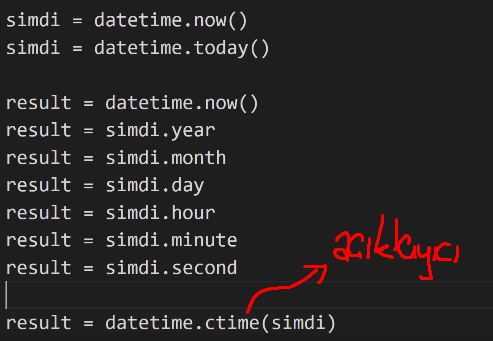
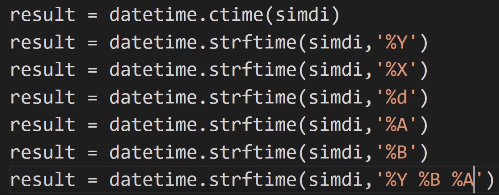
import **datetime** [**Kaynak**](https://docs.python.org/3/library/datetime.html) **🡪 Komutlar ve daha fazlası için**

“import datetime” komutu ile zamanla ilgili olan fonksiyonlara imkan sağlarız (tarih/zaman). Bu modül 3 farklı alana ayrılıyor “time, date, datetime”, bu alanları “from datetime import date” gibi çağırabileceğimiz gibi yalnızca “import datetime” diyip “datetime.date” gibi de belirtebiliriz.

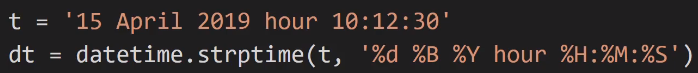
“datetime.now()” ile güncel zaman bilgisine ulaşırız. Bu fonksiyonun sonuna eklenmiş zaman kavramları (“.hour” gibi) bize integer olarak istenilen zamanı verir.



“.ctime(some)” fonksiyonunun belirli değerlerini çekmek istersek “.strftime(some, “%x”)” komutunu kullanabiliriz.



“datetime.strptime(time, …)” komutunu örnekteki gibi uyguladığımız takdirde time parametresinin zaman değerlerini kod düzenlenmiş şekilde bize sunar.



“some = datetime(…)” komutunun içerisine yıl, ay, gün, saat, dakika, saniye bilgileri sırayla girilerek sisteme entegre edilebilir, bu belirlenen parametreyi ise “.timestamp(some)” komutu ile saniye cinsinden yazdırabiliriz (1970-01-01’den itibaren), “.fromtimestamp(somee)” komutu ile de saniye cinsindeki değeri tekrar tarihe çevirebiliriz. Bu parametredeki değeri “.days, .seconds, .microseconds” komutu ile belirli zaman değerlerine dönüştürebiliriz.

“timedelta(days = 15, minutes = 25)” (parantez içi değişir farklı parametre çıkarılır/eklenir) komutu ile belirli tarihe zaman ekleme çıkarma gibi işlemler yapılabilir. Anladığım kadarıyla “from datetime import timedelta” library’i koda çağırmamız gerek (dene).

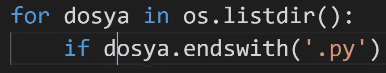
import **os**

“import os” modülü ile modülü aktive ediyoruz. Bu modül işletim sistemi üzerinde yetkiler tanır. “os.name” ile işletim sisteminin ismini öğrenebilirken, “os.chdir(“C:\\”)” komutu ile kod dosyasının konumunu belirli bir alanda gösterir, “os.mkdir(“newdirectory”)” ile C bölgesine kod dosyasını konumlandırabiliriz. “os.chdir(“..”)” komutu ile bulunduğumuz dosyanın bir üst dizinine geçeriz, en üst dizine kadar bu işlem tekrarlanabilir (“os.chdir(“../..” aynı işlemi görür). “os.getcwd()” komutu dosyanın güncel komutunu gösterir.

“os.makedirs(“newdirectory/newfile”)” ile newdirectory dizinine newfile klasörünü ekleyebiliriz. “os.rename(“newdirectory”, “newfile”)” komutu ile newdirectory isimli objenin ismini newfile olarak değiştirdik. “os.rmdir(“newfile”)” komutu ile newfile objesini silebiliriz. “os.removedirs(“newfile/f”)” komutu ile newfile objesi içerisindeki f klasörünü silebiliriz..

“os.listdir()” komutu ile kod dosyasında kullanılan dizin içerisinde etkin olan klasörler gösterilir, parantez içerisine bir alan girdiğimiz takdirde (örneğin C:\\) bu alandaki klasörleri bize liste olarak gösterir.

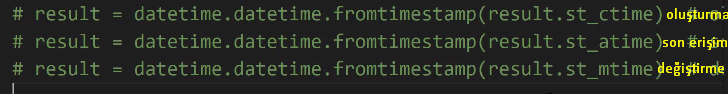
Örnekteki yöntem ile “.py” (python klasörü) ile biten dosyaları ekrana yazdırabiliriz.





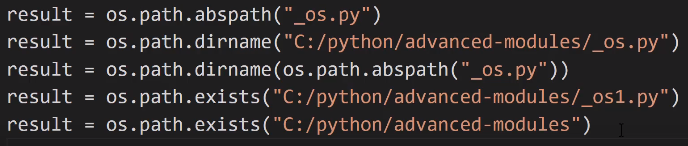
“os.stat(“filename.py”)” ile istenilen dosyanın özelliklerine oluşabiliriz, bu özellikle dosya boyutu, oluşturulma tarihi gibi nitelikleri içerir.

Bu komutu “os datetime” modülü ile birleştirerek anlamlı değerler elde edebiliriz. İki modülü beraber kullanmamız gerekir çünkü tüm statlar saniye cinsinden verilir. Örnekteki komutları kullanarak dosya hakkında bilgi sahibi olabiliriz.

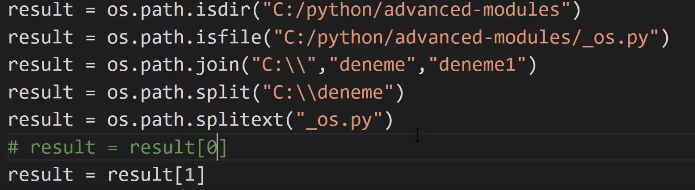


“os.system(“file”)” komutu ile sistemdeki istediğimiz klasörü açabiliriz.

Örnekteki ilk komut hedef olarak gösterilen dosya isminin dizinini bize verir, üçüncü komut ise bu dizin ismini biraz daha farklı biçimde gösterir. “os.path.exists(“…”)” komutu ile hedefteki dosya veya dizinin var olup olmadığı True/False olarak gösterilir.



“os.path.isdir(“…”)” komutu bize hedef objenin klasör olup olmadığını True/False olarak gösterir, aynı şekilde “.isfile(“…”)” ise hedefin dosya olup olmadığını sorgular. “os.path.join(“…”)” komutu ile dosya ve klasörleri aynı dizinde birleştirebiliriz. “os.path.split(“…”)” komutu ile birleşik dizini parçalara ayırabiliriz. “.splitext(“file”) komutu ile de dosyanın uzantısını isminden ayırabiliriz, aşağıdaki örnekte değerleri ayrı ayrı yazdırdık.



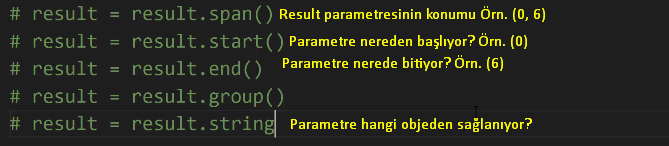
import **re** [**Kaynak**](https://www.w3schools.com/python/python_regex.asp) **🡪** Metakarakter ve Special Sequences’e ulaşılabilir

“\s” bu libraryde “boşluk” anlamına gelir.

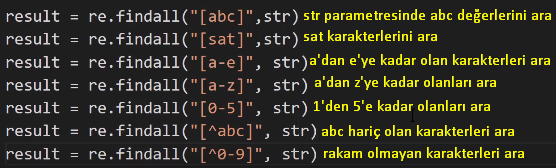
“import re” komutu ile regular expression kütüphanesini aktive ederiz. “re.findall(some)” komutu ile “some” stringini ifademizde bularak liste olarak geri gönderir. “re.sub(“x”,”y”,param)” komutu ile param komutunda bulunan bütün x harflerini y ile değiştirebiliriz.

“re.search(”some”, param)” komutu ile param içerisindeki “some” sözcüğünü match ederek kelimenin konumu elde ederiz. (Örn. Span = (0 ,6)).

Örnekteki fonksiyonlar re içerisinde kullanabileceğimiz diğer birimlerdir, “.group” ile işlem yaptığımız kelimeyi çağırabiliriz.



Örnekte “.findall“ metodunun farklı uygulama stilleri gösterilmiştir.

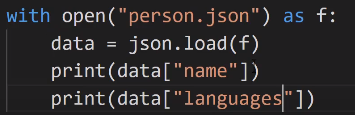


[**Kaynak**](https://www.btkakademi.gov.tr/portal/course/player/deliver/s-f-rdan-ileri-seviye-python-programlama-5877) 🡪 BTK Akademi linki

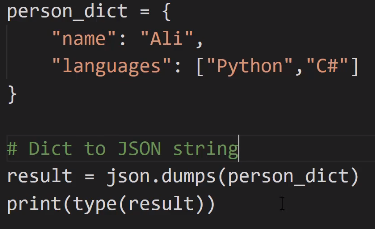
import **json**

dictionary biçiminde yazılmış bir ifadenin baş ve sonuna denden eklenildiği takdirde bu ifade json dictionary haline gelir ve kullanılabilir olur, bu ifadeyi kullanılabilir kılmak için “json.loads(param)” komutu kullanılır (bu ifade normalde yalnızca bir stringdir).

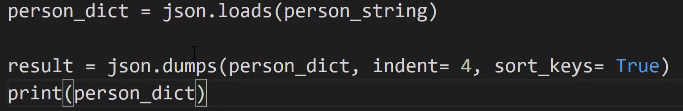
Json ile oluşturulmuş dictionary başka bir dosyadan da çekilebilir, örnek olarak:



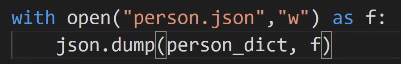
Dictionary olan bir ifadeyi json **stringe** çevirmek için ise aşağıdaki ifadeyi kullanabiliriz.



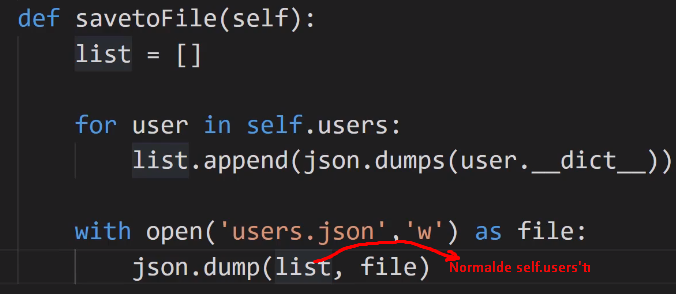
“indent = number, sort\_keys = True” gibi komutlarla person\_dict stringinin görüntüsünde değişim yapabiliriz.



“json.dump(dict, file\_pointer)” komutu ile json ifademizi istediğimiz dosya içerisine yazdırabiliriz.



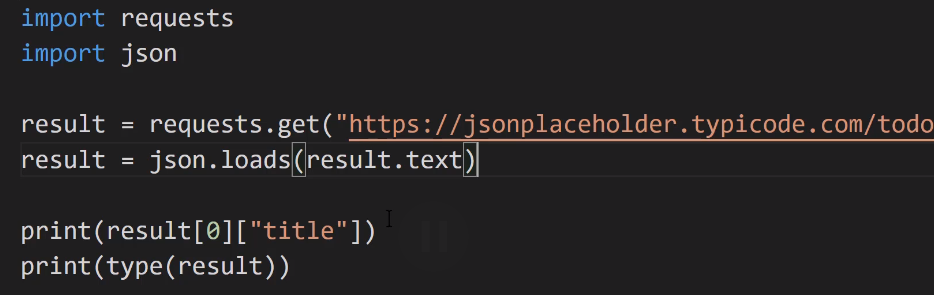
Json farklı platformlarda geliştirilen uygulamalar arasındaki veri alışverişinde sıklıkla kullanılmaktadır. Json metodu class bilgisini desteklemez (bilgi jsonda class olarak depolanamaz), bundan dolayı class olarak oluşturulan bilgiler dictionary’e dönüştürülmelidir.



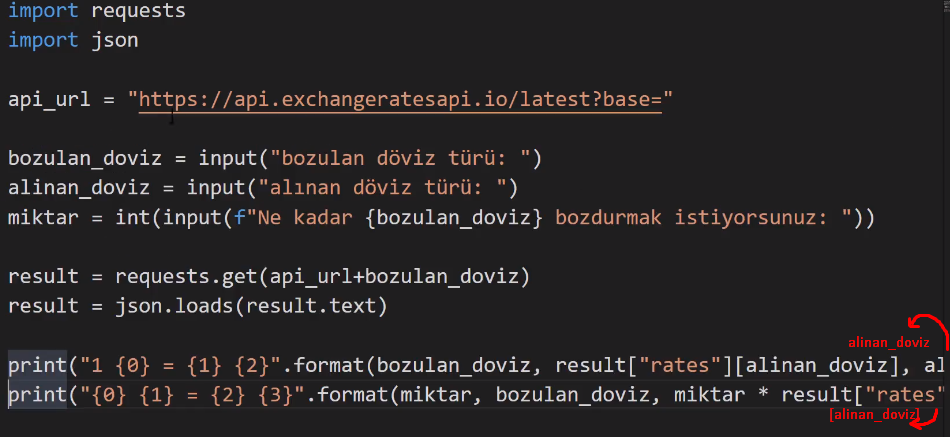
import **requests**

“print(module.\_\_file\_\_)” komutu ile istenilen modülün bilgisayardaki konumu yazdırabiliriz. Request modülü bilgisayarda default olarak bulunmamaktadır, “pip install requests” komutu ile bilgisayara yükleyebiliriz.

Request modülü ile bir internet sayfasının kaynak kodunu python aracılığıyla alabiliriz. Bunu da “requests.get(link)” komutu ile cevabı alırız, <Response [200]> cevabı işlem başarılı anlamını taşır ve bilgiler çekilebilirse bu gösterilir. “param = param.text” ile sitenin gönderdiği kodu yazdırabiliriz.

Bir web sitesinden çektiğimiz bilgiler string tipinde koda yazdırılır, bu bilgileri işlem yapmaya uygun hale getirmek için (örneğin “title” parametresinin karşısına gelen değeri çekeceğiz) json modülü kullanılır. 

“json.loads(result.text)” ile kolaylıkla texti, json string haline döndürebiliriz. Request modül örneği:



Repository içeren github veri alışverişi kodu için 🡪 [**Kaynak**](https://www.btkakademi.gov.tr/portal/course/player/deliver/s-f-rdan-ileri-seviye-python-programlama-5877), bu video aynı zamanda site ile etkileşim için önemli bir kaynak github etkileşimi için [**Kaynak**](https://docs.github.com/en/rest)

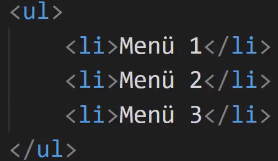
Film Arşiv Uygulaması (API) için 🡪 [**Kaynak**](https://www.btkakademi.gov.tr/portal/course/player/deliver/s-f-rdan-ileri-seviye-python-programlama-5877)

“.content” komutu ile requesti’i kullanarak linkteki bütün bilgileri çekebiliriz (html kodunu).

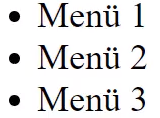
**BeatifulSoup** Eklentisi için temel **HTML** Bilgisi

Aşağıda oldu gibi html sisteminin genel prensibi <…> \_\_\_\_\_ </…> şeklinde açılıp kapanmaktadır. <head> komutu anladığım kadarıyla sekme üzerinde yapılacak değişimleri içerir. <body> komutu ise sayfamızın içeriğini oluşturmaktadır. Bu komutun alt unitesi olan <h1> komutuna ise body kısmının head’i diyebiliriz. <h1> komutu (Google web kurallarına göre) bir defa kullanılmalıdır.



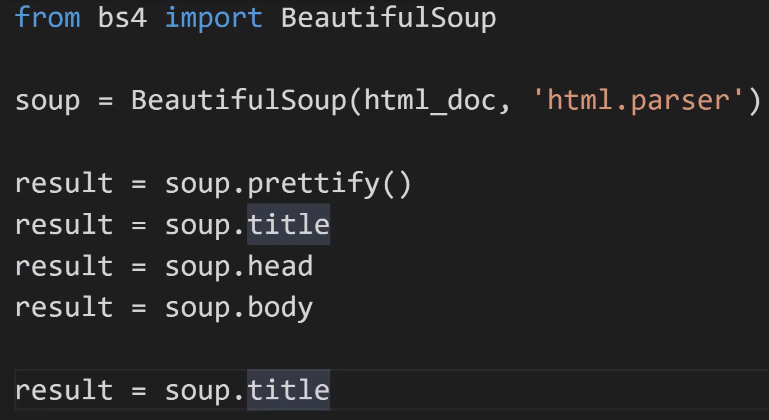
<ul> komutu ile liste oluşturabiliriz bu liste bullet pointlerle gösterilir. <ul> komutu bu noktada liste grubunu belirtirken <li> komutu ise bullet pointlere karşılık gelir. 

<h2> komutunu kodumuza ekleyerek h1’in altına ayrı bir başlık ekleyebiliriz.

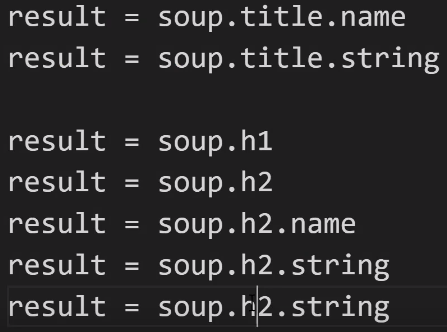


<div> komutu ile grup oluşturabiliriz, bu oluşturduğumuz gruplar sonradan aynı kod satırlarını kullanmada veya gruba bir parametre atayarak bu parametre üzerinde değişimler yapmak adına oldukça kullanışlıdır. Parametre atamak için ise <div class = “group1”> komutu kullanılır. “id = “header” komutu da aynı işlevi görür fakat id komutu bir defa kullanılabilirken class komutu istenildiği kadar kullanılabilir.

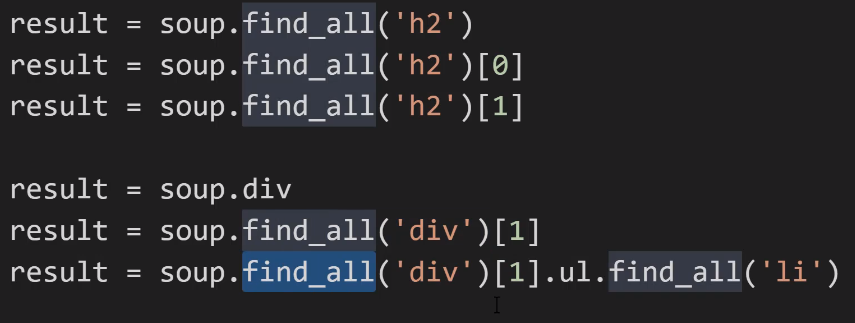
From **bs4** import **BeautifulSoup**



“.prettify()” komutu ile bozulmuş html kodu (pythonla alakalı boşluk işlemleri vs) düzeltilir. “.title” komutu ile title yazdırılır, “.head” ile head kısmı ve “.body” kısmıyla body görüntülenebilir.

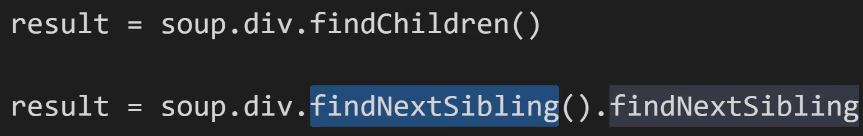


“.name” komutu ile h2’nin name’i (burada direkt h2 yazdırılır) yansıtılırken “.string” komutu ile h2 içerisine yazdığımız karakterler görüntülenir.



“.find\_all(“some”)” komutu ile değişken içerisindeki tüm değerleri list halinde yazdırabiliriz, sonuna “[0]” gibi index ekleyerek list içerisindeki belirli değeri çekebiliriz. Eğer belirli indexin içerisindeki belirli değeri çekmek istiyorsak bu komutun sonuna (örnekteki gibi) istenilen değişkeni ekleyip tekrardan “find\_all()” komutunu kullanarak değişkenin değişkeninin indexini dahi bulabiliriz.

Bu library içerisinde children dediğimiz bir kavram bulunuyor, bu kavram htmldeki bir kod bölümünün alt birimleri anlamına gelir.



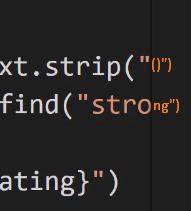
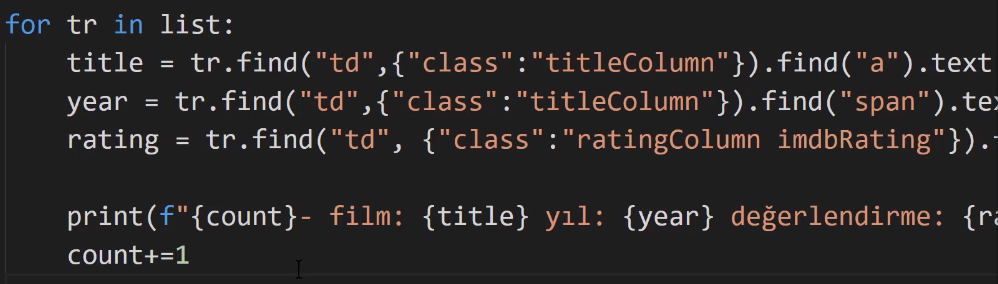
“soup.some.findChildren()” komutu ile bir kod bölümünün tüm alt birimlerini yazdırabiliriz. “.findNextSibling()” ve “.findPreviousSibling()” komutları ile de önceki ve sonraki kardeşe (altbirime) geçiş yapabiliriz.



[**Kaynak**](https://www.crummy.com/software/BeautifulSoup/bs4/doc/) **🡪** Daha fazla komut ve ayrıntı için bs4 komutları

“.find(“area1”, {“class”:”area2”}” komutu ile





Yukarıdaki örnekte ilk olarak url’yi girdik, ardından “html” değişkenini url’nin content kısmını (bütün html bilgisini) alacak şekilde birbirleriyle eşitledik. “soup” değişkeniyle html’i Soup kütüphanesine entegre ettik ve bu sayede “html.parser” şeklinde veriyi alabiliriz. Bunun ardından for döngüsünün içine tanımladığımız “title” değişkeniyle tr classı içerisinden a etiketinin text kısmını çektik. İkinci parametremiz “year” ile aynı tr classı içerisindeki farklı bir td’den span etiketi altındaki text kısmını parantezleri “.strip(“()”)” komutuyla çıkararak aldık. Son değişken “rating” ile tr classındaki strong etiketi altındaki text kısmını almış olduk. Sonunda bu değerleri fstringi kullanarak yazdırdık.

import **numpy**

Bu kütüphane veri analizinde kullanılır ve oldukça popülerdir. “import numpy as p“ genel kullanım şeklidir “np” her şey olabilir.

Bir listeyi normalde yapabildiğimizden çok daha farklı fonksiyonlarla manipüle edebilmek için numpy kütüphanesini kullanabiliriz. Bu fonksiyonlar matematiksel işemlerde oldukça işlevlidir.

“np.array([…])” komutuyla bir dizi oluşturabiliriz. Bu komutla eşitlediğimiz bir değişkeni “.reshape(col, row)” komutuyla birleştirerek istenilen boyutta bir matrix elde edilebilir. “.ndim” komutuyla bir listin kaç boyutlu olduğunu görebiliriz. “.shape” komutuyla ise bir listin kaça kaç olduğunu görüntüleyebiliriz.

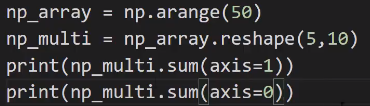
“.arrage(1,10)” şeklinde kullanarak 1’den 9 a kadar olan bir array oluşturabiliriz, komutun sonuna bir parametre daha eklediğimiz takdirde bu parametre de artış miktarını belirtir (3 yazdığımızı düşünürsek 1,4,7,…).

“.zeros(10)” komutu ile 10 değerden oluşan sıfır matrisi oluşur, “.ones(10) ile de bu matrixin birden oluşan halini oluşturtabiliriz. Belirli bir aralıkta belirtilen eş parçaya ayrılmış bir liste istersek “.linspace(x, y, z)” komutu ile bu işlemi yapabiliriz (x başlangıç, y bitiş, z eşit parçaya bölünesi değer). “.linspace” metodunda normalden farklı olarak max değer de hesaba katılır.

“np.random.randint(1, 10)” komutu ile 1 ile 9 arası integer olan bir değer üretilir, tek değer yazarsak 0 dan o sayıya kadar olan değerler belirlenir. Bu komutun sonuna eklenecek üçüncü bir parametre üretilecek random sayı miktarını belirler, ortaya çıkan değer liste olarak yazdırılır.

“np.random.rand(x)” metodu ile sıfırdan bire kadar olan x tane sayı list olarak yazdırılır “.randn()” varyasyonu ile de bu sayılar negatif olabilir.

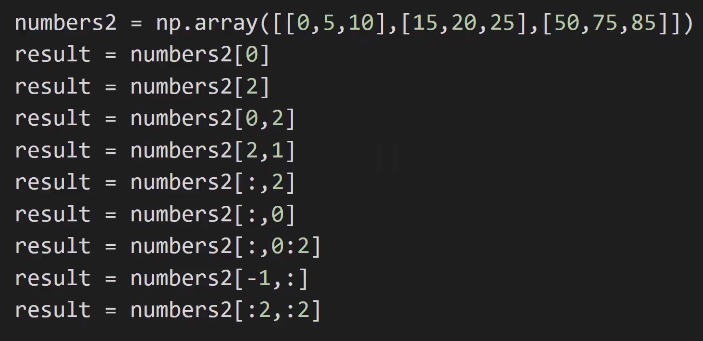
“np.arange(50)” komutuyla sıfırdan elliye kadar olan bir matrix oluşturabilir, bu oluşturduğumuz matrisi reshape gibi komutlarla manipüle edebiliriz.



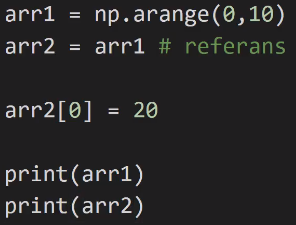
Yukarıdaki örnekte “.sum(axis=0/1)” komutu ile sıra veya sütundaki değerler toplamını hesaplattık.

“.max()/.min()” komutu ile random üretilmiş değerlerin en büyük/küçüğüne ulaşabiliriz (random komutu ile kullanılmak zorunda değil). “.argmax()/argmin” komutu ise max ve min olan sayıların listede kaçıncı indeks olduklarını göstermektedir, bunun yanında “.mean” komutu ile lsitemizin mean değerini hesaplatabiliriz.

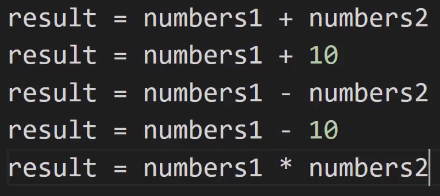
“numbers” isminde 10 farklı değere sahip bir liste tanımladığımızı düşünelim, “numbers[::-1]” komutu ile liste içerisindeki değerlerin indexini tersine çevirebiliriz. “numbers[::-2]” olduğu takdirde liste ters çevirmekle kalınmaz indexler ikişer ikişer alınır (listedeki eleman sayısı öncekinin yarısı olur).



Yukarıdaki örnekte sıra ve sütunlar üzerindeki farklı uygulamalar gösterilmiştir.

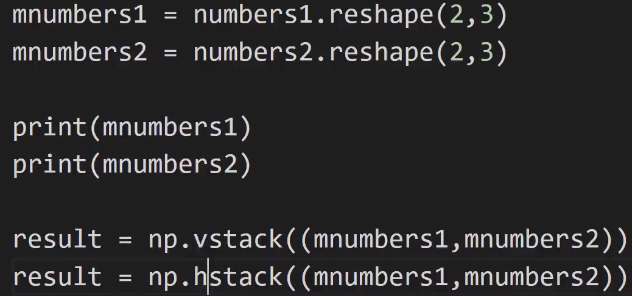


Yukarıdaki örnekte arr2, arr1 ile eşitlenmesinin ardından direkt olarak arr1’in kaynağına bağlanır, bu da arr2 üzerinde yapılacak herhangi bir değişimin arr1 üzerinde de oluşacağı anlamına gelmektedir. Eğer arr2 için “arr1.copy()” eşitlemesini yaparsak arr1’in bir kopyası oluşacak ve adresler farklı alana kaydedileceğinden arr2 üzerinde yapılan değişiklik arr1’e yansımayacaktır.

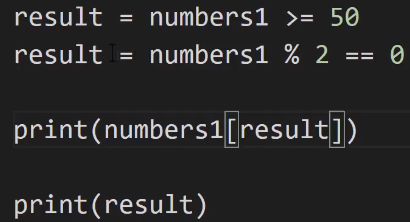


Oluşturulan listeye matematiksel işlemler uygulanabilir (+10 dersek her elemanı 10 ile toplar).

Aynı zamanda bu liste sin, cos, sqrt, log gibi matematiksel işlemleri de desteklemektedir.



Eğer bir matrisi başka bir matrixle birleştirmek istersek bunu yanal ekleme veya dikey ekleme olarak yapabiliriz, “np.vstack((m1, m2))” komutu vertical ekleme yaparken “np.hstack((m1, m2))” horizontal birleştirme sağlar.



Belirlediğimiz değişkeni listemizde koşul belirttiğimiz takdirde cevabı True/False olarak alabiliriz (örneğin numbers1 elemanları 50’den büyük mü [False True True False True]). “numbers1[result]” ise matrisin True değerlerini yazdırır ve boolean olarak değerleri belirtir.