

Python Öğren 4 Saatte Python'a Başlangıç

Python Nedir?

Python; veri biliminden web uygulaması geliştirmeye, API oluşturmaktan sistem yönetimine kadar hem kişisel bilgisayarlarda, hem sunucu dünyasında hem de gömülü ve mobil cihazlarda kullanım alanı bulan, öğrenmesi kolay ve kod geliştirirken işleri oldukça hızlandıran bir dil.

Neden Python Öğrenelim?

- Yüzbinlerce kütüphaneye sahip
- Öğrenmesi ve yazımı kolay
- Okuması rahat
- Günlük işler için çok kısa sürede çözüm üretmeye müsait

Başlangıç

Kurulum

Bilgisayarlarımızda Python'un kurulu olduğundan emin olmamız gerekiyor. Bunun için, <https://www.python.org/downloads/> adresinden işletim sistemimize uygun olan sürümü indiriyoruz.

Kurulum aşamaları indirdiğimiz dosyayı açtığımızda çıkan ekranı takip ettiğimizde tamamlanıyor. Kod yazmaya hazırız!

Test Edelim

Python'u test edebilmek için işletim sistemlerine göre birkaç yöntem mevcut. Onlara göre bakalım:

GNU/Linux ve MacOS

Ubuntu, Fedora, OpenSuse gibi bir GNU/Linux dağıtımında çalışıyorsanız, tek yapmanız gereken bir terminal açıp `python3` yazmak ve enter'a basmaktan ibaret. Eğer hata almak yerine şöyle bir ekran görüyorsanız hazırsınız:

```
→ ~ python3
Python 3.7.6 (default, Jan 5 2020, 20:31:56)
[Clang 11.0.0 (clang-1100.0.33.16)] on darwin
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

Windows

Windows'ta kurulum yönteminize göre, PowerShell'i açıp `python` yazmak ve Enter'a basmak yeterli olur. Ayrıca bir arayüz kullanmak isterseniz uygulamalar arasından IDLE'yi açabilirsiniz.

İlk Komutlar

Python'un bizimle etkileşime geçtiği yer kendi komut satırı.

Python'un bizden bir şeyler beklediğini `>>>` yazmasından anlayabiliriz.

```
>>>
```

Burada istersek matematiksel işlemleri yapabiliriz. Mesela elimizde yer alan fatura ödemelerinin toplamı için bir hesap makinesine ihtiyacımız olsun:

```
>>> 35.4 + 90 + 105.2
230.60
```

Sondaki çıktının noktadan sonraki kısmı daha uzun gözükebilir. Onu şimdilik görmezden geelim ve devam edelim.

Nasıl Çıkarız?

Python'a bir şeyler yazdık ama onu nasıl kapatıp çıkacağımızı henüz görmedik. İsteddiğimiz hesaplamayı yaptıktan sonra, Python'dan çıkmak için `exit()` yazıp entera basmamız yeterli.

`exit()` yazmamız yazılım dünyasında `exit` fonksiyonunu çalıştırdık. Bu fonksiyon Python'u kapatıp çıkıyor olarak geçiyor.

Kütüphaneler & Kütüphane/Kitaplık Nedir?

Python'da bizim kullanabilmemiz için uzun uzun, detaylıca yazılmış hazır kodları kullanmamızı sağlayan birkaç teknoloji var. Biz de olabildiğince bu nimetlerden faydalanıp, günler hatta aylar harcayıp kendimiz yazmak yerine varolanı kullanıyoruz. Mesela hava durumu için bir websitesine girip ordan hava durumunu bulacak kodu ya da Ethereum'un satış fiyatını alıp bize söyleyecek kodu uzun uzun yazmak yerine, yazılmış bir tanesini alıp kullanabiliyoruz.

Bunlara ise Python dünyasında `modül` deniyor.

Kütüphane Kurulumu

Python'a yeni kütüphaneler(modüller) eklemek için `pip` komutunu kullanıyoruz. Bir terminal açıp:

```
pip install requests
```

yazmamız yeterli. Bu örnekte kurduğumuz modülün adı **requests** oldu.

Kütüphane Kullanımı

Python'da bir kitaplığı kullanmak istersek, isminin başına `import` yazmamız yeterli.

```
>>> import requests
```

API Ne İşe Yarar? Örnek API

Mesela, CollectAPI'den döviz fiyatlarını ve alakalı bazı bilgileri almak istersek kendilerinin API'si olan <https://collectapi.com/tr/api/economy/altin-doviz-ve-borsa-api> adresini ziyaret edebiliriz.

API Kullanımı

Şimdi Python'da bu web sayfasındaki içeriği nasıl görebileceğimize bakalım.

```
>>> import requests
>>> requests.get('https://api.collectapi.com/economy/allCurrency')
<Response [200]>
```

En sondaki 200, isteğin başarılı sonuçlandığı anlamına geliyor. Peki daha detaylı bilgi nasıl alırız?

Elde ettiğimiz sonucun detaylarına şöyle ulaşabiliriz:

```
>>> requests.get('https://api.collectapi.com/economy/allCurrency').text
...
>>> requests.get('https://api.collectapi.com/economy/allCurrency').headers
...
```

Bu satırlarda her seferinde tekrardan bir bağlantı kurup, karşı tarafa o anki değerleri sorduk. Bu çoğu zaman verimli bir yöntem değil. İş kolaylaştırmak için, sadece bir kere `requests.get(...)` yazdıktan sonra elde ettiğimiz sonucu saklamaya ihtiyacımız var.

Değişkenler ve Temel Veri Tipleri

Değişken, bilgisayar belleğinde saklanan bir değeri temsil eder. Örneğin;

```
ad = "Aylin"
yas = 26
```

`ad` ve `yas` isminde 2 değişken oluşturduk ve "=" koyarak bu değişkenlere değer atadık.

Python standartlarına göre temel veri tipleri aşağıdaki şekildedir:

1. String (Karakter Dizisi): Karakter dizilerini belirtmek için kullanılır. Tek veya çift tırnak kullanılarak tanımlanır. Örnek: `para_birimi = "bitcoin"`
2. Number (Sayı): Tam sayı, ondalıklı sayı v.b. nümerik değerler için kullanılır. Örnek: `para_miktari = 50`

3. List (Liste): Bir veya birden fazla değeri listelemek için kullanılır. Köşeli parantez ile tanımlanır. Örnek:
`para_birimleri = ["bitcoin", "ethereum", "dolar"]`
4. Boolean (Mantıksal değer): Doğru ya da yanlış olmak üzere iki değere sahiptir. Bu iki değer, `True` ve `False` olarak tanımlanır. Örnek: `kart_gecerli_mi = True`
5. Tuple (Demet): Yapı olarak listeye benziyor. Listedenden farklı olarak değerleri değiştirilemez. Normal parantez ile tanımlanır. Örnek: `para_birimleri = ("bitcoin", "ethereum", "dolar")`
6. Dictionary (Sözlük): Anahtar-değer(key-value) ikilisi şeklindeki değerlerin kümesidir. Küme parantezi ile gösterilir. Her bir sözlük elemanı `anahtar:değer` şeklinde tanımlanır. Örnek:
`{'para_birimi': 'TL', 'miktar': 1000}` `para_birimi` anahtarının değeri `TL`, `miktar` anahtarının değeri ise 1000 olan iki elemanlı bir sözlük tanımladık.

Veriyi Saklamak

Python'da bir veriyi saklamak için, ona bir isim vermemiz ve araya "=" koymamız gerekiyor:

```
>>> para_birimleri =  
requests.get('https://api.collectapi.com/economy/allCurrency')  
>>> para_birimleri.text  
...  
>>> para_birimleri.headers  
...
```

Artık veriyi ilk seferinde sakladık. `para_birimleri` yazarak ulaşabiliriz.

Bu Veri Neden Çirkin Gözüküyor? JSON Nedir?

Verinin görüntüsü aslında bilgisayarın kolay anlayabileceği, bizim de biraz güzelleştirdikten sonra rahatça algılayacağımız bir biçime sahip. Burada kullanılan notasyona JSON (JavaScript Object Notation) deniyor.

```
{"result": [{"name": "Dolar", "code": "USD", "buying": 6.5791, "buyingstr":  
"6.5791", "selling": 6.5845, "sellingstr": "6.5845", "rate": 2.07, "time":  
"20:40:29", "date": "2020-03-30", "datetime": "2020-03-30T17:40:29.000Z",  
"calculated": 0}, {"name": "Euro", "code": "EUR", "buying": 7.2483,  
"buyingstr": "7.2483", "selling": 7.2543, "sellingstr": "7.2543", "rate":  
0.9, "time": "20:40:28", "date": "2020-03-30", "datetime": "2020-03-  
30T17:40:28.000Z", "calculated": 0}, {"name": "\u0130ngiliz Sterlini",  
"code": "GBP", "buying": 8.1341, "buyingstr": "8.1341", "selling": 8.1432,  
"sellingstr": "8.1432", "rate": 1.25, "time": "20:40:29", "date": "2020-03-  
30", "datetime": "2020-03-30T17:40:29.000Z", "calculated": 0}, {"name":  
"\u0130svi\u00e7re Frang\u0131", "code": "CHF", "buying": 6.8484,  
"buyingstr": "6.8484", "selling": 6.8563, "sellingstr": "6.8563", "rate":  
0.97, "time": "20:39:41", "date": "2020-03-30", "datetime": "2020-03-  
30T17:39:41.000Z", "calculated": 0}, {"name": "Kanada Dolar\u0131", "code":  
"CAD", "buying": 4.6441, "buyingstr": "4.6441", "selling": 4.649,  
"sellingstr": "4.649", "rate": 0.8, "time": "20:40:09", "date": "2020-03-  
30", "datetime": "2020-03-30T17:40:09.000Z", "calculated": 0}], "success":  
true}
```

Yukarıdaki JSON objesini daha anlaşılır biçimde görmek için formatlandırmak istersek, Visual Studio Code gibi bir uygulamayı veya internetten bulacağımız bir online JSON formatter aracını kullanabiliriz. Alternatif olarak <https://codebeautify.org/jsonviewer> sitesinden de faydalanabiliriz. Bu şekilde baktıktan sonra ihtiyacımız olan fiyatlandırmanın nerede olduğunu bularak yolumuza devam edebiliriz.

Python'da ihtiyacımız olan kısmı almak için:

```
>>> para_birimleri.json()  
...
```

Tümünü bu şekilde elde ettik. Şimdi içinden istediğimiz kısmını alalım. Önce kaç tane veri olduğuna bakalım:

```
>>> len(para_birimleri.json())  
5
```

Bu veri türüne Python'da `list` deniyor. Toplam 5 adet para birimi tablosunu bir arada gördüğümüz anlamına geliyor. JSON görselleştiricide baktığımızda gördüğümüzle aynı kapağa çıktık.

İlk sıradaki tabloyu görmek istersek:

```
>>> para_birimleri.json()["result"][0]  
...
```

Burada satır gittikçe karmaşılaşmaya başladı. Bunu basitleştirmek için yine bir isimlendirme kullanabiliriz.

```
>>> tumbirimler = para_birimleri.json()["result"]  
>>> tumbirimler[0]  
{'name': 'Dolar', 'code': 'USD', 'buying': 6.5775, 'buyingstr': '6.5775',  
'selling': 6.5813, 'sellingstr': '6.5813', 'rate': 2.02, 'time':  
'20:59:14', 'date': '2020-03-30', 'datetime': '2020-03-30T17:59:14.000Z',  
'calculated': 0}
```

Peki gelen bu süslü parantezle başlayıp bitenler ne? Bunlar Python'da `dictionary` olarak adlandırılıyor. Mesela, `code` alanında yazan değeri almak istersek:

```
>>> tumbirimler[0]["code"]  
USD
```

Uygulamalar

Uygulama olarak bir Telegram botu yazacağız. Bunun için öncelikle terminal ekranından aşağıdaki komut ile Telegram paketini yerel sistemimize kurmalıyız:

```
pip install python-telegram-bot
```

Artık uygulamaya geçebiliriz. Şimdi yapacaklarımıza sırayla göz atıp botumuzu yazmaya başlayalım:

1. Belirli aralıklarla bu değeri almak
2. Dolar değeri Euro değerinden büyük mü diye bakmak
3. Telegram'dan bot oluşturmak -> BotFather -> <https://api.telegram.org/bot<TOKEN>/getUpdates>

NOT: Bu adrese yaptığımız istek ile Telegram botumuzla ilgili tüm gelişmeleri gözlemliyoruz. Örneğin; biri botunuza mesaj attığı zaman mesaj atan kişinin adını, ne mesaj yazdığını ve chat_id değerini buradan görebilirsiniz.

4. Python ile bota mesaj göndermek

```
>>> import telegram
>>> bot = telegram.Bot(token='BOT_TOKEN')
>>> bot.send_message(USER_ID, "Merhaba!")
```

5. Dosyaya uygulamayı koymak
6. Dolar değeri Euro değerinden büyükse botun mesaj göndermesi

```
import requests
import telegram
import time

bot = telegram.Bot(token='TELEGRAM_TOKEN_BURAYA')

base_currency = 6

while True:
    response = requests.get(f"https://api.collectapi.com/economy/exchange?int={base_currency}&to=USD&base=EUR",
        headers = {
            "content-type": "application/json",
            "authorization": "apikey COLLECTAPI_TOKEN_BURAYA"
        }
    )

    json = response.json()
    exchange_data = json["result"]["data"][0]
    calculated_dollar = exchange_data["calculated"]

    if calculated_dollar > base_currency:
        bot.send_message(TELEGRAM_CHAT_ID_DEGERI, f"{str(base_currency)} Euro'nun Dolar karşılığı {str(calculated_dollar)} $ dır")
        time.sleep(30)
```

Python Öğren - Python'a 4 Saatte Başlangıç
Eğitmenler: Güray Yıldırım, Aylin Gümüş, Muhammed Taha Ayan