

ÖĞRENME BİRİMİ 3

PROGRAMLAMA DİLİ

TEMELLERİ



Neler Öğreneceksiniz?

Bu öğrenme birimi ile;

- Program, yazılım ve programlama dili kavramlarını açıklayabilecek,
- Yorumlama ve derleme kavramlarını tanımlayabilecek,
- Python programlama dilinin avantajlarını sıralayabilecek,
- Python ile yapılabilecek proje fikirleri geliştirebilecek,
- Python kurulumu yapabilecek,
- IDLE üzerinde kod çalıştırabilecek,
- Python için gerekli araçları belirleyip kurulumlarını yapabileceksiniz.

Anahtar Kelimeler:

Program, yazılım, programcı, yorumlama, derleme, IDLE, IDE.



Hazırlık Çalışmaları

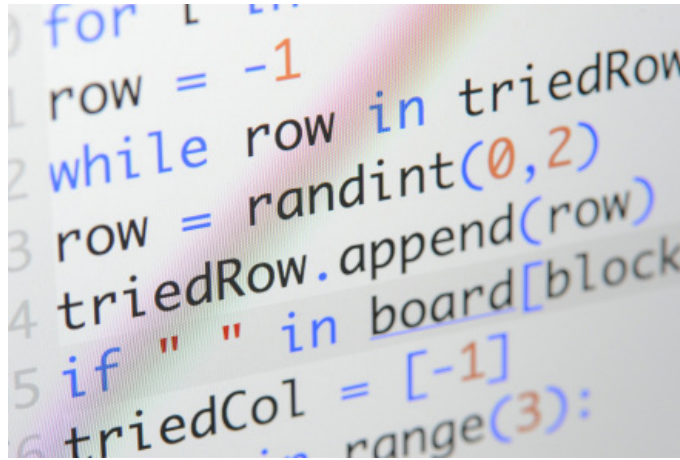
1. Son yıllarda dünya genelinde yazılımcılar tarafından en çok hangi programlama dillerinin tercih edildiğini araştırınız ve sebeplerini tartışınız.
2. Açık kaynak ve kapalı kaynak kod kavramlarını araştırınız.

3. PROGRAMLAMA DİLİ TEMELLERİ**3.1. Program ve Yazılım****Görsel 3.1:** Program ve yazılım

Program, herhangi bir elektronik cihaza işlem yaptırmak için yazılan komutlar dizisidir.

Elektronik cihazlara; bilgisayar, cep telefonu, akıllı saat, akıllı televizyonlar vb. örnek olarak verilebilir. Program yazımı denilince akla ilk gelen cihazlar bilgisayarlar olsa da günümüzde pek çok elektronik cihaza kod yazılabilir.

Çoğu zaman **“program”** kelimesi ile **“yazılım”** kelimesi birbirinin yerine kullanılmaktadır. Ancak bu doğru değildir. Yazılım, programa göre daha geniş kapsamlıdır. Programlar tek başlarına yazılım olarak adlandırılmaz. Yazılım denildiğinde; programın yanında belgeleme, yardım dosyaları, veri kaynakları ve yardımcı programlar akla gelmektedir. Örnek olarak, bilgisayarınızda bulunan **“Hesap makinesi”** bir yazılımdır. **“Hesap makinesi”** yazılımını kullanabilmek için çalıştırılan **“calc.exe”** ise programdır.

3.2. Programlama Dili**Görsel 3.2:** Programlama dili örneği

Tüm elektronik sistemler **“1”** ve **“0”**lardan oluşan kod bloklarını çalıştırır. **“1”** ve **“0”**lardan oluşan kodlama diline **“makine dili”** denir. Aşağıda makine dilinde yazılmış bir kod parçası görülmektedir.

```

11010011101000000
000000000000010001
11011000001100111
00110001100111010
11001111001010111

```

Bu kod parçası bilgisayarda bulunan **“Merkezi İşlem Birimi (CPU)”** için pek çok şey ifade etse de bizler için çok anlamlı gelmemektedir.

Bundan dolayı makine dili bir adım ileriye taşınarak **“Assembly dilleri”** oluşturulmuştur. Assembly dili, daha okunaklı olarak yazılan kodları makine diline dönüştürür. Aşağıdaki Assembly kod bloğu örneğini inceleyebilirsiniz.

```

mov r0,#1
mov r1,#1
l:
add r2,r0,r1
str r2,[r3]
add r3,#4
mov r0,r1
mov r1,r2
b l

```

Görüldüğü üzere, Assembly dilinde yazılan programlar da kolay anlaşılır değildir.

İşte bu noktada kolay yazılabilir ve anlaşılabilir kodlar oluşturmak üzere **“Programlama Dilleri”** geliştirilmiştir. Günlük hayatta kullanılan İngilizce ile programlar yazılmakta ve bu dilin makine diline çevrimi, arka planda otomatik olarak gerçekleşmektedir.

```

a = 5
b = 8
if a + b > 10:
    print "toplam, 10 dan büyüktür."
else:
    print "toplam, 10 dan büyük değildir."

```

Programlama dilini hiç bilmeyen biri bile bu kod parçasının ne amaçla yazıldığını tahmin edebilir. İşte bu yüzden program geliştirmek için **“Programlama dilleri”** kullanılmaktadır.

Programlama dili, belirli bir algoritmaya dayalı olarak bir yazılım programı oluşturmak için kullanılan, sıkı kurallara sahip, bir dizi komut ve talimatlar bütünüdür. Günümüzde, kullanıma sunulmuş onlarca programlama dili vardır. Her bir problem/algoritma için hangi programlama dilinin kullanılacağı tamamen programcıya kalmıştır.

Teorik olarak şu söylenebilir, herhangi bir “A” noktasından “B” noktasına gidilmek istendiğini düşününüz. Bu işlem yürüyerek, bisikletle, otomobille veya uçakla gerçekleştirilebilir. Sonuçta A noktasından B noktasına varılmış olur. Ancak maliyet, zaman, konfor gibi bazı parametreler açısından her bir yöntemin artıları ve eksileri vardır. Yürüyerek veya bisikletle gidildiğinde maliyet sıfıra yakın olurken zaman ve konfor konusunda ciddi sıkıntı yaşanır. Diğer taraftan otomobille daha kısa sürede varılabilir ancak bu sefer de maliyet yükselecektir. Uçak kullanıldığında süre daha çok kısaltılabilir ve yüksek konfor sağlanabilir fakat maliyet daha çok yükselecektir.

Özetle; en iyi denilebilecek programlama dili yoktur. Önemli olan önümüzde duran problem veya algoritma için en uygun programlama dilini seçmektir.

Python’un Kısa Hikâyesi

Python, son zamanlarda popüler bir dil hâline gelse de aslında yeni bir dil değildir. 1990’lı yılların başında **“Guido van Rossum”** isimli Hollandalı bir yazılımcı tarafından geliştirilmeye başlanmıştır. Çoğu kişi Python dilinin adını piton yılanından aldığını düşünür. Ancak gerçek böyle değildir. Python dilini geliştiren Guido van Rossum bu dili, “The Monty Python” adlı bir komedi grubunun “Monty Python’s Flying Circus” adlı gösterisinden esinlenerek isimlendirmiştir. Hâl böyle olsa da pek çok Python kitabının kapağında çeşitli piton yılanı figürlerini görmek artık sıradan bir durumdur.

3.3. Neden Python?



Görsel 3.3: Python'a giriş

Python programlama dilinin basit ve temiz bir söz dizimi vardır. Bu özelliğinden dolayı program yazmak, yazılan programı okumak ve anlamak diğer dillere nazaran daha kolaydır.

Python'un önemli bir özelliği de pek çok dilin aksine **"yorumlanan"** bir dil olmasıdır. Bu dilde yazılan kodlar derlenmeden direkt çalıştırılır. Python, bu özelliği ile teknik olarak bir programlama dili değil, bir betik (script) dilidir. Python'da hızlı bir şekilde program geliştirilebilir. Bu noktada yorumlama ve derleme olaylarını kısaca açıklamak faydalı olacaktır.

Yorumlama (Interpretation) işlemi, yazılan kodun satır satır okunup bilgisayarın işlemcisine özel makine diline anında çevrilmesi işlemidir. Program her çalıştırıldığında yorumlama işlemi tekrardan yapılır. Bundan dolayı derlenen yazılımlara göre bir nebze yavaş çalışacağını söylemek mümkündür. Bağımsız platform desteği sağlanması sayesinde hazırlanmış olan program, desteklenen her ortamda kolaylıkla çalıştırılabilir. Bu sayede yazılan programın boyutu da küçük olmaktadır.

Yorumlanarak çalıştırılan yüksek seviyeli diller doğrudan yorumlanmazlar. Genellikle bir ara forma (Opcode, Bytecode vs.) dönüştürülür ve bu kodlar yorumlanarak yerel makine diline çevrildikten sonra işletilir. Java, PHP, Python gibi yorumlanan diller aslında yorumlama aşamasına geçilmeden önce en az bir kere derlenir.

Derleme (Compilation) işlemi; yüksek seviyeli bir dilde yazılan programın bir başka hedef dile veya makine diline çevrilmesi işlemidir. Programı çalıştırmak için bir kere derleme işleminden geçirmek yeterlidir. Program her çalıştırılışında tekrardan derleme olayı gerçekleşmez.

Direkt makine dili veya çevrilme işleminde, platform bağımlılığından bahsedilmesi gerekir. Program her işletim sisteminde veya işlemcide çalıştırılmak isteniyorsa, her işletim sisteminde veya işlemcide ayrı ayrı derleme işleminden geçirmek gerekir. **Örnek:** C, C++... vb.

Python'da ve oldukça eski, popüler bir dil olan C++'ta birer klasik **"Merhaba Dünya"** uygulaması yazarak aralarındaki farkı inceleyebilirsiniz.

C++:

```
#include <iostream>;
using namespace std;
int main(){
    cout << "Merhaba Dünya!" << endl;
    return 0;
}
```

Aynı programı Python'da aşağıdaki şekilde yazmak mümkündür.

```
print("Merhaba Dünya!")
```

Her iki program da ekrana **"Merhaba Dünya!"** yazdırmaktadır. Ancak birinde 6 satır, diğerinde sadece 1 satır kod yazılmıştır. Bu örnekler bakıldığında Python'un son derece basit ve hızlı bir şekilde kod geliştirilebilen bir programlama dili olduğu görülmektedir. Python, birçok işletim sisteminde sınırsız bir şekilde çalıştırılabilmektedir: Windows, Linux/Unix, Mac OS X ve daha fazlası.

Python'u farklı kılan bazı özellikler şu şekilde listelenebilir:

- Açık kaynak kodlu olması
- Ücretsiz olması
- Hızlı ve kolay kurulabilmesi
- Sade ve kolay okunabilen kod yapısı
- Toplu ve düzenli kod yapısı
- Öğrenme ve adapte olma kolaylığı
- Kolay anlaşılır nesne tabanlı programlama özellikleri
- Güçlü ifade yeteneği
- Son derece esnek modüler yapısı
- "Exception" tabanlı hata yönetimi
- Yüksek seviye dinamik veri yapıları
- Oldukça geniş standart kütüphanelerinin olması
- Otomatik hafıza temizliği
- C, C++, Java ile kolay entegre edilebilmesi
- Hemen her tür platformda sınırsız çalışması
- Az kod / çok iş anlayışı

3.4. Python ile Neler Yapılabilir?

Python dili, yukarıda sayılan avantajları sayesinde büyük küçük pek çok şirketin kullandığı bir dildir. Dev sosyal medya şirketleri her zaman Python programcılarını ihtiyaç duymaktadır. Bu nedenle Python dilinin popülaritesinin son yıllarda arttığı söylenebilir. Bazı internet sitelerinde yıllara göre programlama dilleri popülarite indeksi yayınlanmaktadır. Bu sitelerden birisi incelendiğinde aşağıdaki gibi bir liste oluştuğu görülür:

May 2020	May 2019	Change	Programming Language
1	2	▲	C
2	1	▼	Java
3	4	▲	Python
4	3	▼	C++
5	6	▲	C#
6	5	▼	Visual Basic
7	7		JavaScript
8	9	▲	PHP
9	8	▼	SQL
10	21	▲	R

Görsel 3.4: Mayıs 2019 ve Mayıs 2020 ayları itibarıyla en çok tercih edilen programlama

Liste incelendiğinde Python'un Mayıs 2020 itibariyle dünyada en çok kullanılan 3. programlama dili konumunda olduğu görülür.

Aşağıda Python dilinin kullanıldığı alanlar listelenmiştir. Bu liste Python'un neden bu kadar popüler olduğunu açıklamaktadır.

- Yapay zekâ ve makine öğrenmesi
- Web uygulamaları
- Bilimsel hesaplamalar
- Veri analizi
- Masaüstü uygulama geliştirme
- Ağ ve soket programlama
- Nesnelerin interneti
- Kriptoloji
- Sistem yönetimi
- Oyun geliştirme vb.

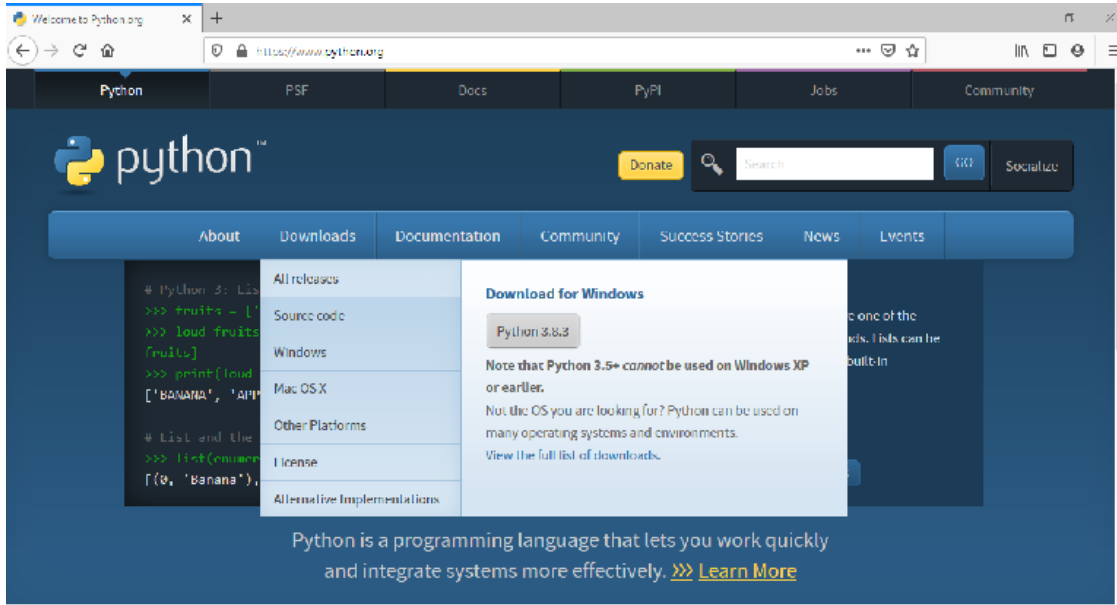
Görüldüğü üzere, Python ile çok çeşitli alanlarda uygulama geliştirmek mümkündür. Sonuç olarak denilebilir ki Python, öğrenilmesi ve kullanılması kolay, neredeyse her iş için bir kütüphanesi olan güçlü bir programlama dilidir.

3.5. Python Kurulumu

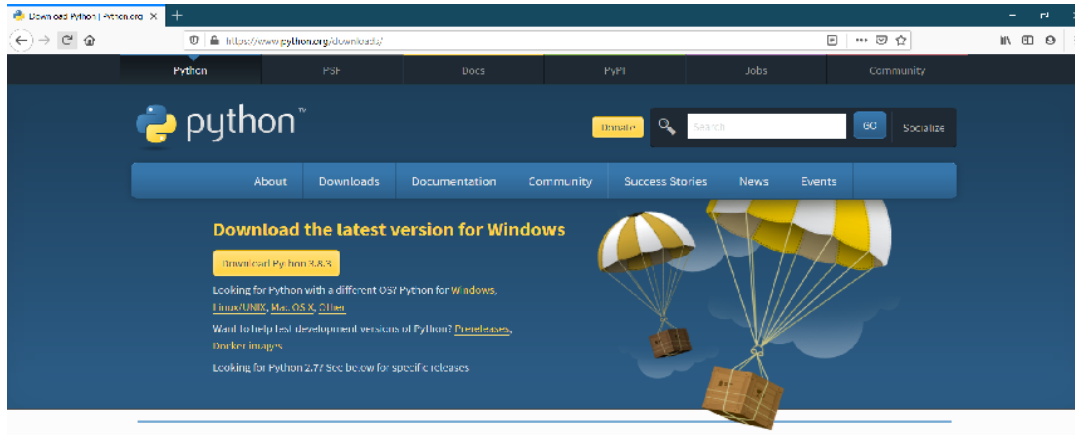
Python dilini bilgisayara kurarak program yazılabilir. Daha önce de belirtildiği gibi Python dilini bilgisayara kurmak oldukça hızlı ve kolay bir işlemdir.

Bunun için Python resmî web sitesi olan **python.org** adresi ziyaret edilmelidir. Siteye girildikten sonra;

- Üst tarafta bulunan **"Downloads"** menüsü altında yer alan **"Python x.y.z"** butonuna veya **"python.org/downloads/"** adresinde yer alan **"Download Python x.y.z"** butonuna tıklamak yeterlidir.



Görsel 3.5: Python kurulumu



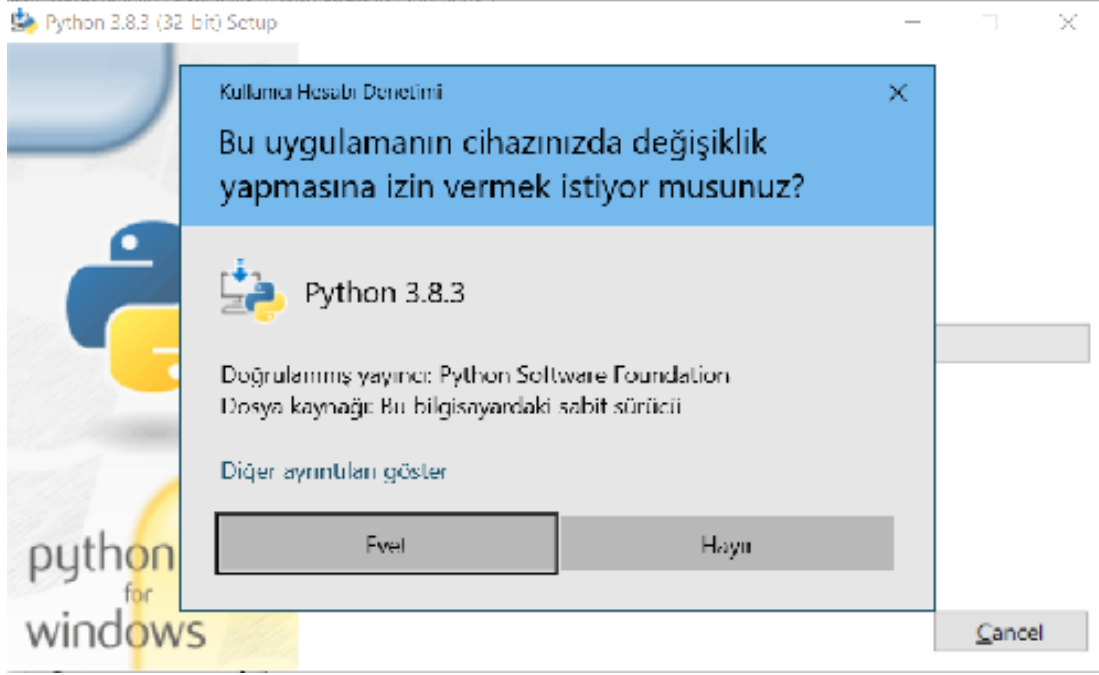
Görsel 3.6: Python kurulumu

Butona tıkladığınızda **"Python-x.y.z.exe"** dosyası bilgisayarınıza indirilecektir. İndirme işleminin ardından **.exe** uzantılı dosyayı çalıştırdığınızda kurulum aşaması başlayacaktır. Karşınıza çıkan ilk ekranda **"Add Python x.y to PATH"** seçeneğini işaretlemeniz tavsiye edilir. Ardından **"Install Now"** butonuna tıklanıp bir sonraki ekrana geçiş sağlanır.



Görsel 3.7: Python kurulumu

İşletim sistemi **"Kullanıcı Hesabı Denetimi"** onayı isterse **"Evet"** seçeneğine tıklanarak kurulumu devam edilir.



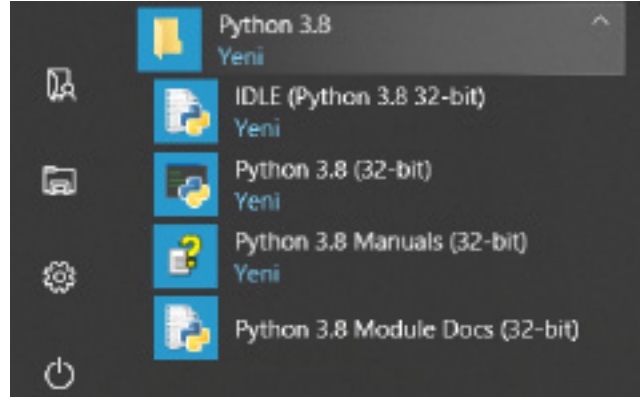
Görsel 3.8: Python kurulumu

Kısa bir süre sonra **“Setup was successful”** yazısı görüldüğünde kurulum işlemi tamamlanmış olacaktır. Artık Python dili bilgisayarda kullanılabilir hâle gelmiş demektir. **“Close”** butonu tıklanarak pencere kapatılır.



Görsel 3.9: Python kurulumu

İlk komutu yazmak için kurulumun ardından **“Başlat”** menüsü içinde yer alan **“Python 3.8”** menüsündeki **“Python 3.8 (32-bit)”** programını çalıştırmak gerekir. Sürüm bilgisinin (3.8) kurulum yapılan sürüme göre değişiklik gösterebileceği unutulmamalıdır.

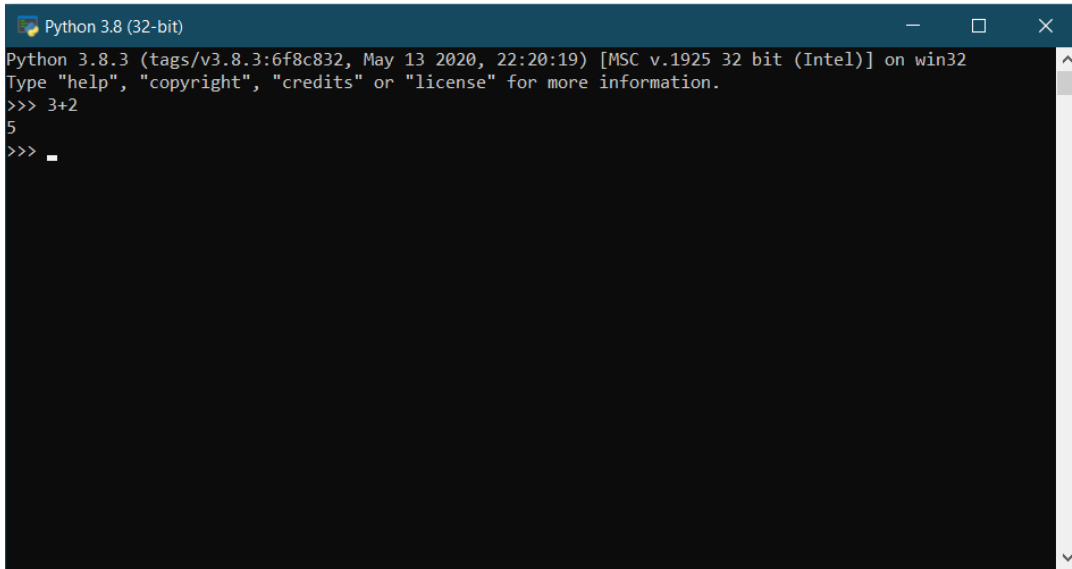


Görsel 3.10: Python programlama dilini başlatma

Ardından “>>>” karakterlerinin yanında yanıp sönen imlecin olduğu yere ilk komut yazılabilir. Bu ekran, “**konsol ekranı**” olarak adlandırılır.

3 + 2

Kod Örneği 3.1



Görsel 3.11: Python konsol ekranı

Python dili, bu ekranda komutları çalıştırabilir. Ancak kısıtlı imkânların bulunduğu bir program olduğu bilinmelidir. Bu nedenle komutların daha rahat yazılabileceği bir arayüz de kurulum ile gelmektedir.

Bu sefer, “Python 3.8” menüsündeki “**IDLE (Python 3.8 32-bit)**” programı çalıştırılır. (IDLE: Integrated Development and Learning Environment / Bütünleşik Geliştirme ve Öğrenme Ortamı)

Karşınıza aşağıdaki pencere gelecektir. Bu ekrana “**Shell**” adı verilir.

Aynı komut buraya yazıldığında aynı sonucun hesaplandığı görülür.

```
Python 3.8.3 Shell
File Edit Shell Debug Options Window Help
Python 3.8.3 (tags/v3.8.3:6f8c832, May 13 2020, 22:20:19) [MSC v.1925 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> 3+2
5
>>> |
```

Görsel 3.12: Python Shell ekranı

Tek bir satır komut çalıştırmak değil de birden fazla kod satırı çalıştırılmak istenirse, **"File -> New File"** menü seçeneğinden yeni bir dosya oluşturulur ve kodlar bu dosyaya yazılıp çalıştırılır.

```
*untitled*
File Edit Format Run Options Window Help
a = 3
b = 4
print(a * b)
```

Görsel 3.13: Python kod dosyası

Başlık çubuğunda **"untitled"** yazdığı görülüyorsa henüz dosyanın kaydedilmediği anlaşılır. Hemen **"File -> Save"** menü seçeneğinden dosya istenen bir klasör altına kaydedilir. Dosya kaydedildikten sonra başlık çubuğundaki metnin değiştiği görülür(Dikkat edilirse dosya uzantısı **".py"** olarak kaydedilmiştir).

"Run -> Run Module" menü seçeneği ile yazılan program çalıştırılır. Programın çıktısı ayrı bir pencerede sunulacaktır.

Görsel 3.14: Python Shell ekranı

Görüldüğü üzere sadece birkaç dakika içinde sıfırdan Python dili bilgisayara kurularak ilk denemeler gerçekleştirilebilir. Bundan sonra diğer Python dili komutlarını ve özelliklerini öğrenecek ve kendi programlarınızı yazabilir hâle geleceksiniz.

3.6. Python için Gerekli Araçlar

3.6.1. Editör Kurulumu

Python'da kod yazıp çalıştırmak için önceki bölümde bahsedilen kurulumlar yeterlidir. Ancak kod satır sayısı arttıkça birden fazla dosya ile çalışmaya başlandığında şu ana kadar kullanılan IDLE (**Integrated Development and Learning Environment / Bütünleşik Geliştirme ve Öğrenme Ortamı**)'nin yetersiz kalacağı da bilinmelidir.

Editör, çok fazla özelliği olmayan bir kelime işlemci olarak düşünebilir (Notepad gibi). Program yazımlarını kolaylaştırması ve hızlandırması için daha gelişmiş editörler mevcuttur. Bu yazılımlara IDE (**Integrated Development Environment / Bütünleşik Geliştirme Ortamı**) adı verilmektedir. Şu an kullanıma sunulmuş pek çok IDE mevcuttur.

IDE'ler size kod yazmak için sadece editör imkânı sağlamaz. Aynı zamanda;

- Dosyalarınız arasında kolayca gezinme,
- Gelişmiş dosya işlemleri (bul, değiştir, satıra git vb.),
- Editör ortamını özelleştirme (renk, font, yerleşim vb.),
- Otomatik kod tamamlama,
- Kod renklendirme,
- Hazır geliştirme araçları,
- Gelişmiş hata ayıklama aracı,
- Kütüphanelere erişim kolaylığı vb. daha pek çok kullanım kolaylığı da sağlar.

Dolayısıyla Python ile gelen IDLE yerine, aşağıda bazıları sıralanan IDE'lerden kullanılması tavsiye edilir:

- Pycharm
- Spyder
- Eclipse + Pydev

- Sublime Text
- Visual Studio Code
- Vim
- Atom
- Jupyter vb.

IDE tercihinizi, internetten kısa bir araştırma ile belirleyebilirsiniz.

3.6.2. Kütüphane Kullanımı

Kütüphane kavramını, önceden yazılmış ve çok sık kullanılan kod parçacıklarını programın içine dâhil ederek o kod parçacıklarını kendiniz yazmışsınız gibi kullanabildiğiniz bir yapı olarak düşünebilirsiniz. Bazı kaynaklarda “kütüphane” kavramı “paket” veya “modül” olarak da geçmektedir.

Python ile gelen pek çok hazır kütüphane “**Standart kütüphaneler**” olarak isimlendirilir. Programlar içinde doğrudan kullanılabileceği gibi internetten veya çeşitli araçlarla rahatlıkla bulunabilecek, daha özelleşmiş fonksiyonları barındıran diğer kütüphaneler de kullanılabilir.

Aşağıdaki örnekte Python ile gelen örnek bir kütüphane programa dâhil edilip kullanıldığında 1-100 arası rastgele bir sayı oluşturulmaktadır.

```
import random
x = random.randint(1, 100)
print(x)
```

İlk satırdaki **import random** komutu ile programa **random** kütüphanesi eklenmiş olunur. Ardından **random.randint(1,100)** komutu ile 1-100 arası rastgele bir tam sayı oluşturulması sağlanır ve ekrana yazdırılır. “**Random**” kütüphanesi Python kurulumu ile geldiği için ekstra bir kurulum gerektirmez.

Örnek 1: 2 üzeri 3 ü hesaplayan bir program kodu yazınız.

```
import math
x = math.pow(2, 3)
print(x)
```

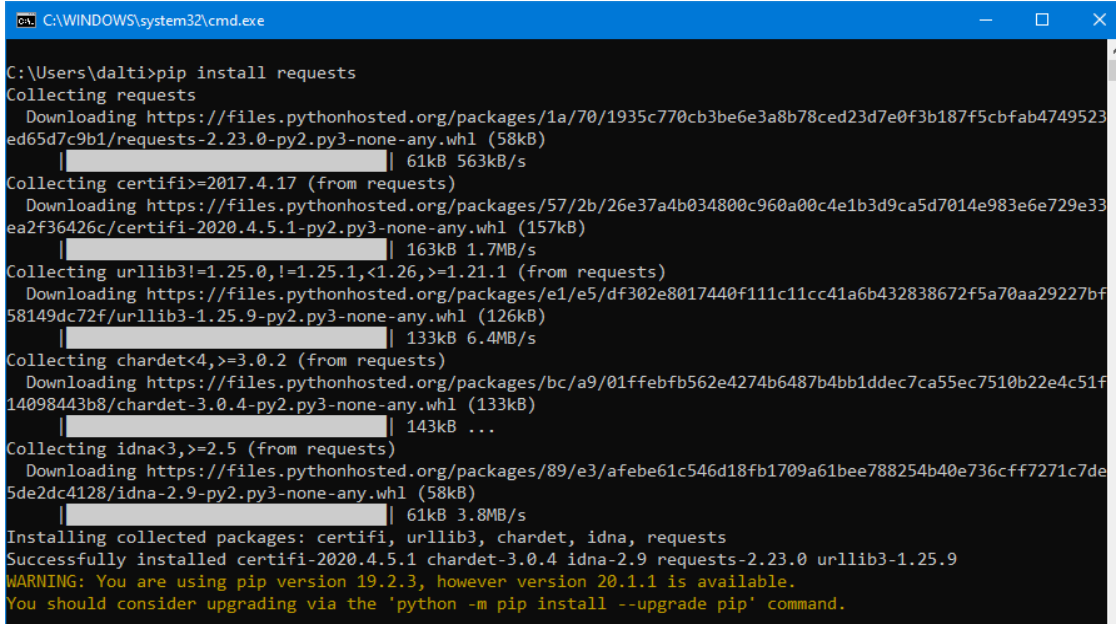
İlk satırda bu sefer **math** kütüphanesinin eklendiği görülmektedir. “**math**” kütüphanesi de Python kurulumu ile bilgisayara otomatik yüklenen kütüphanelerden biridir. **math.pow(2, 3)** komutu ile 2 üzeri 3’ün hesaplanması gerçekleştirilmiştir.

Python ile gelen standart kütüphanelerin tam listesi <https://docs.python.org/3/library/index.html> adresinden incelenebilir.

Öte yandan; Python kurulumu ile gelmeyen bir kütüphaneyi programda kullanabilmek için öncelikle ilgili kütüphane bilgisayara yüklenmelidir. Python kurulumu ile gelen “**PIP**” (Package Installer for Python / Python için paket yükleyicisi) programı kullanılarak bilgisayara Python kütüphaneleri yüklenebilir.

Örnek bir paket yüklemek için konsol ekranı kullanılarak aşağıdaki komut yazılır.

```
pip install requests
```



```
C:\WINDOWS\system32\cmd.exe
C:\Users\dalti>pip install requests
Collecting requests
  Downloading https://files.pythonhosted.org/packages/1a/70/1935c770cb3be6e3a8b78ced23d7e0f3b187f5cbfab4749523ed65d7c9b1/requests-2.23.0-py2.py3-none-any.whl (58kB)
    | 61kB 563kB/s
Collecting certifi<=2017.4.17 (from requests)
  Downloading https://files.pythonhosted.org/packages/57/2b/26e37a4b034800c960a00c4e1b3d9ca5d7014e983e6e729e33ea2f36426c/certifi-2020.4.5.1-py2.py3-none-any.whl (157kB)
    | 163kB 1.7MB/s
Collecting urllib3!=1.25.0,!1.25.1,<1.26,>=1.21.1 (from requests)
  Downloading https://files.pythonhosted.org/packages/e1/e5/df302e8017440f111c11cc41a6b432838672f5a70aa29227bf58149dc72f/urllib3-1.25.9-py2.py3-none-any.whl (126kB)
    | 133kB 6.4MB/s
Collecting chardet<4,>=3.0.2 (from requests)
  Downloading https://files.pythonhosted.org/packages/bc/a9/01ffebe61c546d18fb1709a61bee788254b40e736cff7271c7de14098443b8/chardet-3.0.4-py2.py3-none-any.whl (133kB)
    | 143kB ...
Collecting idna<3,>=2.5 (from requests)
  Downloading https://files.pythonhosted.org/packages/89/e3/afebe61c546d18fb1709a61bee788254b40e736cff7271c7de5de2dc4128/idna-2.9-py2.py3-none-any.whl (58kB)
    | 61kB 3.8MB/s
Installing collected packages: certifi, urllib3, chardet, idna, requests
Successfully installed certifi-2020.4.5.1 chardet-3.0.4 idna-2.9 requests-2.23.0 urllib3-1.25.9
WARNING: You are using pip version 19.2.3, however version 20.1.1 is available.
You should consider upgrading via the 'python -m pip install --upgrade pip' command.
```

Görsel 3.15: Kütüphane kurulumu

“Successfully installed ...” yazan satır, istenilen kütüphanenin doğru bir şekilde bilgisayara yüklendiğini ifade etmektedir. Artık **“requests”** kütüphanesi programda kullanılabilir.

```
import requests
```

```
icerik = requests.get('https://www.w3schools.com/xml/note.xml')
print(icerik.text)
```

Örnek program, belirtilen adresteki içeriği okuyup ekrana yazdırmaktadır.

Kullanışlı bazı Python kütüphane listelerine <https://wiki.python.org/moin/UsefulModules> adresinden erişilebilir.

Kütüphaneler, 6. öğrenme biriminde daha geniş bir şekilde işlenecektir.

ÖLÇME VE DEĞERLENDİRME 3

1. Python programlama dili ile “işletim sistemi” yazılabilir mi? Araştırınız.
2. Bir yazılımın niçin en son versiyonunun kullanılması gerektiğini açıklayınız.
3. Python programlama dilini kullanabileceğimiz IDE’lerini (Integrated Development Environment / Bütünleşik Geliştirme Ortamı) inceleyiniz. Hangilerinin tercih edilebileceğini açıklayınız.
4. Shell ekranında aşağıdaki komutları çalıştırınız ve bu komutların ne iş yaptıklarını karşlarına yazınız.
 - i. $10 + 20$:
 - ii. $4 * 30$:
 - iii. $2 ** 1000$:
 - iv. `print(“merhaba”)` :
 - v. $36 / 4 * (3 + 2) * 4 + 2$:
5. [] Python yorumlanan bir dildir. (Evet/Hayır)

NOT: Cevaplarınızı cevap anahtarıyla karşılaştırınız. Yanlış cevap verdiğiniz ya da cevap verirken tereddüt ettiğiniz sorularla ilgili konuları veya faaliyetleri geri dönerek tekrarlayınız. Cevaplarınızın tümü doğru ise bir sonraki öğrenme birimine geçiniz.