.

1. Öncelikle bir Google hesabı gerekmektedir. Google hesabı yoksa veya kodlar yerel bilgisayarda çalıştırmak istenirse (py uzantılı dosyalar) sayfanın altında bulunan kare kod kullanılmalıdır.

MODÜL 1

2. Akıllı cihazınızdan **kare kod okuma uygulaması** açılmalıdır.
3. Bölümler için ayrı ayrı oluşturulmuş kare kodlardan istenilen bir tanesi okutulabilir. **Linki aç, tarayıcıda aç** tuşuna basılmalıdır.



Şekil 1.15: Mobil cihazdan karekod okuyucu ile kodun okutulması

4. Telefonda Google oturumu zaten açık olacağı için otomatik olarak Google Colab üzerinden örnek kodlara erişilebilir.

Metin Bölümü
Yazılar, Açıklamalar

Kodu çalıştırmak için tıklayınız.

Kodun çıktısı

Kod bölümü

Bu bölüm
kodu çalıştırdığınızda görünecektir.

Örnek 7
Aşağıda 3 değişkene de tek satırda 1 değeri atanmıştır.

```
[ ] 1 a = b = c = 1
      2 print ('1. sayı=', a)
      3 print ('2. sayı=', b)
      4 print ('3. sayı=', c)
```

1. sayı= 1
2. sayı= 1
3. sayı= 1

Örnek 8
Değişkenler yan yana yazılır (aralarına virgül eklenerek) değerleri de aynı şekilde

Şekil 1.16: Mobil cihazdan Google Colab'a erişim

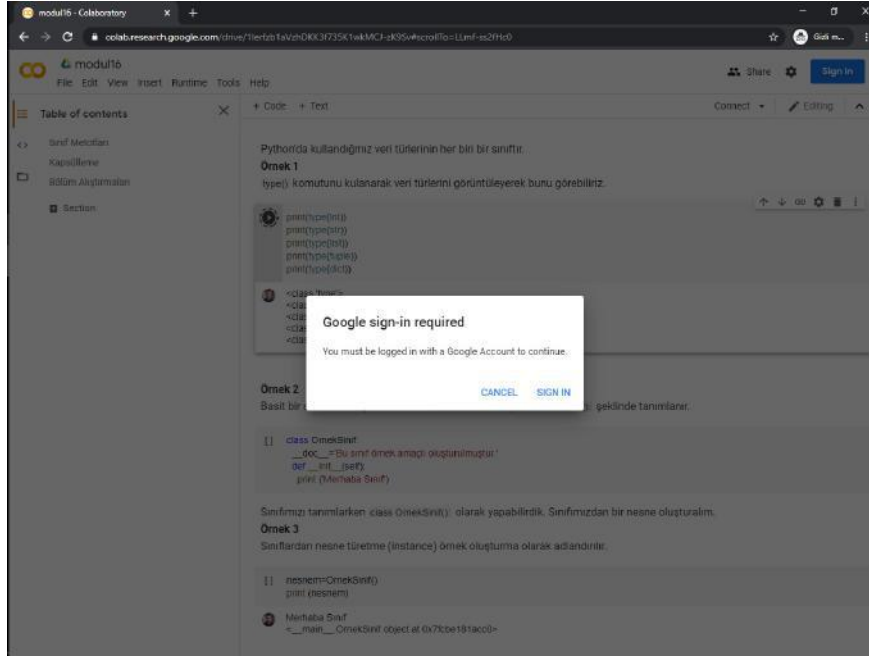
MODÜL 1

Herhangi bir kodu çalıştırmak için kod bloğuna gelerek [] tuşuna basılması yeterlidir.



Şekil 1.17: Butona basılınca kodun çıktısı görüntülenecektir.

5. Kodu çalıştır'a tıkladığında aşağıdaki gibi bir hata mesajı alınır. **SIGN IN**'e basılarak Google hesabına giriş yapılmalıdır.



Şekil 1.18: Google hesabına giriş ekranı

6. Program kodunu ilk defa çalıştırırken güvenlikle ilgili bir uyarı mesajı gelirse mesaj okunduktan sonra **Run Anyway** tuşuna basılmalıdır.

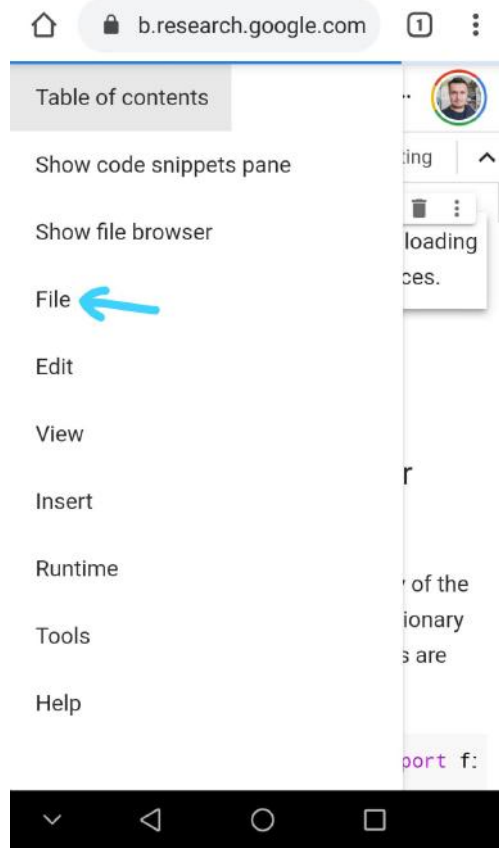
7. İstendiği takdirde kodlar düzenlenebilir, notlar eklenebilir ve değişiklik yapılabilir. Kodlar bilgisayara indirilebilir ve/veya Google Drive'a kaydedilerek yine Google Colab üzerinden çalışmaya bilgisayardan devam edilebilir. Bunun için telefonda Google Colab menüsüne erişilmelidir. Sayfanın sol başında bulunan menü işaretine tıklanmalıdır.



Şekil 1.19: Google hesabına geçiş

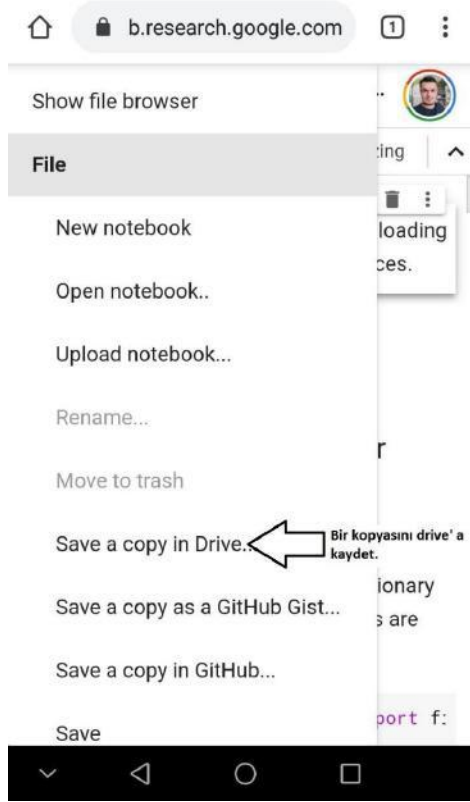
MODÜL 1

8. Daha sonra **File** menüsü tıklanmalıdır.



Şekil 1.20: Google hesabına geçiş adımları

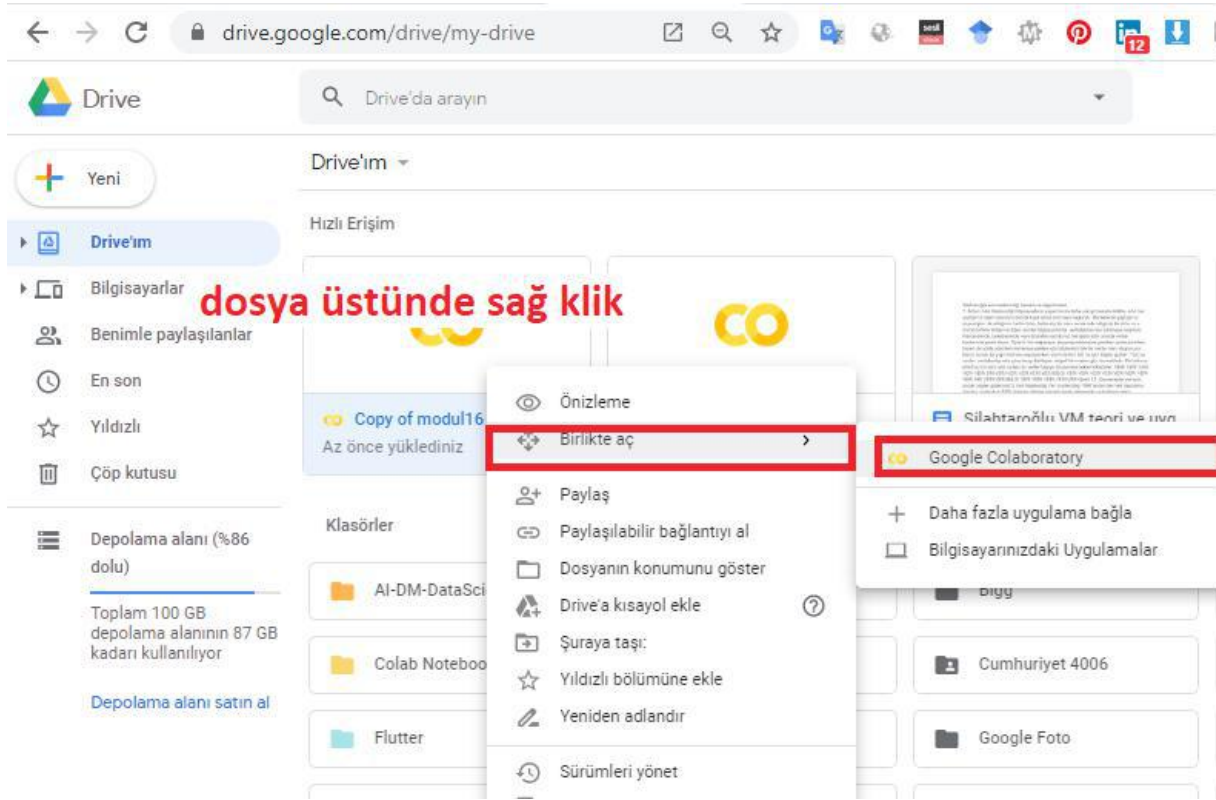
9. **Save a copy in Drive**'a basılarak, dosyanın bir kopyası kullanıcının drive hesabı içinde oluşturulup, oluşturulan kopya açılarak çalışma, kullanıcının kendi dosyası üzerinden devam edebilmesini sağlar.



Şekil 1.21: Uygulama kodlarının kendi hesabımıza aktarılması

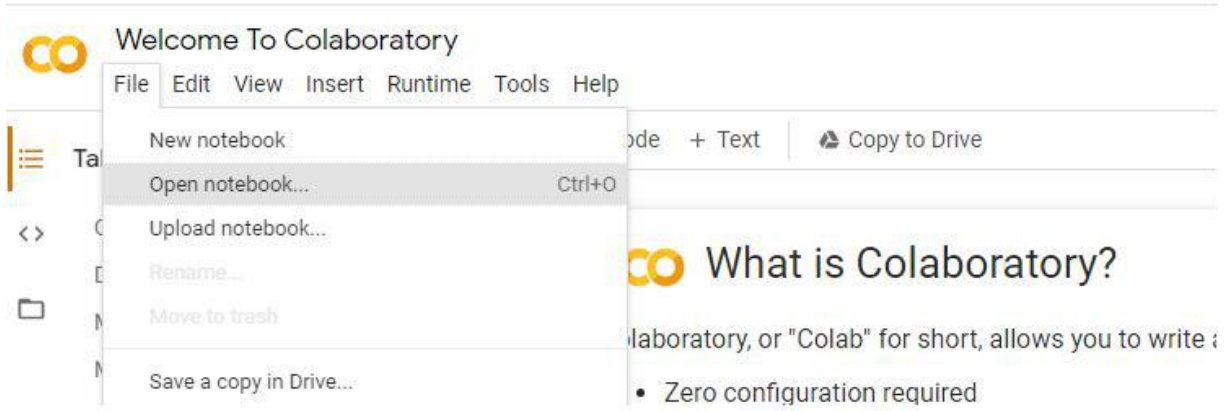
MODÜL 1

10. Bu aşamadan sonra bilgisayardan Drive üzerinden, dosyaya sağ tıklanarak birlikte aç Google Colab seçilip çalışmaya devam edilebilir.



Şekil 1.22: Kodların Google Colab ile açılması

11. Başka bir yöntem ise <https://colab.research.google.com/> adresine **File** menüsü altında **open notebook** seçildikten sonra Google Drive seçilerek ve ilgili dosyaya tıklanmalıdır.



Şekil 1.23: Google Colab açmak için alternatif yöntem

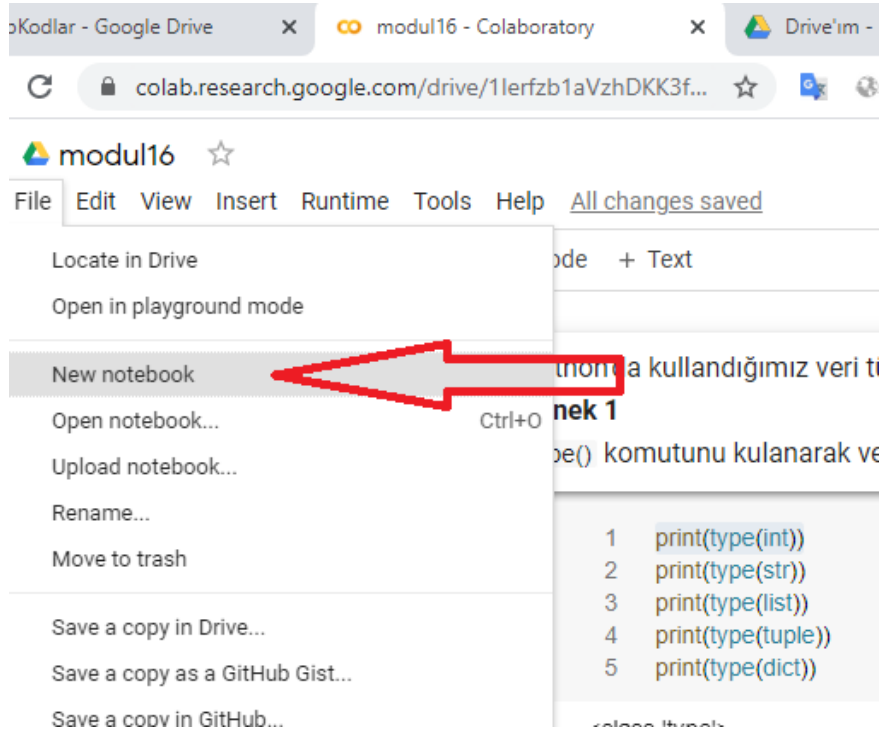
MODÜL 1

The screenshot shows a Google Drive interface with a navigation bar at the top containing 'Examples', 'Recent', 'Google Drive', 'GitHub', and 'Upload'. The 'Google Drive' tab is highlighted with a blue box. Below the navigation bar, there is a 'Filter notebooks' section with a search icon. A table lists several notebooks with columns for 'Title', 'Owner', 'Last modified', and 'Last opened'. The first notebook, 'Copy_of_modul16', is highlighted with a blue arrow pointing to it. The 'CANCEL' button is located at the bottom right of the interface.

Title	Owner	Last modified	Last opened
<u>Copy_of_modul16</u>	Murat ALTUN	9 minutes ago	9 minutes ago
modul16	Murat ALTUN	41 minutes ago	41 minutes ago
Modul NTP.ipynb	Murat ALTUN	1 hour ago	1 hour ago
modul7	Murat ALTUN	2 hours ago	2 hours ago
modul8.ipynb	Murat ALTUN	2 hours ago	2 hours ago

Şekil 1.24: Uygulama kodlarının kullanıcının Google hesabına

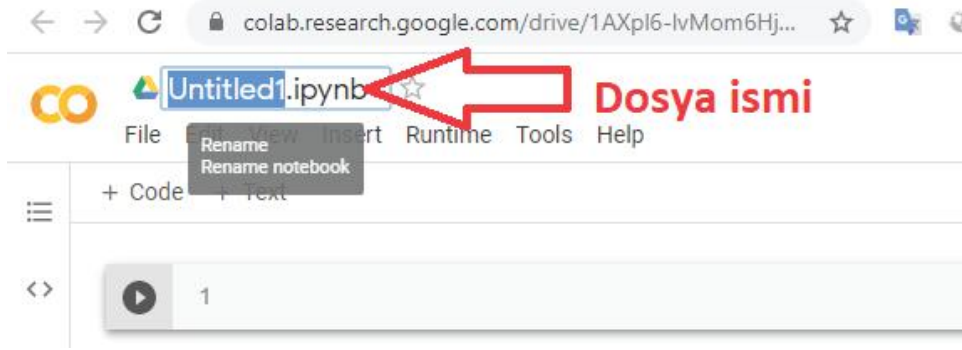
12. Tüm bu süreçlerden bağımsız olarak <https://colab.research.google.com/> adresine yeni bir notebook oluşturularak, Python kodları kolayca denenebilir.



Şekil 1.25: Boş bir notebook dosyasının oluşturulması

MODÜL 1

13. Yeni notebook açtıktan sonra ismine tıklanarak değiştirilebilir. Değişiklikler otomatik olarak kaydedilmektedir. Notebook daha sonra kaydedilen isimle, Google Drive üzerinde Colab Notebooks klasörü içinden erişilebilir.



Şekil 1.26: Notebook dosyasının isminin değiştirilmesi

Sorular ve ayrıntılı yardım için:



Şekil 1.27: Yardım için karekodu okutun.