

Python Programlama

Ders 5

Ali Mertcan KOSE Msc.

`amertcankose@ticaret.edu.tr`

İstanbul Ticaret Üniversitesi



İSTANBUL TİCARET
ÜNİVERSİTESİ

Python, güçlü programlar oluşturmak için kullanılabilecek standart bir kütüphane içerir. Kütüphane, programınızda kullanmaya hazır olan bir kod koleksiyonudur. Standart bir kütüphane dilin bir parçası olarak kabul edilir ve herhangi bir Python sistemine dahil edilmelidir.

Python'un standart kütüphanesi modüller halinde düzenlenmiştir; Her modül ilgili işlevler içerir. Standart kütüphaneleri kullanmanın yanı sıra, kendi modüllerinizi de oluşturabilirsiniz, ancak matematiksel veya dize işlevleri için kendi uygulamalarınızı yazmanız nadiren gereklidir.

Python Library

Modüllerin Kullanımı- import

Bir modülde bulunan işlevleri kullanmadan önce, önce modülü programınıza aktarmalısınız.

İçe aktarma işlemi bir import deyimi kullanılarak yapılır.

import ifadesini kullanarak, bir modül içindeki belirli işlevleri veya modülde yer alan tüm işlevleri bir kerede içe aktarabiliriz.

Syntax:

`from math import sqrt` → makes available only sqrt in math

`from math import sqrt, sin, cos` → makes available specific functions listed

`from math import *` → makes available all function in math module

`import math` → makes available all functions in math module, whene

Kendi Modüllerimizi Oluşturma

Modüller problem ayrışmasına izin verir. Tüm programımızı tek bir dosyaya yerleştirmek yerine, programı modüllere ayırırız (farklı kaynak kodu python modülleri, bir programdan birden fazla dosyada kolayca bir program oluşturmamıza izin verir. Bir modül, python tanımları ve ifadeleri içeren bir .py dosyasıdır.

Dosyalar

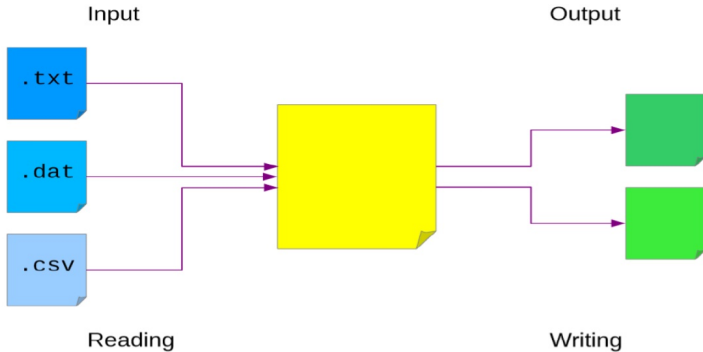


Figure 1: Dosya Detayları

Dosyalar

Function	Purpose
<code>open</code> (<code>fileName</code> , 'w')	Yazmak için yeni bir dosya oluşturur ve dosya nesnesini (dosya tutucu) döndürür.
<code>open</code> (<code>fileName</code> , 'r')	Mevcut bir dosyayı okumak için açar ve bir dosya tanıtıcısı döndürür.
<code>open</code> (<code>fileName</code> , 'a')	Mevcut bir dosyayı eklemek için açar ve bir dosya tanıtıcısı döndürür.
<code>fh.read()</code>	Dosyayla ilişkili dosyanın içeriğini içeren bir dize döndürür halletmek.
<code>fh.readline()</code>	Dosya tanıtıcısıyla ilişkili dosyadaki sonraki satırı döndürür.
<code>fh.readlines()</code>	Her bir ögesi dosyayla ilişkili dosyanın bir satırından oluşan listeyi döndürür halletmek.
<code>ffhçfh.write(s)</code>	s dizesini dosya tanıtıcısıyla ilişkili dosyanın sonuna yazar.
<code>fh.writeLines(S)</code>	S bir dizi dizisidir, S'nin her elemanını dosyaya ayrı bir satır olarak yazar dosya tanıtıcısı ile ilişkilidir.

`fh.close()` | Dosya tanıtıcısıyla ilişkili dosyayı kapatır.

Verileri Dosyaya Yazma

writing data to a file

```
fileHandle = open( '05_file1.txt', 'w')
```

```
for i in range(2):
```

```
    name = input('Enter name:')
```

```
    fileHandle.write( name+'n')
```

```
fileHandle.close()
```

File Contents:

Melisa Aksoy

Will Smith

Verileri bir dosyaya eklemek

append data to end of file

```
fileHandle = open( '05_file1.txt', 'a')
```

```
fileHandle.write('Jane Doen')
```

```
fileHandle.close()
```

File Contents:

Melisa Aksoy

Sam Smith

Jane Doe

Read() Dosyasından Veri Okumak

reading data from a file

```
read()
```

```
fileHandle = open( '05_file1.txt', 'r')
```

```
fileContents = fileHandle.read()
```

```
print(fileContents)
```

```
fileHandle.close()
```

Dosyadan Satır Okumak okuma satırı()

```
#reading specific lines from a file  
readline()
```

```
## [1] ""
```

```
fileHandle = open( '05_file1.txt', 'r')
```

```
for i in range(2) :
```

```
line = fileHandle.readline()
```

```
print(line[:1])
```

```
fileHandle.close()
```

Not:

Yazdırma yönteminde [1] sözdizimini kullanma, yazdırma yönteminin Dosya verilerinden NewLine karakterini çıkarmasını durdurur. Önde gelen ve sondaki boşluk karakterlerini kaldıran ve yeni dizeyi döndüren String Strip () işlevini de kullanabiliriz.

Bir Dosyadan Veri Okuma for döngüsü

#read all lines from the file

```
fileHandle = open( '05_file1.txt', 'r')
```

```
for line in fileHandle:
```

```
    print(line[:1])
```

```
fileHandle.close()
```

Yararlı Dize İşlevleri

Dosyaları işlerken yararlı olabilecek çeşitli yerleşik dize işlevleri vardır.

Function	Purpose
s.count(s1)	s1 dizesinin s içinde kaç kez geçtiğini sayar.
s.find(s1)	s1 alt dizesinin s cinsinden ilk geçtiği yerin dizinini döndürür, ve -1 eğer s1 s'nin içinde değilse.
s.rfind(s1)	Find ile aynıdır ancak s'nin sonundan başlar (rfind'deki 'r', ters için)
s.replace(old,new)	Eski dizenin s içindeki tüm oluşumlarını dizıyla değiştirir yeni.
s.strip()	S'nin başındaki ve sonundaki boşlukları kaldırır.
s.rstrip()	s'nin sonundaki boşlukları kaldırır.

- 1 Bir dosyadan otel verilerini okuyan ve kullanıcı tarafından girilen şehirdeki tüm otelleri yazan bir program yazınız. oteller.txt adlı yeni bir dosyaya kopyalayın. Not: Şehir adının şu adreste olduğunu varsayabilirsiniz:her satırın başında yer alır ve şehrin ardından bir tire (-) ve ardından otel adı gelir.
- 2 Çiçeklerin adlarını içeren iki sıralı dosyayı giren ve ikisini birleştiren bir program yazın dosyaları yeni bir sıralanmış dosyaya dönüştürün.
- 3 İki ülke veri dosyasını (05_worldpop.txt) okuyan bir program yazın ve 05_worldarea.txt. Her iki dosya da aynı ülkeleri aynı sırayla içeriyor. Ülke adlarını ve nüfus yoğunluklarını (km2 başına düşen kişi sayısı) içeren bir world_pop_density.txt dosyası yazın.