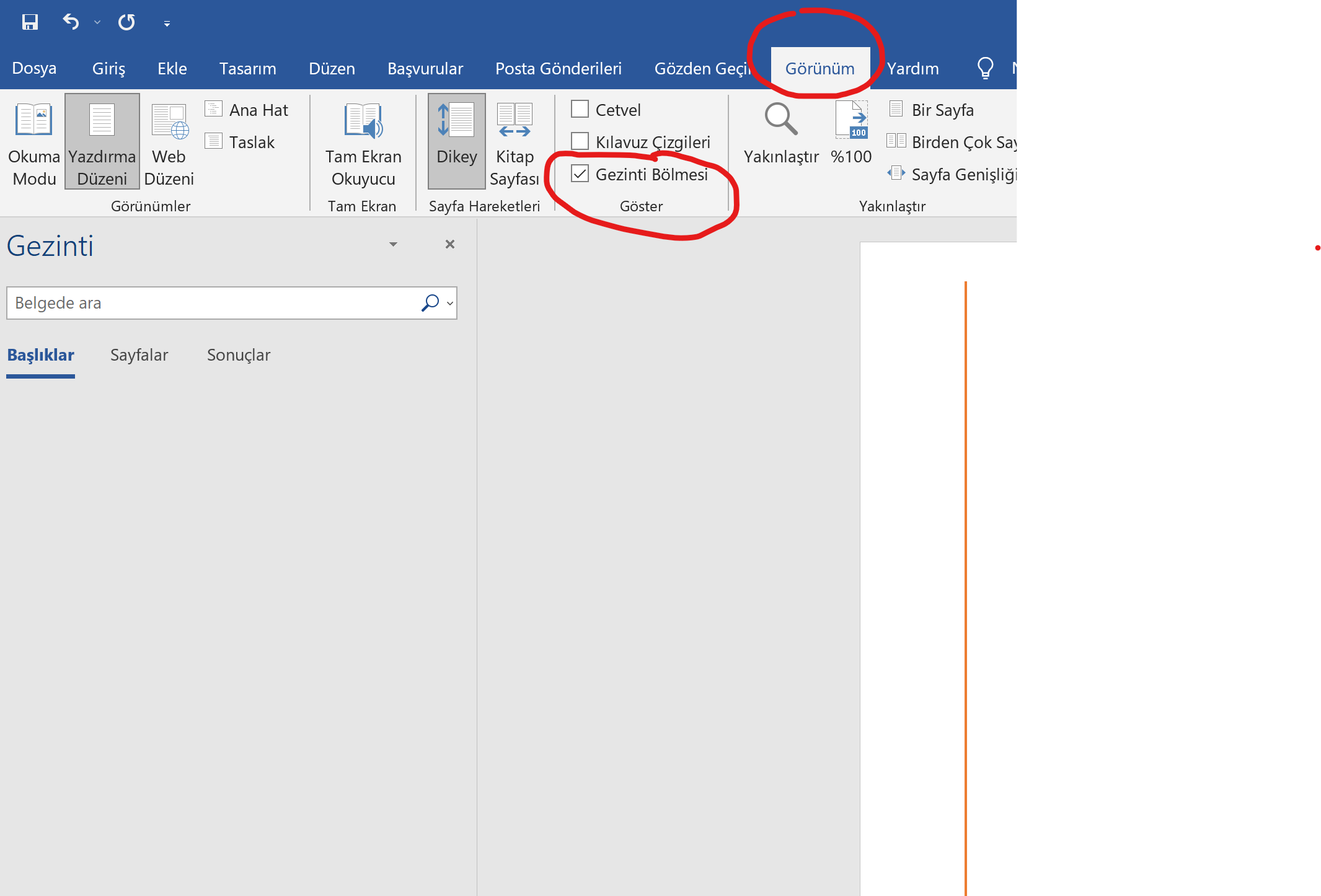
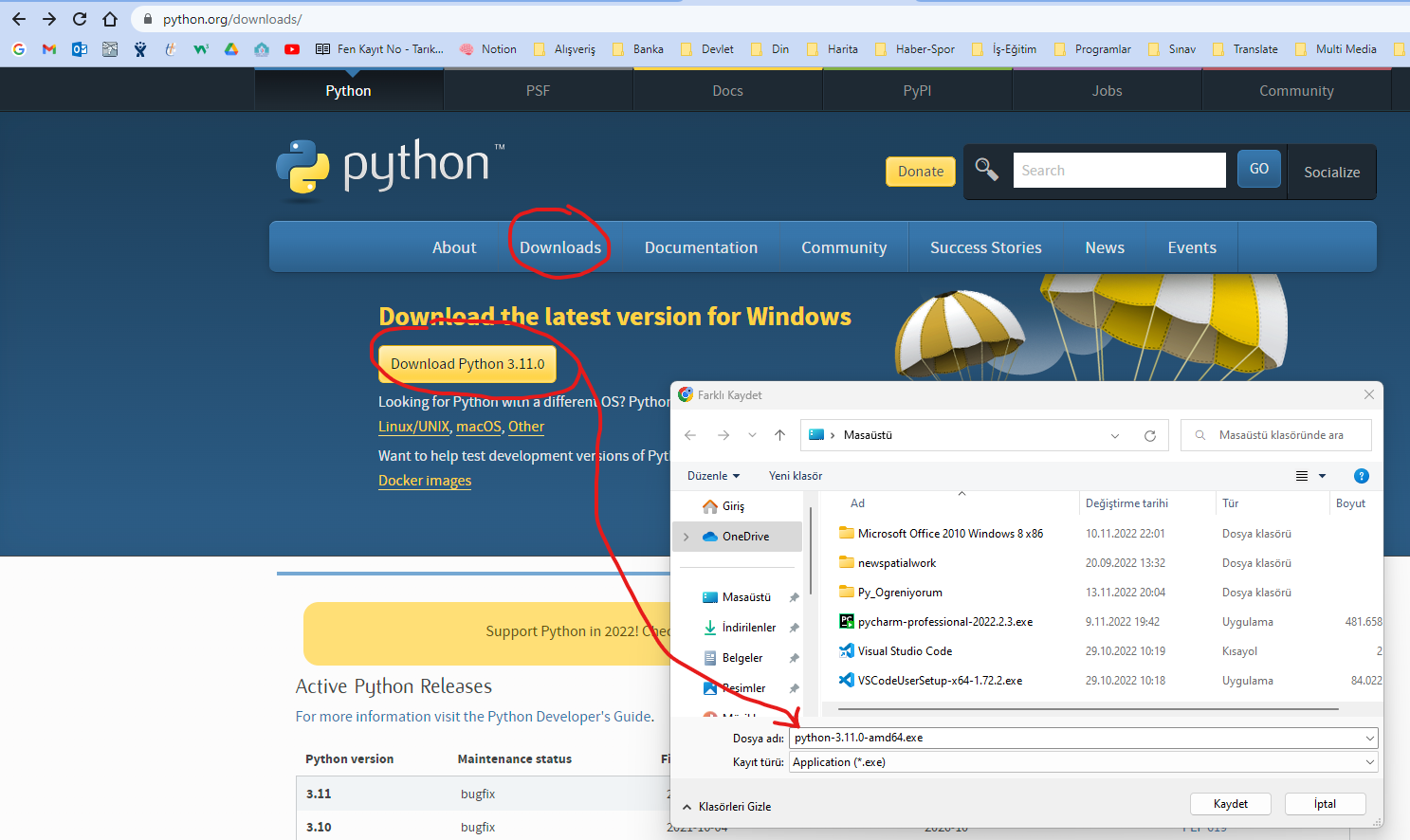
Bu çalışma dosyasında gezinti bölümünü açarsanız daha rahat bilgilere erişebilirsiniz.



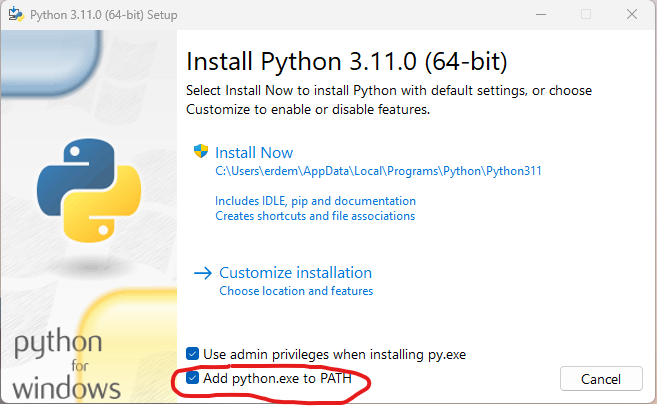
# Kurulumu

## Python Kurulumu

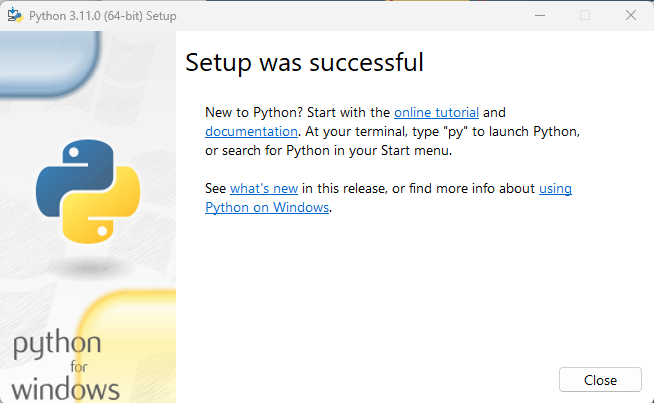
<https://www.python.org/downloads/> adresinden python’ın en son sürümünü indirin. Aşağıdaki görseldeki gibi.



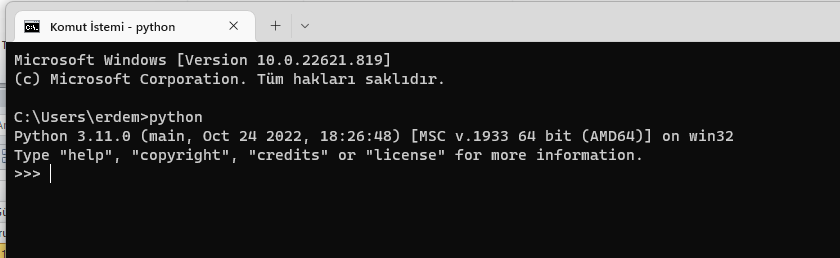
İndirdiğiniz exe’yi çalıştırın. Alttaki ekrandaki gibi “Add pytgon.exe to PATH” yazan yeri işaretleyin ve Install Now ile kurulumu başlatın.



Kurulum bittikten sonra Setup was seccessful geldiğini göreceksiniz. Close butonu ile kurulum ekranından başarılı bir şekilde çıkın.



Windows’unuzun başlat çubuğundan yada kalveyenizdeki  + R tuşuna basarak çalıştır ekranı gelecektir. Bu ekrana CMD yazıp enter tuşuna basın. Yeni gelen siyah ekrana Python kelimesini yazarak python dilinin hangi versiyonun yüklendiğini göreceksiniz. Not: Eğer, aşağıdaki gibi siyah ekranda python 3.11.0 gibi bir yazı gelmiyorsa düzgün yüklenmemiştir. Bu durumda Program ekle kaldır’dan Python ‘ı silip tekrardan en baştaki [Python Kurulumu](#_Python_Kurulumu) ‘unu yapınız.



## Pychar kurumulu

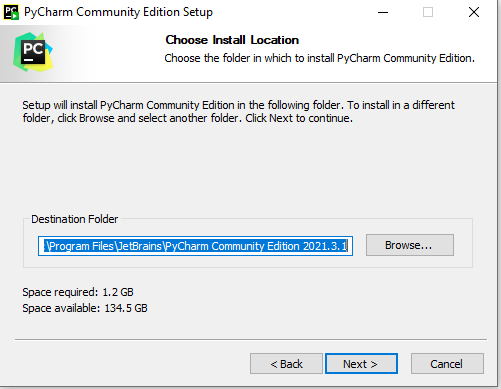
[https://www.jetbrains.com/pycharm/download/#section=windows](https://www.jetbrains.com/pycharm/download/%23section=windows) sitesinden Pycharm IDE’sini indirin.



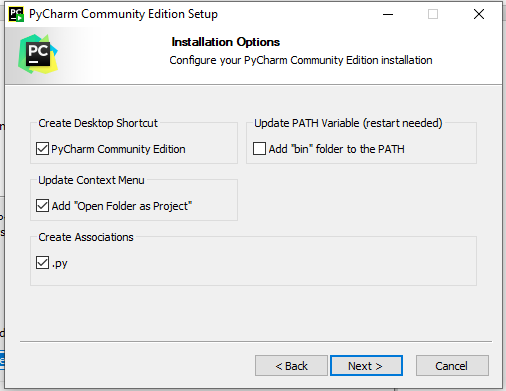
İndirmiş olduğunuz Pycharm IDE’sinin exe’sine tıklayarak kurulumu başlatın. Gelen ekrandan NEXT tuşu ile bir sonraki adıma geçin.

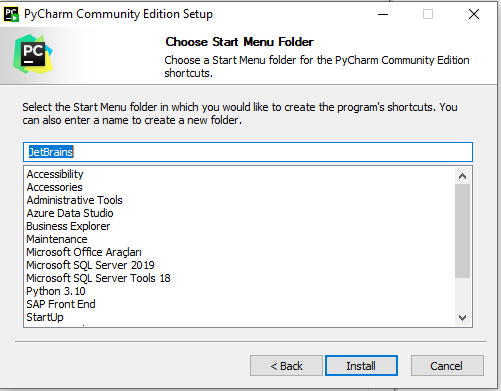


Aşağıdaki gibi tekrar NEXT tuşuna basarak devam edin.

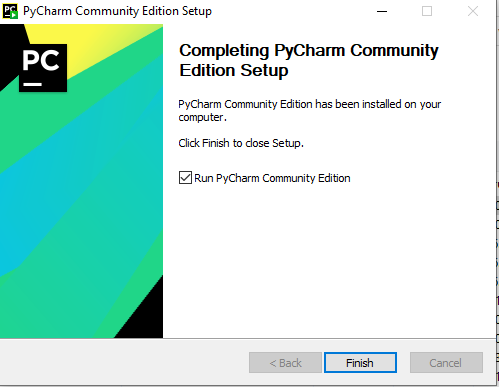


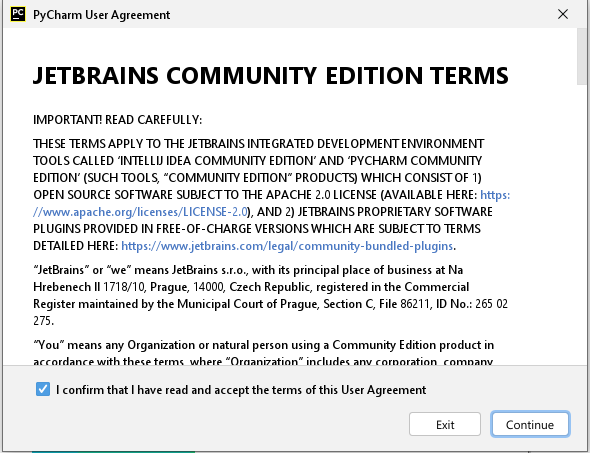
Bu ekranda aşağıdaki işaretlinmiş alanlar gibi işaretleyip NETX tuşuna basın ve diğer gelecek ekranlarda da NEXT tuşuna basın.





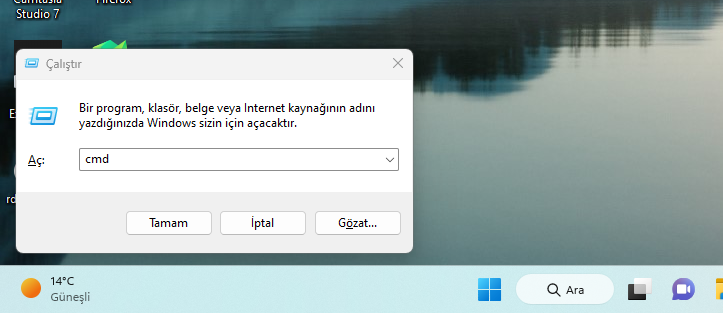
Kurulum tamamlnadıktan sonra FINIS tuşuna basarak bitirin.



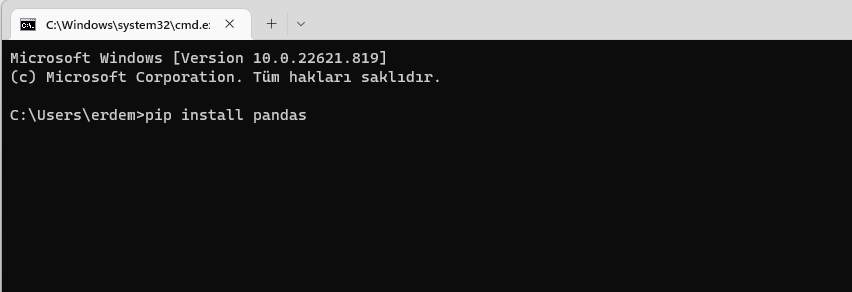


## PIP – Python Kütüphaneleri Kurulumu

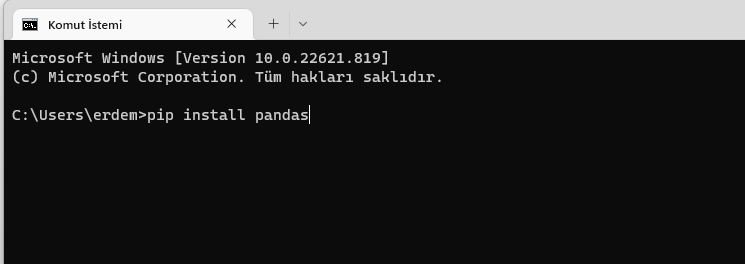
Windows’unuzun başlat çubuğuna çalıştır yazın yada kalveyenizdeki  + R tuşuna nastığınızda ÇALIŞTIR ekranı gelecektir. Bu ekrana CMD yazıp enter tuşuna basın.



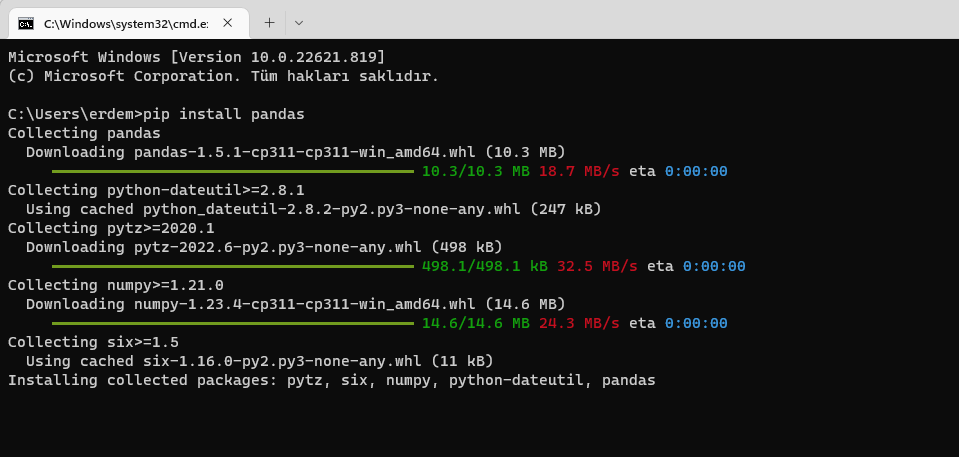
Enter tuşuna bastıktan sonra siyah ekran gelecektir. Bu ekrana aşağıdaki gibi “pip install pandas”



Şimdi gelin Python’ın PANDAS kütüphanesini kuralım. Bunun için yukarıdaki ekrandaki >>> işaretinden sonra “pip install pandas” kelimesini yazalım. Yani aşağıdaki gibi.

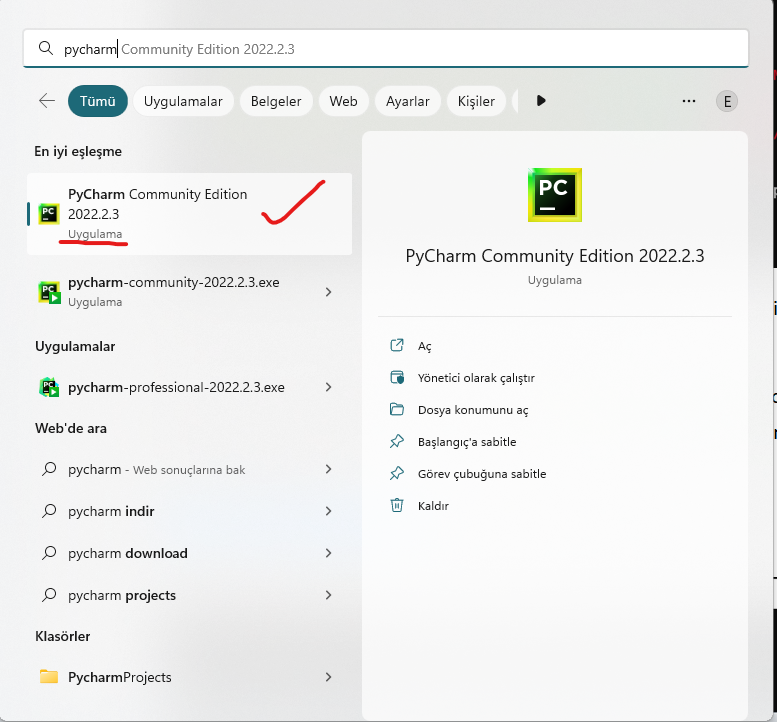


Bu kelimeyi yazdıktan sonra enter tuşuna bastığımızda aşağıdaki gibi PANDAS kütüphanemizi yüklemeye başlayacaktır.

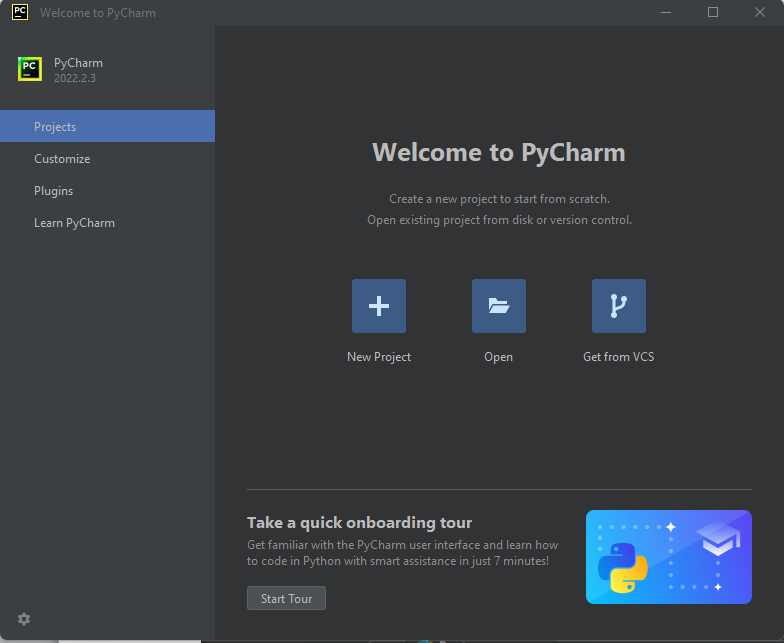


Python kütüphanelerinin yüklemesinin başka bir şekli daha var. Gelin şimdi bir de python kütüphanelerini Pycharm IDE’sini içinden nasıl kurulduğunda bakalım.

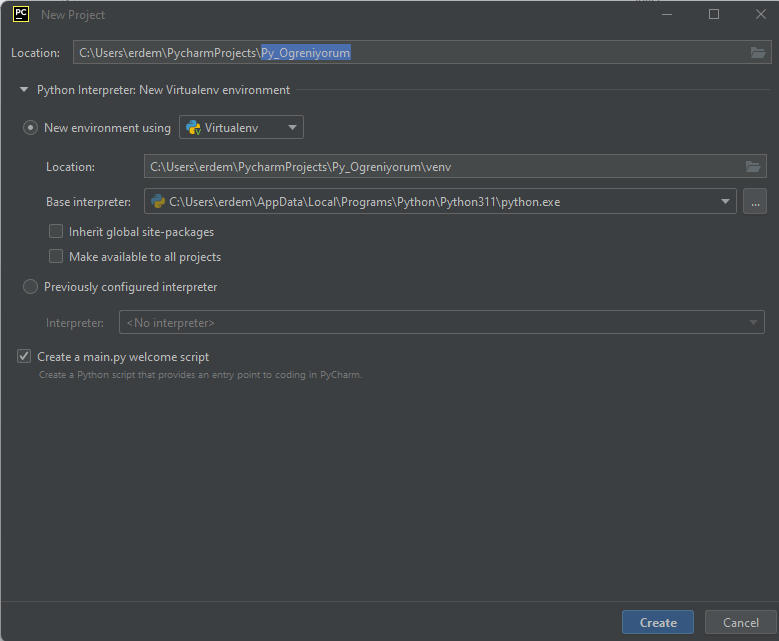
Daha önce kurmuş olduğumuz Pycharm IDE’mizi masaüstündeki iconundan çalıştıralım. Eğer masaüstünüzde yoksa o zaman Windows başlat çubuğundan Pycharm IDE’mizin adını yazıp çalıştıralım. Yani aşağıdaki gibi.



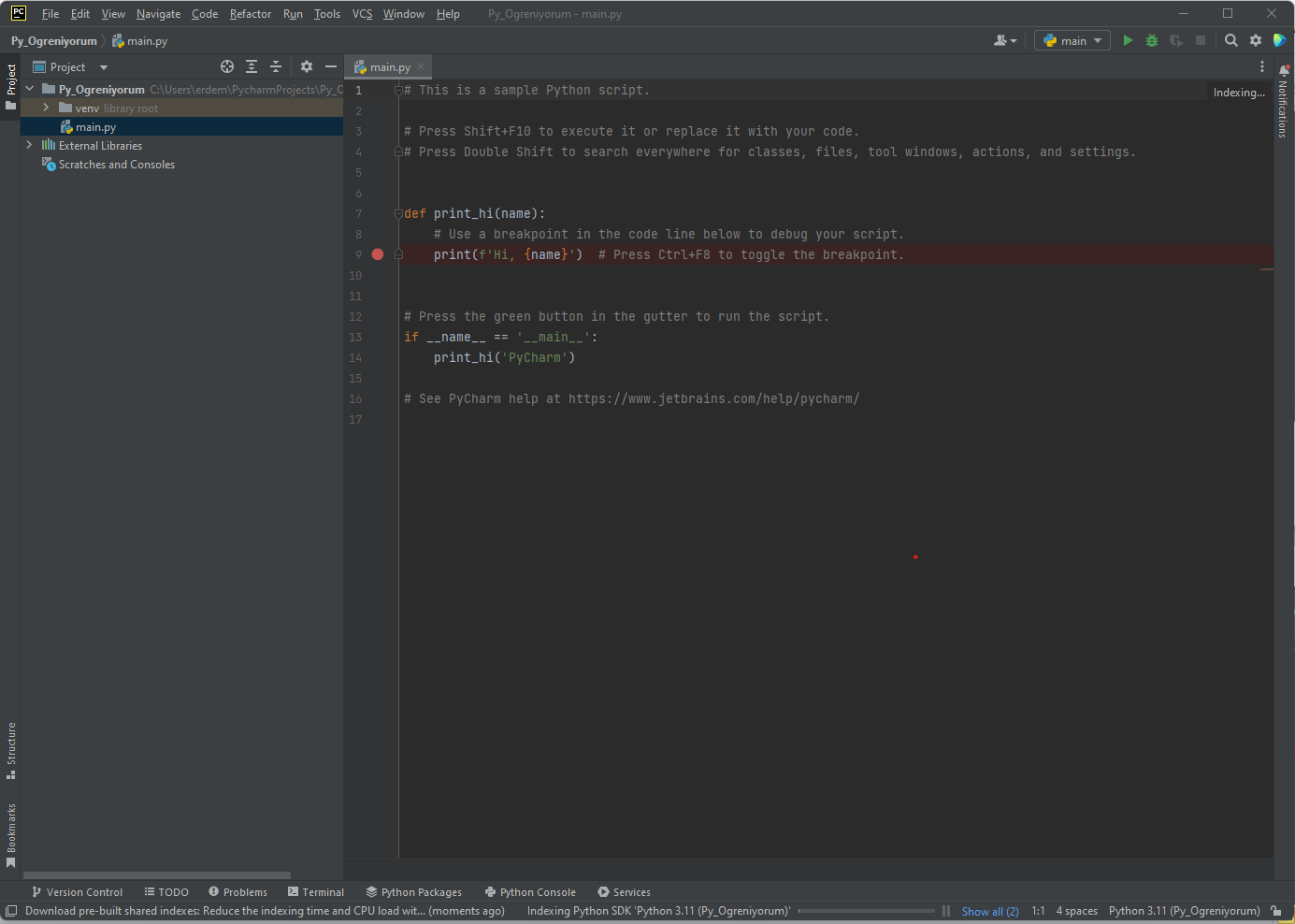
Pycharm IDE’miz çalıştırdığımıza göre aşağıdaki gibi bir ekran gelecektir. Bu ekranda New Project olana tıklıyoruz.



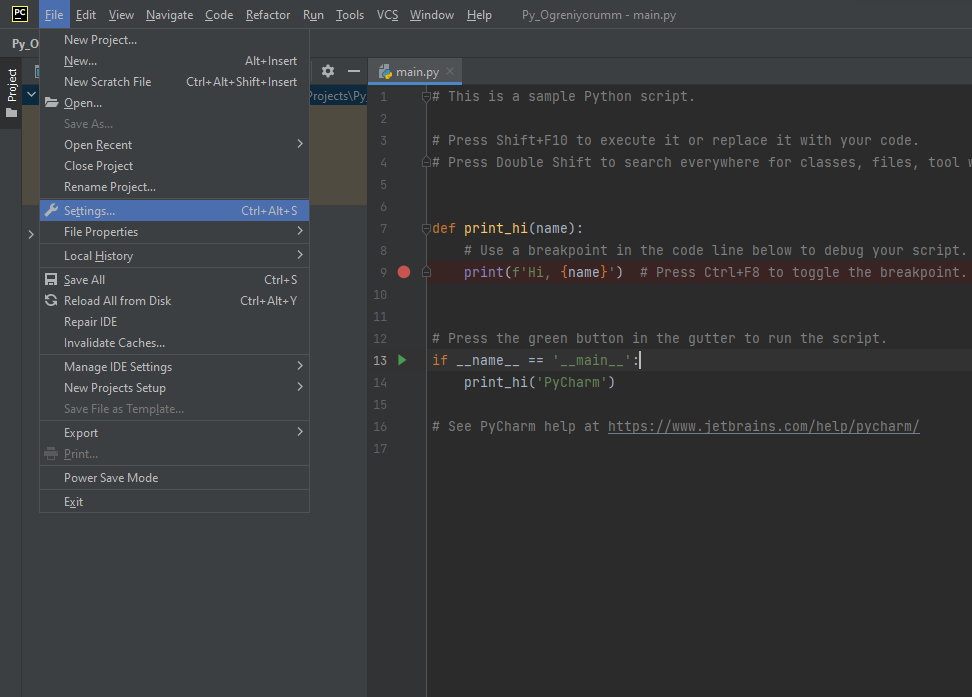
Bu ekranda ise çalışmak istediğimiz projemizin adını belirliyoruz. Örneğin ben adını “***Py\_Ogreniyorum***” olarak belirledim. Siz de istediğiniz bir ad verebilirsiniz. Yalnız dikkat edilmesi gereken Türkçe karakter kullanmamızdır. Birden fazla bir kelime yazacaksak kelimeler arasında boşluk ve nokta gibi karkterler kullanılmaz. Proje adımızıda belirlediğimize göre artık **CREATE** tuşuna basarak projemizi oluşturuyoruz.



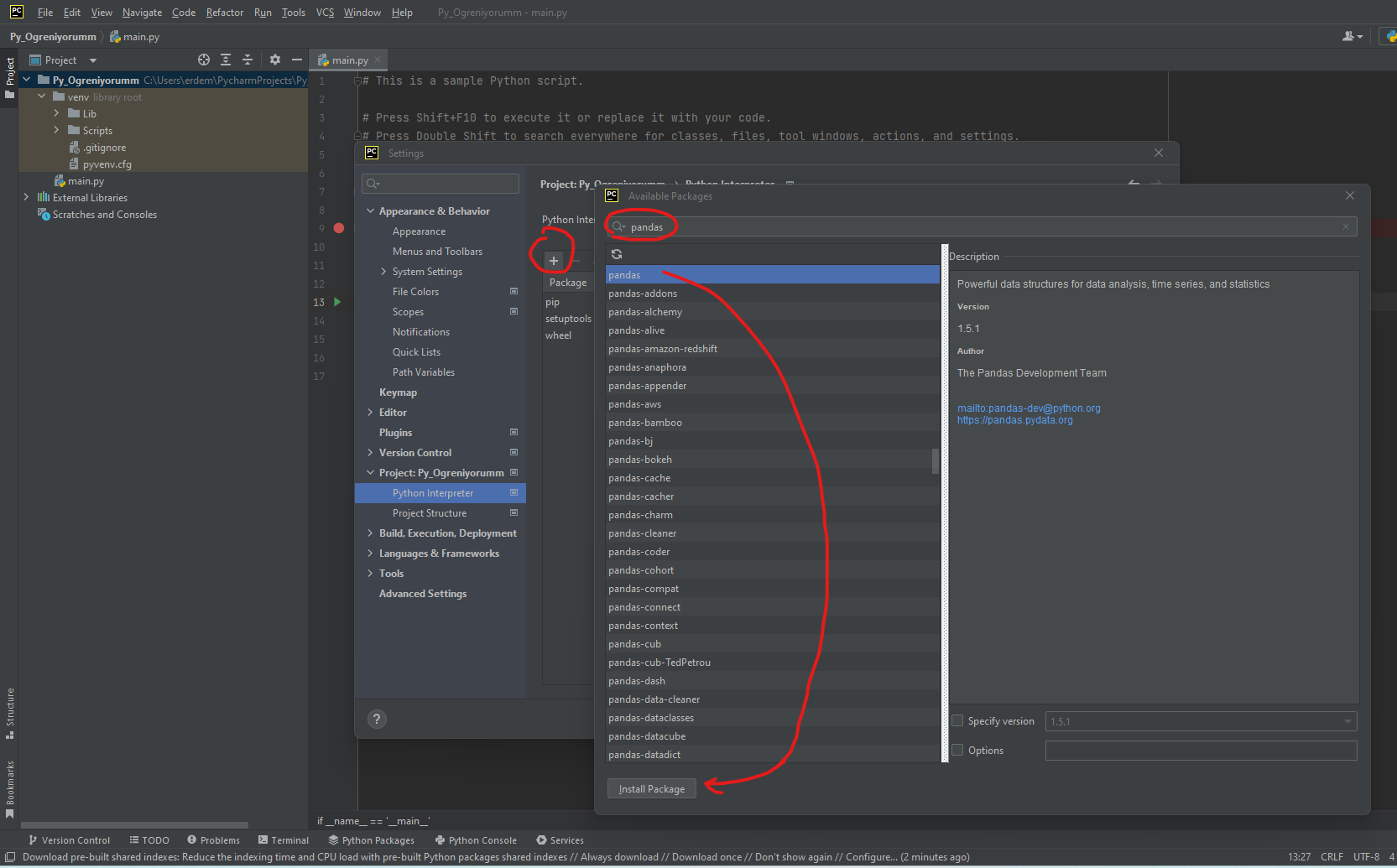
Artık Pycharm IDE’mizi açabildik.



Şimdi gelin bir python kütüphanesi Pycharm IDE’sinden nasıl yüklenire bakalım. Aşığıdaki gibi File menüsünden “Settings” sekmesine tıklayalım. Kısa yolu CTRL + ALT + S tuşuna basarsanızda settings kısmına girebilirsiniz.



Settings ekranı açıldıktan sonra aşağıdaki gibi Python: Py\_Ogreniyorumm sekmesini altındaki “Python Interpreter“ a tıklayalım. Bu ekrandaki + butonuna bastığımızda tüm kütüphaneleri inderebileceğimiz üçüncü bir ekran açılacak. Search (arama) kısmına PANDAS yazarak altta çıkan pandas’a tıklayıp en alttaki Install Package butonu ile kütüphanemizi kendi projemize eklemiş oluyoruz.

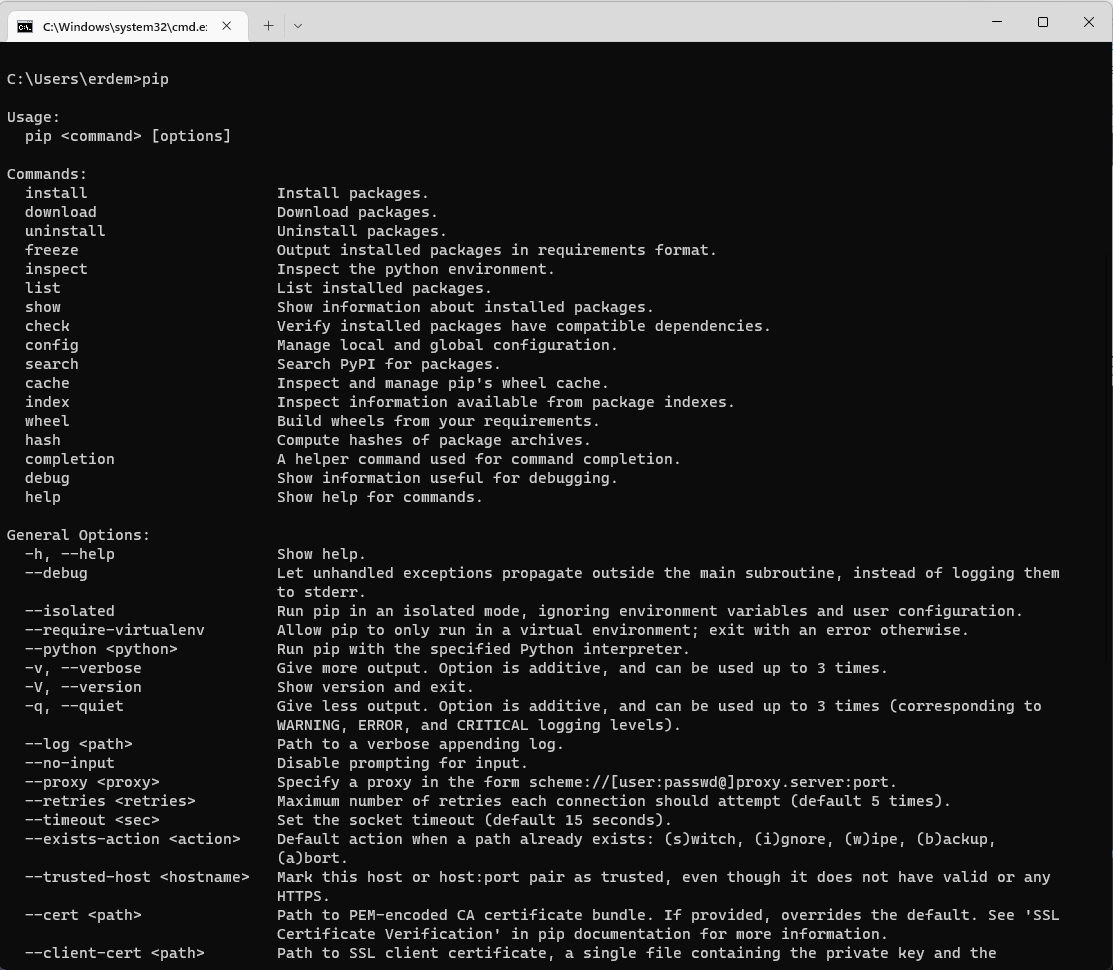


#### Kısa Bilgiler – Python – Pycharm

##### Bilgi 1: Tüm Paython Kütüphaneleri İçin Aşağıdaki Linkten Pip’ların Neler Olduğunu Öğrenebilirsiniz.

<https://pypi.org/project/pip/> nuget ‘lerin indirildiği site

##### Bilgi 2: Cmd’ye Pip Yazarsanız, Pip Komuatlarının Neler Olduğunu Görebilirsiniz.



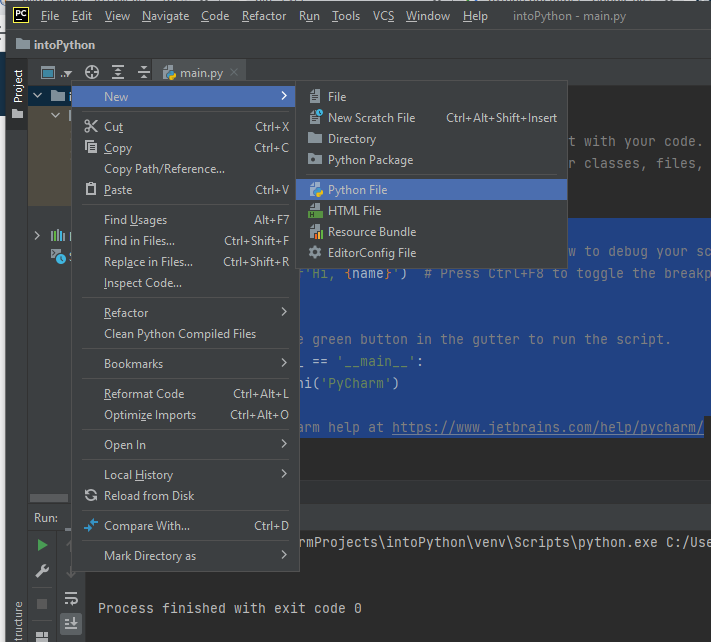
##### Bilgi 3: Sınıf çağırma

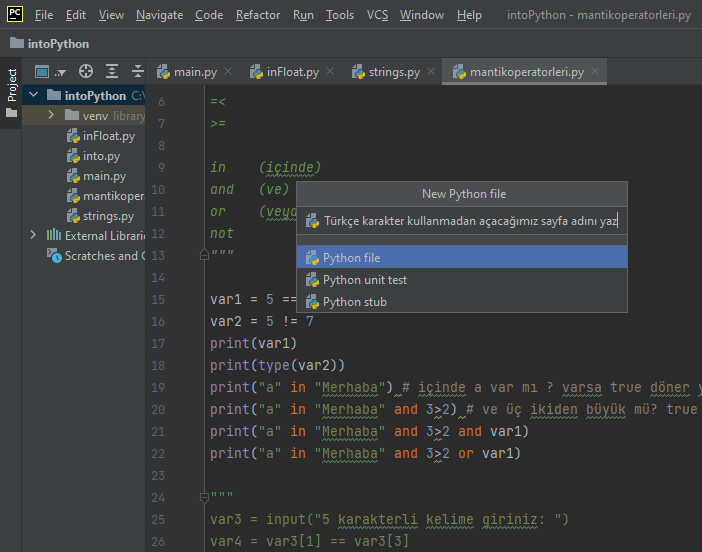
def print\_hi(name):  
 # Use a breakpoint in the code line below to debug your script.  
 print(f'Hi, {name}') # Press Ctrl+F8 to toggle the breakpoint.  
  
  
# Press the green button in the gutter to run the script.  
if \_\_name\_\_ == '\_\_main\_\_':  
 print\_hi('PyCharm')  
  
# See PyCharm help at https://www.jetbrains.com/help/pycharm/

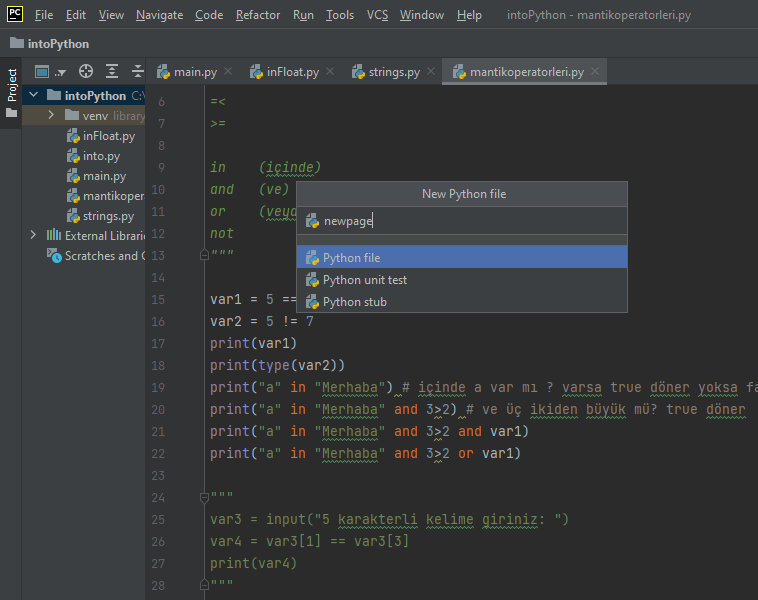
Süslü parantez içindeki {name}bir sınıftır. Name ile bir sonraki koda atama yapılabilmektedir.

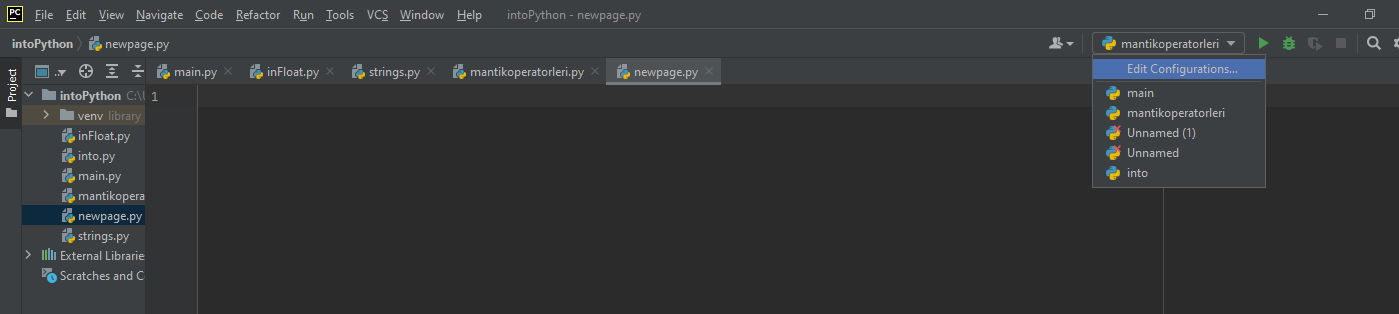
Bir sınıfı çağırmak için -- -- işaretlerinin arasına yazılarak sınıf çağırılır.

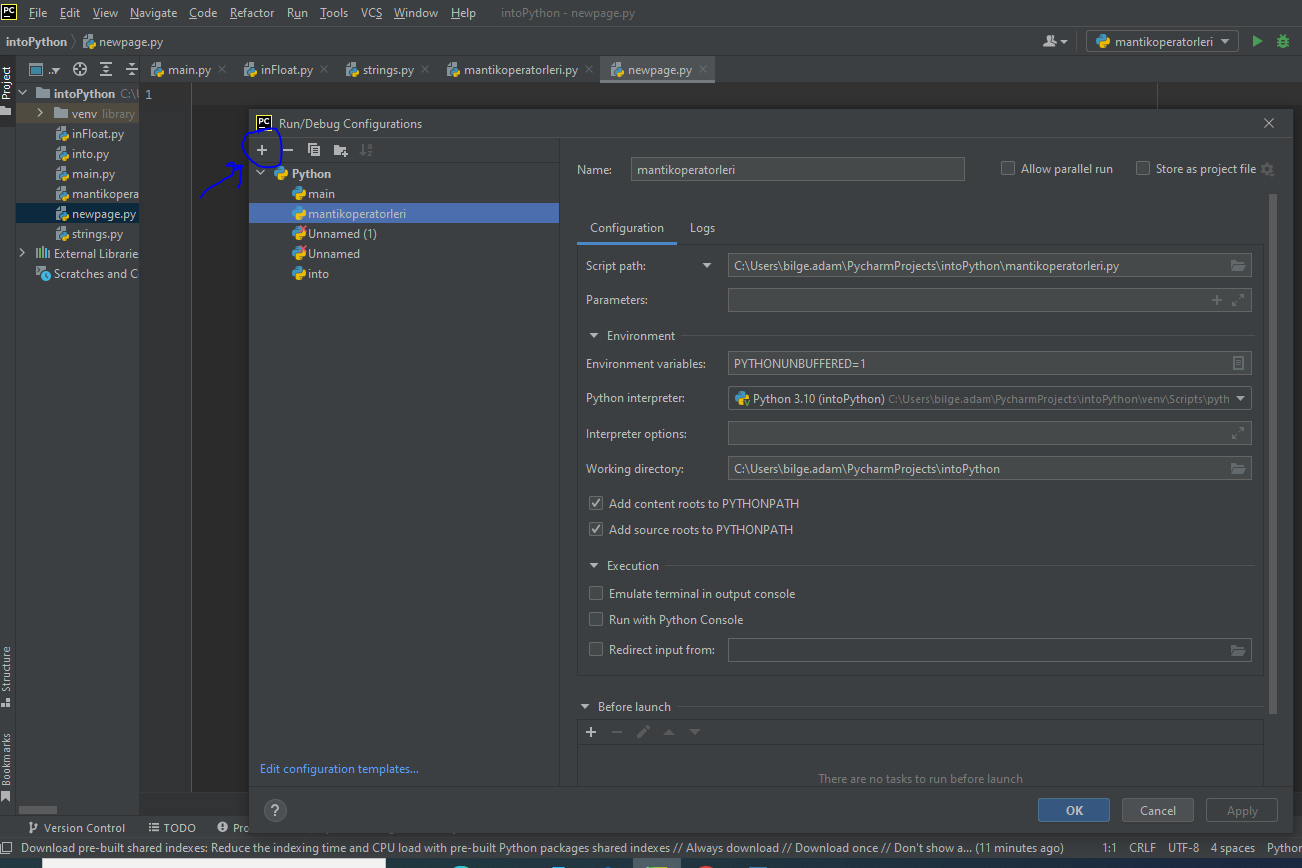
##### Bilgi 4: Pycharm Yeni bir dosya

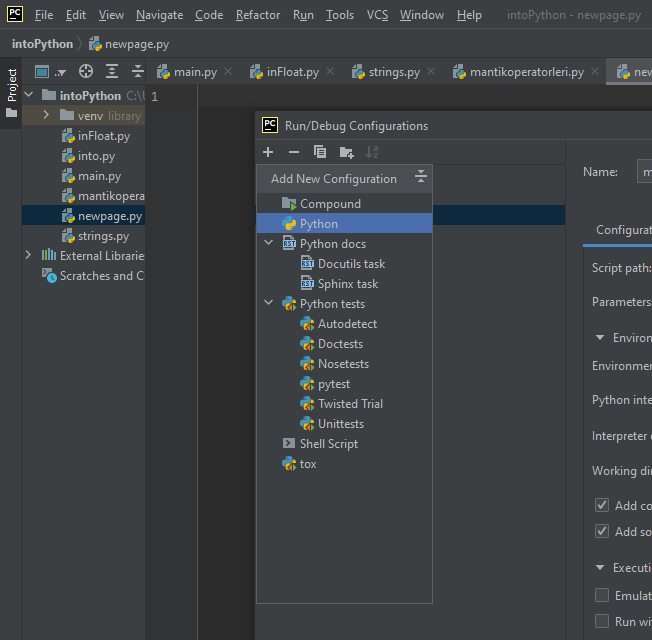


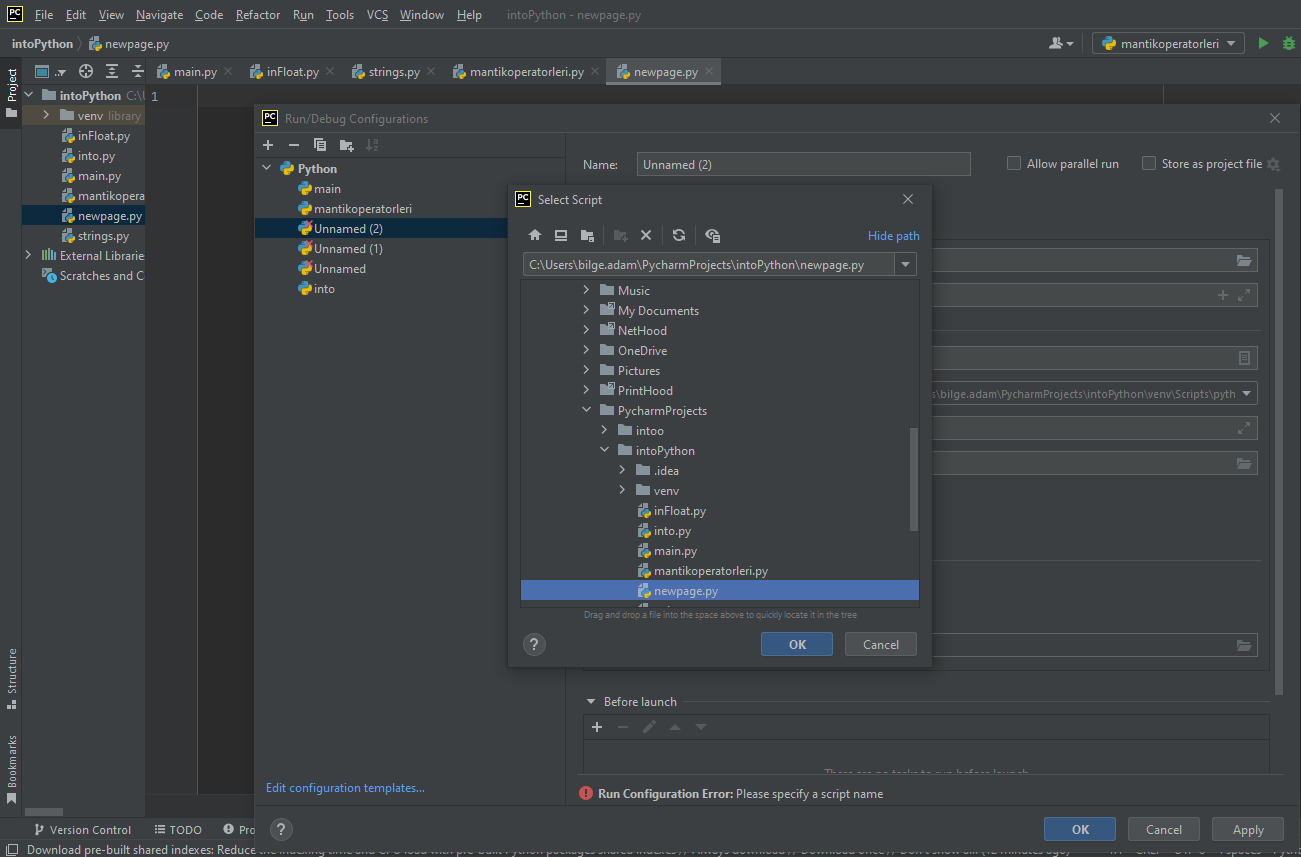






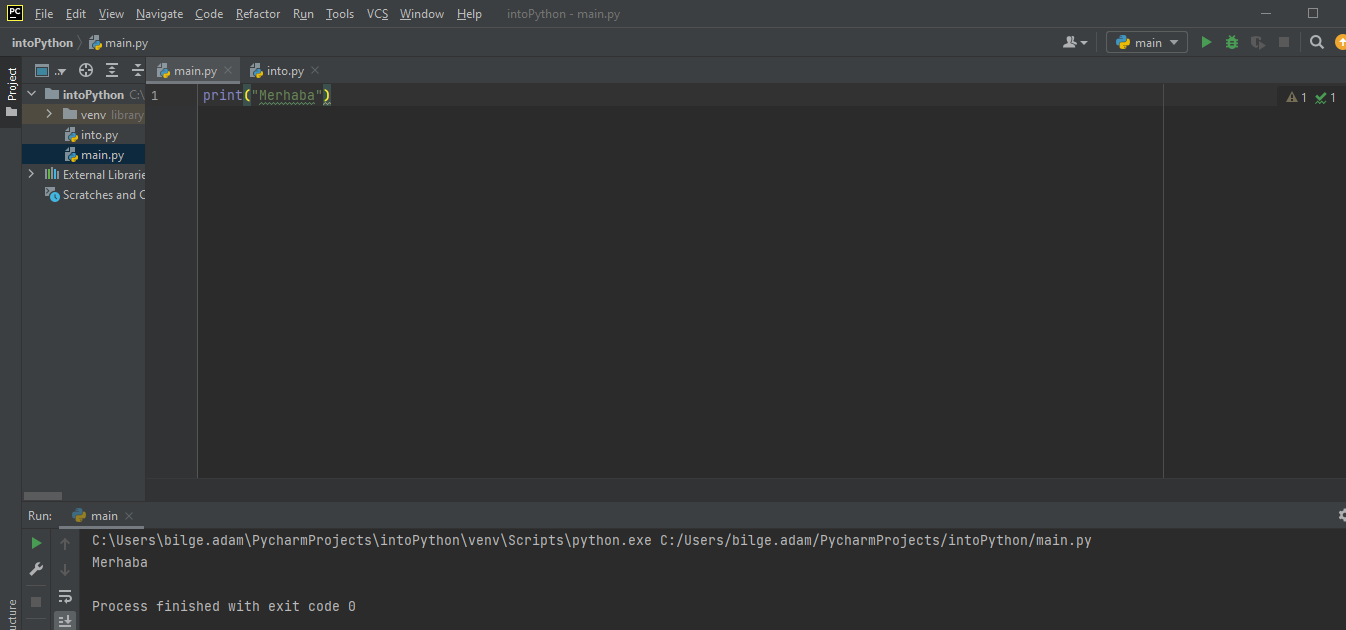




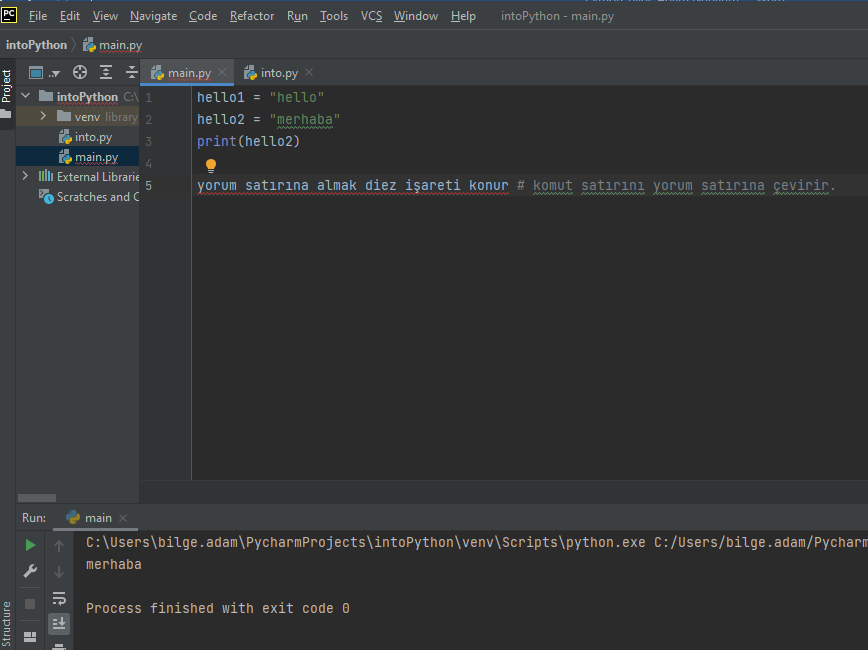


###### Yeni dosya açtıktan sonra CTRL+SHIFT+F10 tuşuna basarak çalıştırın.

##### Bilgi 5: Print Komutu



##### Bilgi 6: Diez # (yorum satırı – command line)



Birden fazla satırı hızlıca command line (yorum satırına) almak istiyorsak CTRL + /

İşaretine aynı anda basarak alınabilir.

Yada “”” “”” (üç adet çift tırnak ) işareti ile de alınabilir.

## Önemli Linkler:

1. https://www.hackerrank.com/domains/python



6. Emre Yazıcı GitHub Sayfası : <https://github.com/eyazicimina/tk>
7. PostGIS Doküman: <https://postgis.net/workshops/postgis-intro/>
8. PostGIS Doküman: <https://postgis.net/docs/manual-3.3/>
9. PostGIS St\_List : <https://postgis.net/docs/reference.html#PostGIS_Types>

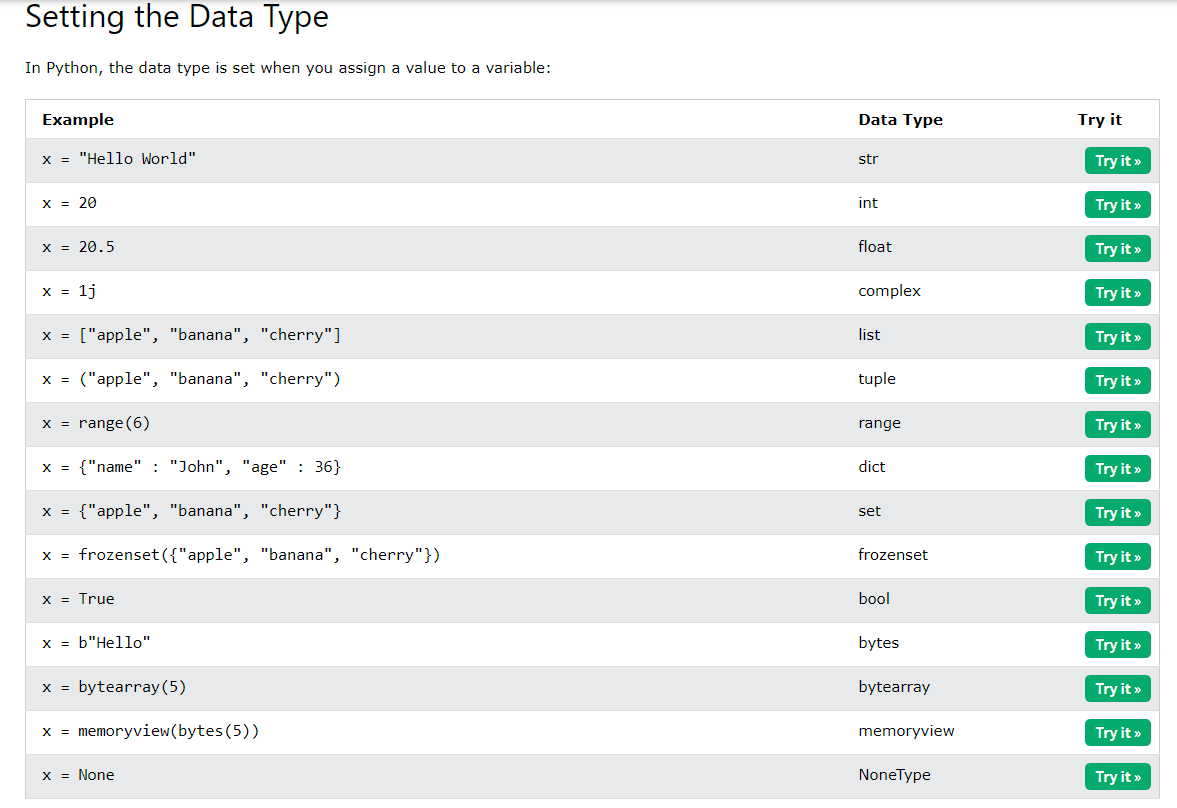
12. w3School sitesinden Python öğrenme: <https://www.w3schools.com/python/>
13. Python KeyWord’leri: <https://realpython.com/python-keywords/>
14. : <https://deparkes.co.uk/>
15. Farklı altlık haritalar: <https://www.datasciencearth.com/folium-ile-harita-gorsellestirme/>
16. Polygon çizme/görüntüleme programı : <https://arthur-e.github.io/Wicket/sandbox-gmaps3.html>

# PYTHON ÖĞRENİYORUM

# Data Types – Data Tipleri

|  |  |
| --- | --- |
| Text Type: | str |
| Numeric Types: | int, float, complex |
| Sequence Types: | list, tuple, range |
| Mapping Type: | dict |
| Set Types: | set, frozenset |
| Boolean Type: | bool |
| Binary Types: | bytes, bytearray, memoryview |
| None Type: | NoneType |

## **Setting the Data Type**

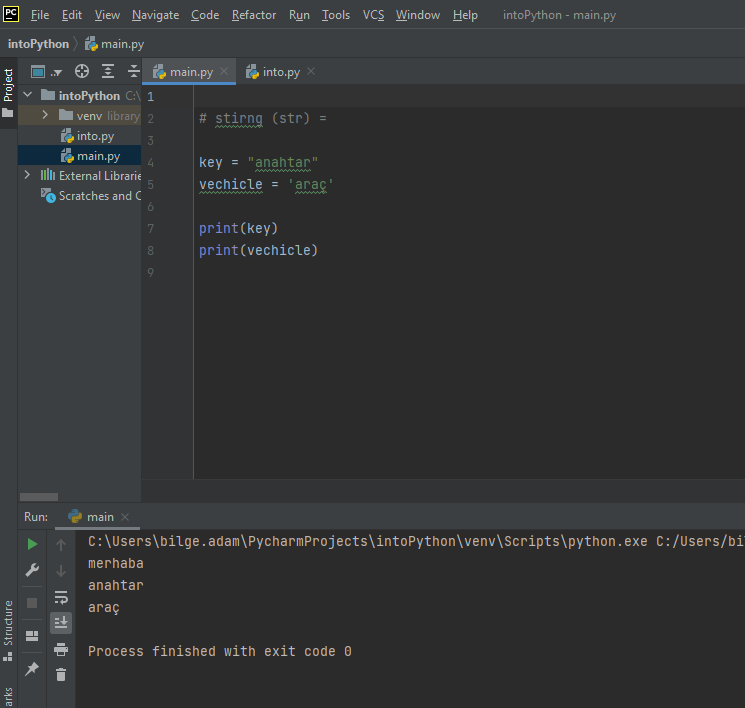


## **Setting the Specific Data Type**



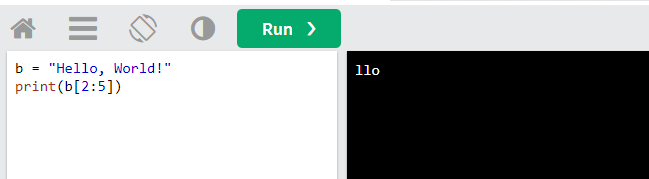
<https://www.w3schools.com/python/python_datatypes.asp>

## String (str) - Metin

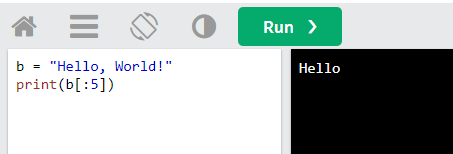


### Slicing Strings – Dilimleme dizeleri

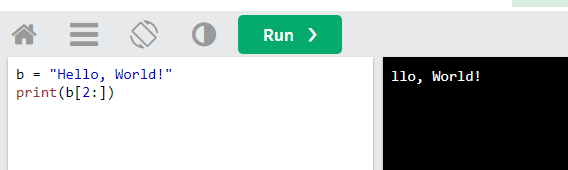
#### [2:5]: <https://www.w3schools.com/python/trypython.asp?filename=demo_string2>



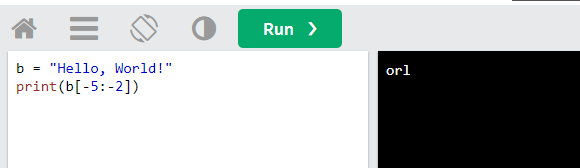
#### [:5]: <https://www.w3schools.com/python/trypython.asp?filename=demo_string_slice_start>



#### [2:]: <https://www.w3schools.com/python/trypython.asp?filename=demo_string_slice_end>



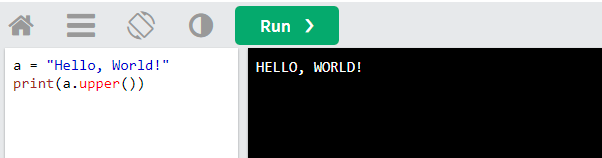
#### [-5:-2]: <https://www.w3schools.com/python/trypython.asp?filename=demo_string_negativeindex>



### Modify Strings – Dizeleri Değiştirme

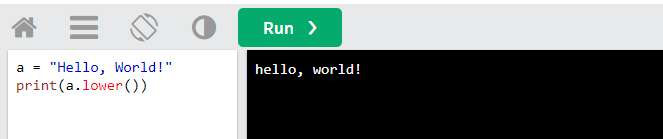
#### Upper Case – Büyük Harf Yapma

<https://www.w3schools.com/python/trypython.asp?filename=demo_string_upper>



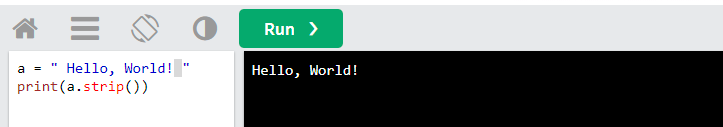
#### Lower Case – Küçük Harf Yapma

<https://www.w3schools.com/python/trypython.asp?filename=demo_string_lower>



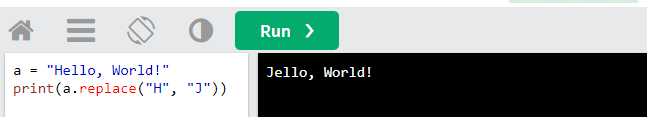
#### Remove Whitespace – Boşluğu Kaldırma

<https://www.w3schools.com/python/trypython.asp?filename=demo_string_strip>



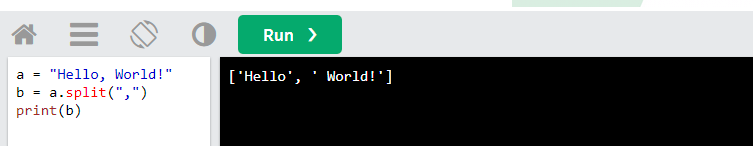
#### Replace String – Dizeyi Değiştirme

<https://www.w3schools.com/python/trypython.asp?filename=demo_string_replace>



#### Split String – Dizelere Bölme

<https://www.w3schools.com/python/trypython.asp?filename=demo_string_split>



#### \*\*\* Python String Methods – String Method Listesi \*\*\*

Yöntem Açıklama

|  |  |
| --- | --- |
| capitalize() | İlk karakteri büyük harfe dönüştürür |
| casefold() | Dizeyi küçük harfe dönüştürür |
| center() | Ortalanmış bir dize döndürür |
| count() | Belirtilen bir değerin bir dizgede kaç kez oluştuğunu döndürür |
| encode() | Dizenin kodlanmış bir sürümünü döndürür |
| endswith() | Dize belirtilen değerle bitiyorsa true değerini döndürür |
| expandtabs() | Dizenin sekme boyutunu ayarlar |
| find() | Belirtilen bir değer için dizeyi arar ve bulunduğu konumu döndürür |
| format() | Belirtilen değerleri bir dizgede biçimlendirir |
| format\_map() | Belirtilen değerleri bir dizgede biçimlendirir |
| index() | Belirtilen bir değer için dizeyi arar ve bulunduğu yerin konumunu döndürür |
| isalnum() | Dizedeki tüm karakterler alfasayısal ise True döndürür |
| isalpha() | Dizedeki tüm karakterler alfabedeyse True döndürür |
| isascii() | Dizedeki tüm karakterler ascii karakter ise True döndürür |
| isdecimal() | Dizedeki tüm karakterler ondalık ise True döndürür |
| isdigit() | Dizedeki tüm karakterler rakam ise True döndürür |
| isidentifier() | Dize bir tanımlayıcı ise True döndürür |
| islower() | Dizedeki tüm karakterler küçük harf ise True döndürür |
| isnumeric() | Dizedeki tüm karakterler sayısal ise True döndürür |
| isprintable() | Dizedeki tüm karakterler yazdırılabilirse True döndürür |
| isspace() | Dizedeki tüm karakterler boşluksa True döndürür |
| istitle() | Dize bir başlığın kurallarına uyuyorsa True döndürür |
| isupper() | Dizedeki tüm karakterler büyük harf ise True döndürür |
| join() | Yinelenebilir bir öğenin öğelerini bir dizeye dönüştürür |
| ljust() | Dizenin sola dayalı bir sürümünü döndürür |
| lower() | Bir dizeyi küçük harfe dönüştürür |
| lstrip() | Dizenin sol kırpılmış sürümünü döndürür |
| maketrans() | Çevirilerde kullanılacak bir çeviri tablosu döndürür |
| partition() | Dizenin üç parçaya ayrıldığı bir tanımlama grubu döndürür |
| replace() | Belirtilen bir değerin belirtilen bir değerle değiştirildiği bir dize döndürür |
| rfind() | Belirtilen bir değer için dizeyi arar ve bulunduğu yerin son konumunu döndürür |
| rindex() | Belirtilen bir değer için dizeyi arar ve bulunduğu yerin son konumunu döndürür |
| rjust() | Dizenin sağa yaslanmış bir sürümünü döndürür |
| rpartition() | Dizenin üç parçaya ayrıldığı bir tanımlama grubu döndürür |
| rsplit() | Dizeyi belirtilen ayırıcıda böler ve bir liste döndürür |
| rstrip() | Dizenin sağ kırpılmış sürümünü döndürür |
| split() | Dizeyi belirtilen ayırıcıda böler ve bir liste döndürür |
| splitlines() | Satır sonlarında dizeyi böler ve bir liste döndürür |
| startswith() | Dize belirtilen değerle başlıyorsa true değerini döndürür |
| strip() | Dizenin kırpılmış bir sürümünü döndürür |
| swapcase() | Harfleri değiştirir, küçük harf büyük harf olur ve tam tersi |
| title() | Her kelimenin ilk karakterini büyük harfe dönüştürür |
| translate() | Çevrilmiş bir dize döndürür |
| upper() | Bir dizeyi büyük harfe dönüştürür |
| zfill() | Dizeyi başlangıçta belirtilen sayıda 0 değerle doldurur |

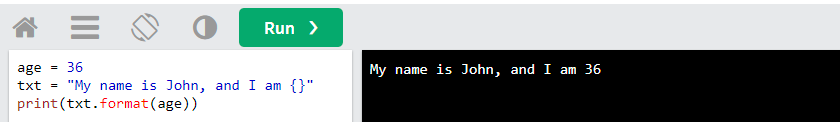
### String Concatenation – Dize(metin) Birleştirme

<https://www.w3schools.com/python/trypython.asp?filename=demo_string_concat2>

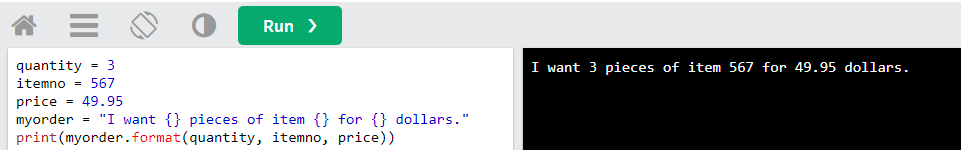
String Format

### String Format – Dize Biçimlendirme

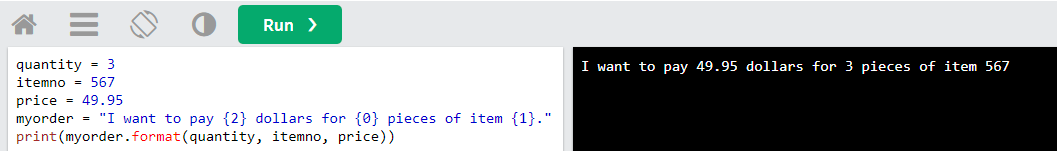
<https://www.w3schools.com/python/trypython.asp?filename=demo_string_format1>



<https://www.w3schools.com/python/trypython.asp?filename=demo_string_format2>

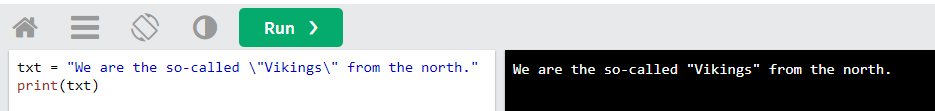


<https://www.w3schools.com/python/trypython.asp?filename=demo_string_format3>



### Escape Character –

<https://www.w3schools.com/python/trypython.asp?filename=demo_string_escape>



#### Escape Characters

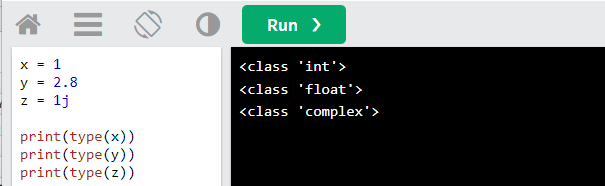
|  |  |  |  |
| --- | --- | --- | --- |
|  | Açıklama | Yapılışı | Sonuç Çıktısı |
| \' | Tek alıntı | txt = 'It\'s alright.' | It's alright. |
| \\ | tek slash işareti koyma | txt = "This will insert one \\ (backslash)." | This will insert one \ (backslash. |
| \n | Yeni hat | txt = "Hello\nWorld!" | Hello Word! |
| \r | Satırbaşı | txt = "Hello\rWorld!" | Hello Word! |
| \t | TAB | txt = "Hello\tWorld!" | Hello Word! |
| \b | bir boşluk siler | txt = "Hello \bWorld!" | HelloWord! |
| \f | Form besleme |  |  |
| \ooo | sekizlik değerde yazma | txt = "\110\145\154\154\157" | Hello |
| \xhh | onaltılık değer yazma | txt = "\x48\x65\x6c\x6c\x6f" | Hello |

## Integer (Int, float, complex)

<https://www.w3schools.com/python/trypython.asp?filename=demo_numbers>

* int
* float
* complex

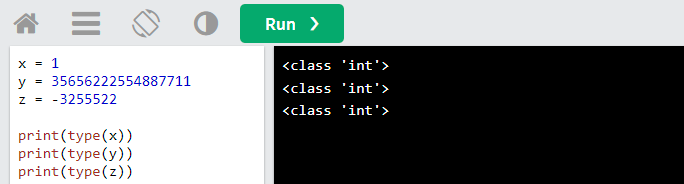
x = 1    # int  
y = 2.8  # float  
z = 1j   # complex



### Int – Intager

<https://www.w3schools.com/python/trypython.asp?filename=demo_numbers_int>

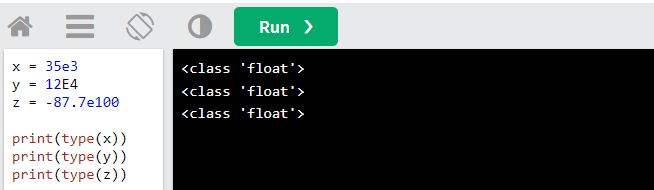
x = 1  
y = 35656222554887711  
z = -3255522  
  
print(type(x))  
print(type(y))  
print(type(z))



### Float

<https://www.w3schools.com/python/trypython.asp?filename=demo_numbers_float2>

x = 35e3  
y = 12E4  
z = -87.7e100  
  
print(type(x))  
print(type(y))  
print(type(z))



### Complex

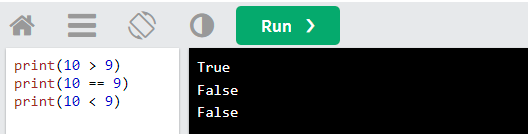
<https://www.w3schools.com/python/trypython.asp?filename=demo_numbers_complex>

x = 3+5j  
y = 5j  
z = -5j  
  
print(type(x))  
print(type(y))  
print(type(z))

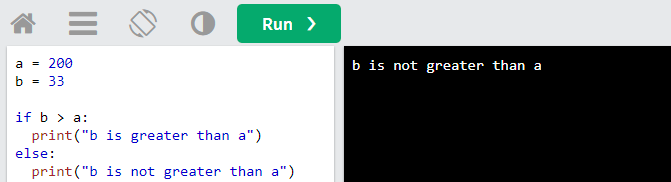


## Booleans

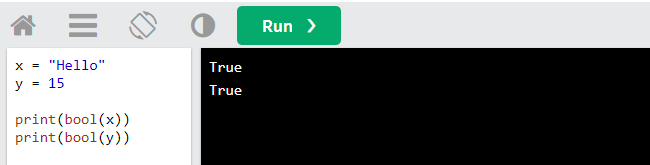
<https://www.w3schools.com/python/trypython.asp?filename=demo_booleans1>



<https://www.w3schools.com/python/trypython.asp?filename=demo_booleans2>



<https://www.w3schools.com/python/trypython.asp?filename=demo_booleans4>



# Operators – Operatörler

## Aritmetik Operatörleri - Arithmetic Operators

+ Addition(Toplama) x + y

- Subtraction(Çıkartma) x - y

\* Multiplication (Çarpma) x \* y

/ Division (Bölme) x / y

% Modulus(Mod alma) x % y

\*\* Exponentiation(Karesini alma) x \*\* y

// Floor division(Kalansız Bölme) x // y

## Karşılaştırma Operatörleri - Comparison Operators

== Equal (eşit ise) x == y

!= Not equal (Eşit Değil ise) x != y

> Greater than (büyük) x > y

< Less than (Küçük) x < y

>= Greater than or equal to (büyük eşit ise) x >= y

<= Less than or equal to (küçük eşit ise) x <= y

## Atama Operatörleri - Assignment Operators

= x = 5 x = 5

+= x += 3 x = x + 3

-= x -= 3 x = x - 3

\*= x \*= 3 x = x \* 3

/= x /= 3 x = x / 3

%= x %= 3 x = x % 3

//= x //= 3 x = x // 3

\*\*= x \*\*= 3 x = x \*\* 3

&= x &= 3 x = x & 3

|= x |= 3 x = x | 3

^= x ^= 3 x = x ^ 3

>>= x >>= 3 x = x >> 3

<<= x <<= 3 x = x << 3

## Mantıksal Operatörleri - Logical Operators

and Her iki ifade de doğruysa True döndürür x < 5 and x < 10

or İfadelerden biri doğruysa True döndürür x < 5 or x < 4

not İfadelerden biri doğruysa True döndürür not(x < 5 and x < 10)

## Kimlik Operatörleri - Identity Operators

is Her iki değişken de aynı nesneyse True döndürür x is y

is not Her iki değişken de aynı nesne değilse True döndürür x is not y

## Üyelik Operatörleri - Membership Operators

in Nesnede belirtilen değere sahip bir dizi varsa True döndürür x in y

not in Nesnede belirtilen değere sahip bir dizi yoksa True döndürür x not in y

## Bitsel Operatörler - Bitwise Operators

& AND Her iki bit de 1 ise her biti 1'e ayarlar

| OR İki bitten biri 1 ise her biti 1'e ayarlar

^ XOR İki bitten yalnızca biri 1 ise her biti 1'e ayarlar

~ NOT Tüm bitleri ters çevirir

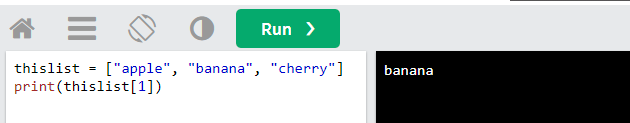
<< Zero fill left shift Sağdan sıfırları iterek sola kaydırın ve en soldaki bitlerin düşmesine izin verin

>> Signed right shift En soldaki bitin kopyalarını soldan içeri iterek sağa kaydırın ve en sağdaki bitlerin düşmesine izin verin

# List

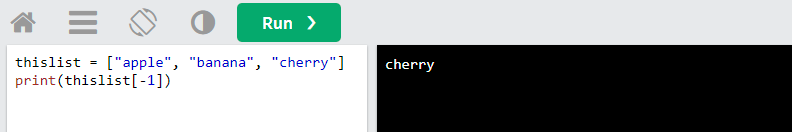
## Access List Items

thislist = ["apple", "banana", "cherry"]  
print(thislist[1])



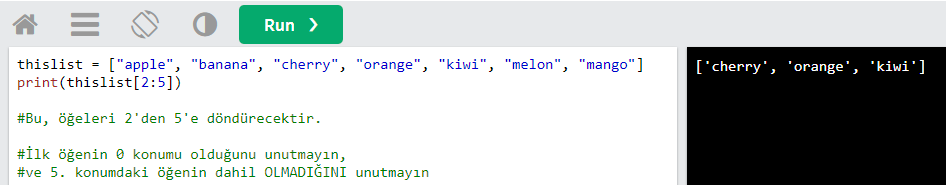
### Negative Indexing

thislist = ["apple", "banana", "cherry"]  
print(thislist[-1])

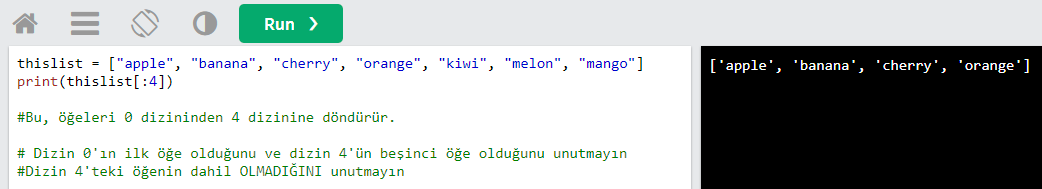


### Range of Indexes

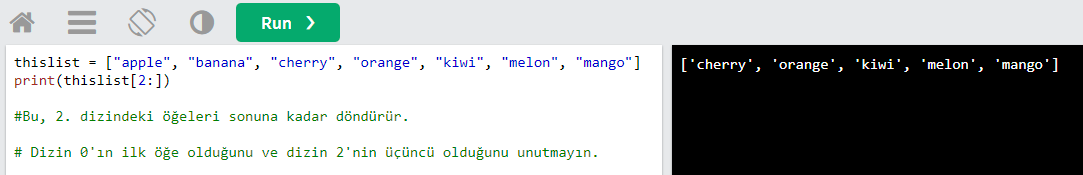
thislist = ["apple", "banana", "cherry", "orange", "kiwi", "melon", "mango"]  
print(thislist[2:5])



thislist = ["apple", "banana", "cherry", "orange", "kiwi", "melon", "mango"]  
print(thislist[:4])



thislist = ["apple", "banana", "cherry", "orange", "kiwi", "melon", "mango"]  
print(thislist[2:])



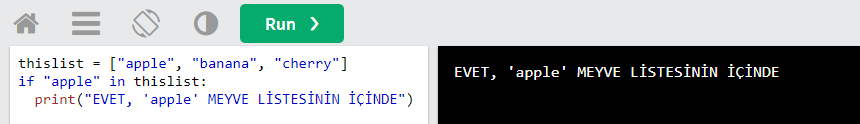
### Range of Negative Indexes

thislist = ["apple", "banana", "cherry", "orange", "kiwi", "melon", "mango"]  
print(thislist[-4:-1])



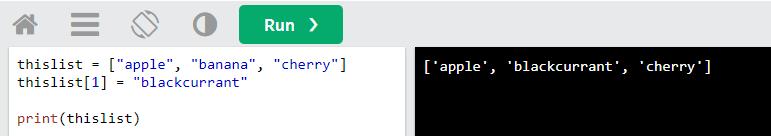
### Check if Item Exists

thislist = ["apple", "banana", "cherry"]  
if "apple" in thislist:  
  print("Yes, 'apple' is in the fruits list")



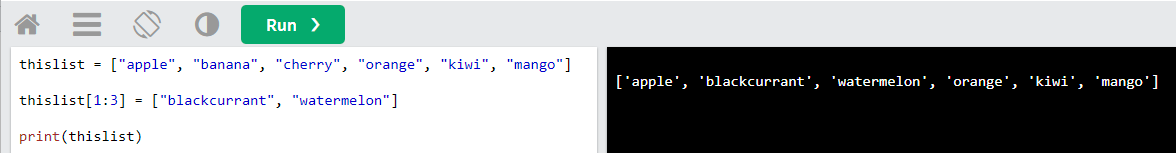
## Change List Items

### Change Item Value

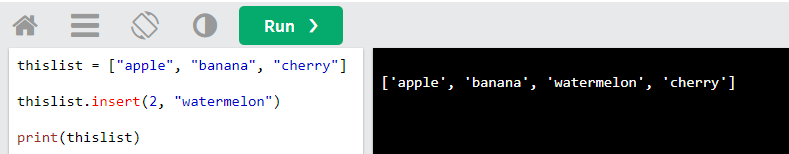


### Change a Range of Item Values

"Muz" ve "kiraz" değerlerini "kara frenk üzümü" ve "karpuz" değerleriyle değiştirin:



### Insert Items

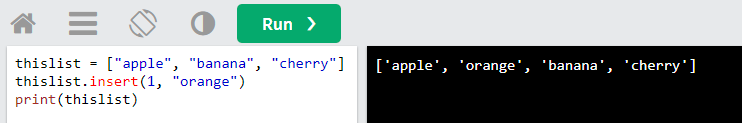


## Add List Items

### Append Items

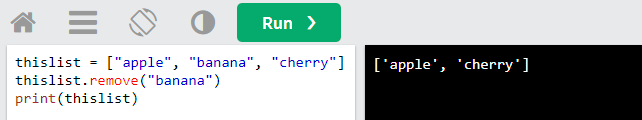


### Insert Items

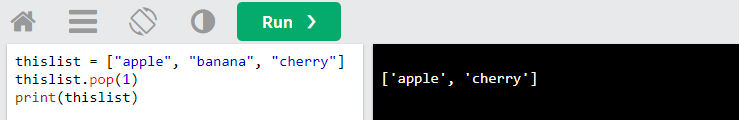


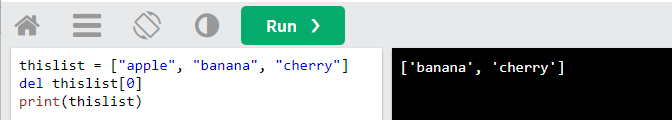
## Remove List Items

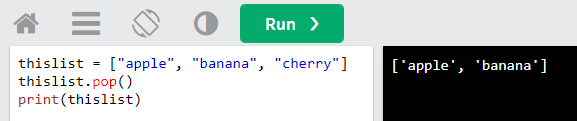
### Remove Specified Item

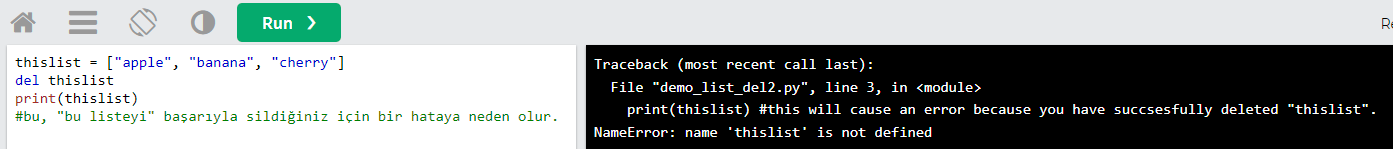


### Remove Specified Index – pop – Sondan Silme

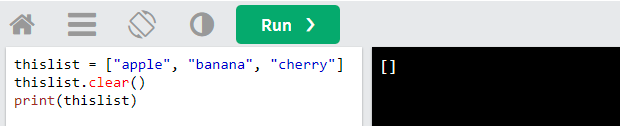








### Clear the List

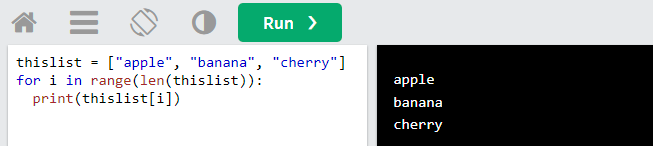


## Loop Lists

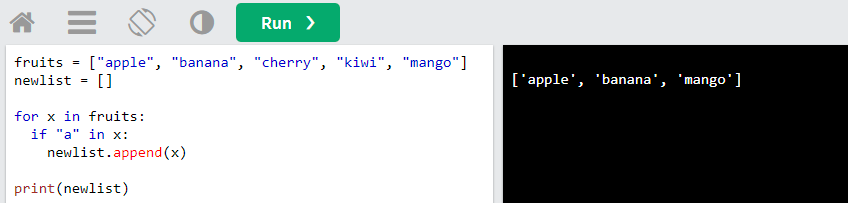
### Loop Through a List – Listede Döndür

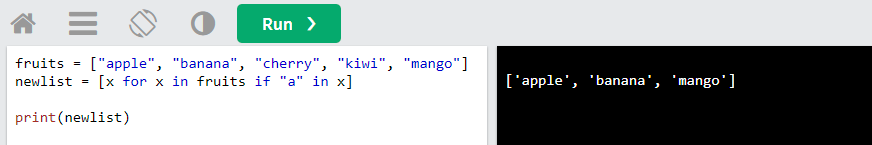


### Loop Through the Index Numbers – Dizin Numaralarında Döndürme



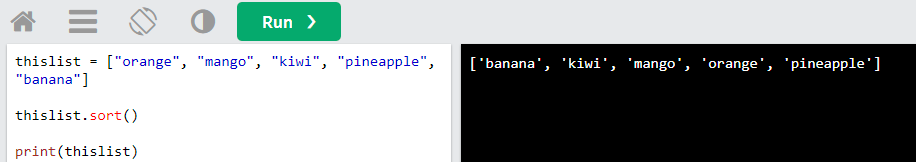
## List Comprehension – Liste içinde arama ve yeni liste oluşturma

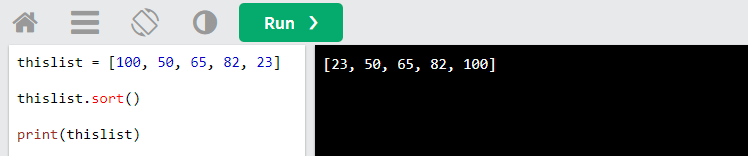




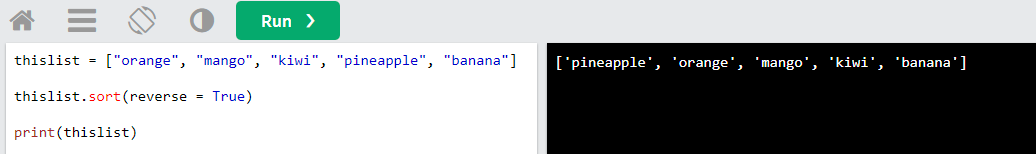
## Sort List

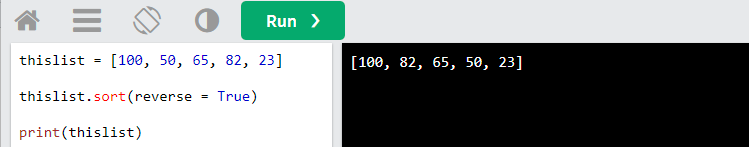
### Sort List Alphanumerically



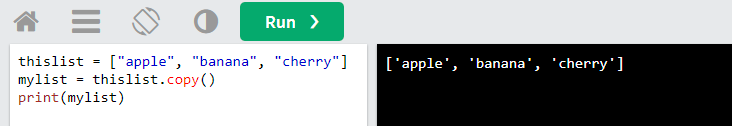


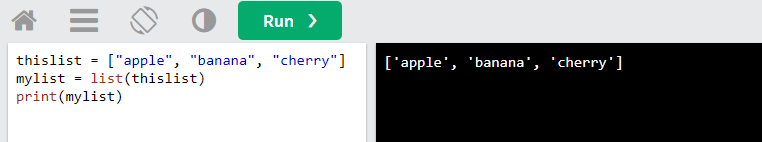
### Sort Descending



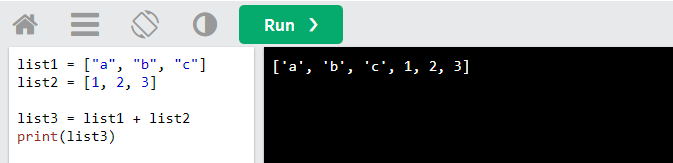


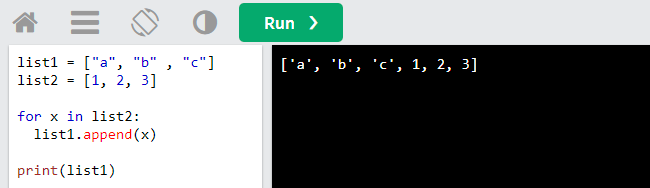
## Copy Lists

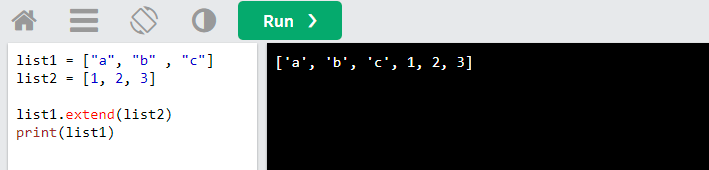




## Join Lists







## List Methods

|  |  |
| --- | --- |
| [append()](https://www.w3schools.com/python/ref_list_append.asp) | Listenin sonuna bir öğe ekler |
| [clear()](https://www.w3schools.com/python/ref_list_clear.asp) | Tüm öğeleri listeden kaldırır |
| [copy()](https://www.w3schools.com/python/ref_list_copy.asp) | Listenin bir kopyasını döndürür |
| [count()](https://www.w3schools.com/python/ref_list_count.asp) | Belirtilen değere sahip öğelerin sayısını döndürür |
| [extend()](https://www.w3schools.com/python/ref_list_extend.asp) | Geçerli listenin sonuna bir listenin öğelerini (veya herhangi bir yinelenebilir) ekler |
| [index()](https://www.w3schools.com/python/ref_list_index.asp) | Belirtilen değere sahip ilk öğenin dizinini döndürür |
| [insert()](https://www.w3schools.com/python/ref_list_insert.asp) | Belirtilen konuma bir eleman ekler |
| [pop()](https://www.w3schools.com/python/ref_list_pop.asp) | Öğeyi belirtilen konumda kaldırır |
| [remove()](https://www.w3schools.com/python/ref_list_remove.asp) | Belirtilen değere sahip öğeyi kaldırır |
| [reverse()](https://www.w3schools.com/python/ref_list_reverse.asp) | Listenin sırasını tersine çevirir |
| [sort()](https://www.w3schools.com/python/ref_list_sort.asp) | Listeyi sıralar |

.

#### Fatih Ercan \_ List

#2.GÜN  
# LIST  
myList = [1, "iki", True, 3.14, [1, "Ali", False]]  
print(myList[4][2])  
  
# Listedye ekleme , silme  
names = ["Ali", "Veli", "Osman"]  
names.append("Hakkı") # yukarıdaki listeye ilave etmek için append komutunu kullan.  
names[0] = "Aslı" # istenilen index numarasına göre ekleme yapar  
names.insert(2,"Erdem") # istenilen index numarasına göre ekleme yapar  
names.remove("Veli") # istediğin bir veriyi silme kiçin kullanılır.  
names.pop() # pop özelliği listenin en son elimanını siler.  
names.pop(0) # index sayısını yazarsak o indeksi de siler. Dikkat! olamayan indexi silemez. Hataya düşer.  
print(names)  
  
#listeyi sıralama  
letters = ["A", "Z", "B", "F", "L"]  
numbers = [1,2,3,4,5,6,56,81,42,1]  
# letters.sort() # sıralamak için kullanılır  
# numbers.sort(reverse=True) # büyükten küçüğe doğru sıralar  
newNumbers = numbers.copy()  
newNumbers2 = numbers  
# print(letters)  
# print(numbers)  
print(newNumbers)

#### Soru1=Cevap

strings = input("7adet karakter girin: ")   
mylist = strings.split()  
print(mylist)  
print(len(mylist))  
mylist[0] = "FATİH"  
print(mylist)  
newList = mylist[2:6]  
newList.reverse()  
print(newList.reverse())  
  
if "FATİH" in mylist:  
 print("Yeminle var.")

#### Soru = Cevap

*"""  
1- Ekrana Meraha Yazan ve isim parametresi olan bir fonksiyon yazın  
2- Gönderilen bir sayının tam bölenlerini bulan ve Liste şeklinde geri döndüren bir fonksiyon yazın  
3- bir strinig ifadesinin sesli harflerini silen bir fonksiyon yazın  
 "merhaba" -> "mrhb"  
 3.1. - Sesli harfler -> a,e,i,ı,o,ö,u,ü  
4- Gönderilen 2 farklı veri setini aynı değere sahip olanlarını bulan bir fonksiyon yazınız.  
 -> list1= [1,2,5,7,8] , list2 = [1,asdas,7,2,9] [1,2,7]  
"""*#1'inci sorunun cevabı  
def Hello(name):  
 print(f"Merhaba {name}")  
Hello("TKGM") # burada fonksiyonu çağırıyoruz.  
  
#2'inci sorunun cevabı  
def tamBolunenleriBul(sayi):  
 result=[]  
 for x in range(2,sayi+1):  
 if sayi % x == 0:  
 result.append(x)  
 return result  
print(tamBolunenleriBul(10)) # buradaki 10 ifadesi döngünün kaça kadar gideceği ile ilgilidir.  
  
#3  
def silSesli(metin):  
 metin = str(metin)  
 metin = metin.replace("a","")  
 metin = metin.replace("e", "")  
 metin = metin.replace("ı", "")  
 metin = metin.replace("i", "")  
 metin = metin.replace("o", "")  
 metin = metin.replace("ö", "")  
 metin = metin.replace("u", "")  
 metin = metin.replace("ü", "")  
 metin = metin.replace("ğ", "")  
 return metin  
print(silSesli("Merhaba benim adım yağmur"))  
  
#4  
def findSome(veriSet1,veriSet2):  
 veriSet1 = set(veriSet1)  
 veriSet1 = set(veriSet2)  
  
 result = veriSet1.intersection(veriSet2)  
 return result  
  
list1 = [1,2,"ali",12,43,12312]  
list2 = [1,"asd","odasd",123,55,42]  
  
print(findSome(list1,list2))

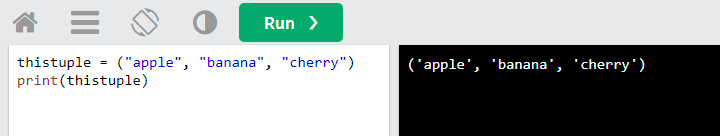
#### Soru = Cevap

*"""  
studets = {  
 "Ogrno": {  
 "name": "value",  
 "className" : "11A",  
 "notes": {  
 "math": 10,  
 "bio": 20,  
 "phsy": 100  
 }  
 },  
 "Ogrno2": {  
 "name": "value",  
 "className" : "11A",  
 "notes": {  
 "math": 10,  
 "bio": 20,  
 "phsy": 100  
 }  
 },  
}  
  
1 - 3 adet Öğrenci Bilgisi Alınız. OK  
 1.1 - Boş Değer Girilemez. OK  
  
2 - Sırası ile öğrenci notlarını sorunuz (0-100). OK  
 2.1 - Hatalı Girişlerde hata fırlatarak ekrana tekar sorulacak. OK  
  
3 - Girilen Öğrenci Notlaına Göre Derslerin durumunu ekrana yazdırınız.  
 3.1 - 0-39 : 1 , 40-49 : 2 , 50 - 69 : 3, 70-84: 4 , 85-100 : 5  
  
4 - Sınıf Genel Durum Yazılacak. Ortalama 0-49 : Kaldı , 50-100 : Geçti  
  
"""*# Stage 1  
# 1 - 3 adet Öğrenci Bilgisi Alınız. OK  
# 1.1 - Boş Değer Girilemez. OK  
students = {}  
def getStudentInfo(sira,message):  
 while True:  
 try:  
 result = input(f"{sira}.{message} : ")  
 if not result:  
 raise ValueError(f"{message} Boş Olamaz...")  
 else:  
 return result  
 except Exception as ex:  
 print(f"Hata : {ex}")  
  
for i in range(1, 2):  
 ogrenciNo = getStudentInfo(i, "Öğrenci No")  
 ogrenciAdi = getStudentInfo(i, "Öğrenci Adı")  
 ogrenciSinifi = getStudentInfo(i,"Öğrenci Sınıfı")  
  
 students[ogrenciNo] = {  
 "name": ogrenciAdi,  
 "className": ogrenciSinifi,  
 }  
  
print(students)  
  
  
# Stage 2  
# 2 - Sırası ile öğrenci notlarını sorunuz (0-100). OK  
# 2.1 - Hatalı Girişlerde hata fırlatarak ekrana tekar sorulacak. OK  
def getOgrenciNotes(sira, message):  
 while True:  
 try:  
 result = input(f"{sira}.{message} : ")  
 if not result.isdigit():  
 raise Exception(f"{message} Matematik Notu Sadece Sayısal Olmalı....")  
 else:  
 mathNote = float(mathNote)  
 if mathNote < 0 or mathNote > 100:  
 raise Exception("Hata : Matemnatik Notu 0 ile 100 Arasında Olmalıdır...")  
 else:  
 break  
 except Exception as ex:  
 print(ex)  
  
for key in students.keys():  
 ogrenciNo = getStudentInfo(key, "Öğrenci No")  
 mathNote = getStudentInfo(key, "Metamatik Notu")  
 bioNote = getStudentInfo(key, "Biyoloji Notu")  
 phsyNote = getStudentInfo(key, "Fizik Notu")  
  
 students[ogrenciNo]["notes"] = {  
 "Matematik" : mathNote,  
 "Biyoloji" : bioNote,  
 "Fizik" : phsyNote,  
 }  
print(students[ogrenciNo]["notes"])  
  
  
# Stage 3  
# 3 - Girilen Öğrenci Notlaına Göre Derslerin durumunu ekrana yazdırınız.  
# 3.1 - 0-39 : 1 , 40-49 : 2 , 50 - 69 : 3, 70-84: 4 , 85-100 : 5  
  
  
  
  
# Stage 4  
#4 - Sınıf Genel Durum Yazılacak. Ortalama 0-49 : Kaldı , 50-100 : Geçti

#### Hocanın çözümü

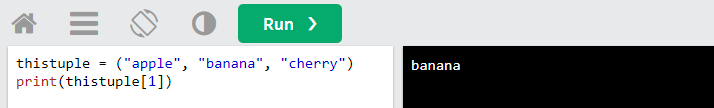
*"""  
studets = {  
 "Ogrno": {  
 "name": "value",  
 "className" : "11A",  
 "notes": {  
 "math": 10,  
 "bio": 20,  
 "phsy": 100  
 }  
 },  
 "Ogrno2": {  
 "name": "value",  
 "className" : "11A",  
 "notes": {  
 "math": 10,  
 "bio": 20,  
 "phsy": 100  
 }  
 },  
}  
  
1 - 3 adet Öğrenci Bilgisi Alınız. OK  
1.1 - Boş Değer Girilemez. OK  
  
2 - Sırası ile öğrenci notlarını sorunuz (0-100). OK  
2.1 - Hatalı Girişlerde hata fırlatarak ekrana tekar sorulacak. OK  
  
3 - Girilen Öğrenci Notlaına Göre Derslerin durumunu ekrana yazdırınız.  
3.1 - 0-39 : 1 , 40-49 : 2 , 50 - 69 : 3, 70-84: 4 , 85-100 : 5  
  
4 - Sınıf Genel Durum Yazılacak. Ortalama 0-49 : Kaldı , 50-100 : Geçti  
  
"""*students = {}  
mathName = "Matematik"  
bioName = "Biyoloji"  
phsyName = "Fizik"  
  
def getStudentInfo(sira, message):  
 while True:  
 try:  
 result = input(f"{sira}. {message} : ")  
 if not result:  
 raise ValueError(f"{message} Boş Olamaz...")  
 else:  
 return result  
 except Exception as ex:  
 print(f"Hata : {ex}")  
  
def getLessonNote(sNumber, lessonName):  
 while True:  
 try:  
 note = input(f"{sNumber} No'lu Öğrenci {lessonName} Not: ")  
 if not note.isdigit():  
 raise Exception(f"Hata : {lessonName} Notu Sadece Sayısal Olmalı....")  
 else:  
 note = float(note)  
 if note < 0 or note > 100:  
 raise Exception(f"Hata : {lessonName} Notu 0 ile 100 Arasında Olmalıdır...")  
 else:  
 return note  
 except Exception as ex:  
 print(ex)  
  
def writeLessonGeneralNote(noteValue, className, name, lesson, sNumber):  
 noteResult = 0  
 if noteValue < 39:  
 noteResult = 1  
 elif 39 < noteValue < 50:  
 noteResult = 2  
 elif 49 < noteValue < 70:  
 noteResult = 3  
 elif 69 < noteValue < 85:  
 noteResult = 4  
 else:  
 noteResult = 5  
  
 print(f"{sNumber} No'lu Öğrenci {className} sınıfında okuyan {name} "  
 f"adlı öğrencinin {lesson} ders sonucu : {noteResult} ")  
  
for i in range(1, 3):  
  
 ogrenciNo = getStudentInfo(i, "Öğrenci No")  
 ogrenciAdi = getStudentInfo(i, "Öğrenci Adı")  
 ogrenciSinifi = getStudentInfo(i, "Öğrenci Sınıfı")  
  
 students[ogrenciNo] = {  
 "name": ogrenciAdi,  
 "className": ogrenciSinifi  
 }  
  
 mathNote = getLessonNote(ogrenciNo, mathName)  
 bioNote = getLessonNote(ogrenciNo, bioName)  
 phsyNote = getLessonNote(ogrenciNo, phsyName)  
  
 students[ogrenciNo]["notes"] = {  
 mathName: mathNote,  
 bioName: bioNote,  
 phsyName: phsyNote,  
 }  
  
 writeLessonGeneralNote(mathNote, ogrenciSinifi, ogrenciAdi, mathName, ogrenciNo)  
 writeLessonGeneralNote(bioNote, ogrenciSinifi, ogrenciAdi, bioName, ogrenciNo)  
 writeLessonGeneralNote(phsyNote, ogrenciSinifi, ogrenciAdi, phsyName, ogrenciNo)

# Tuple

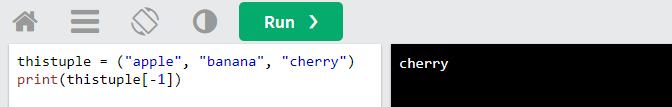


## Access Tuple Items

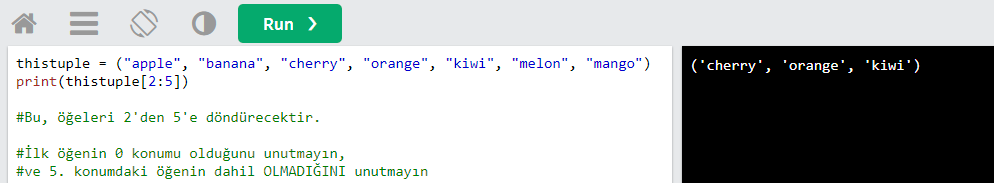
### Access Tuple Items



### Negative Indexing

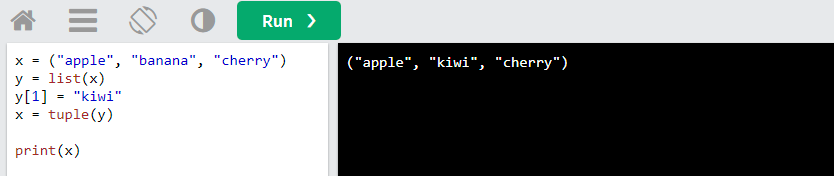


### Range of Indexes

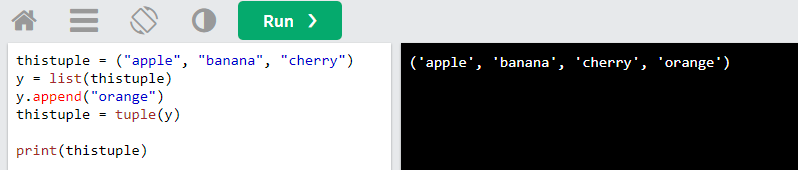


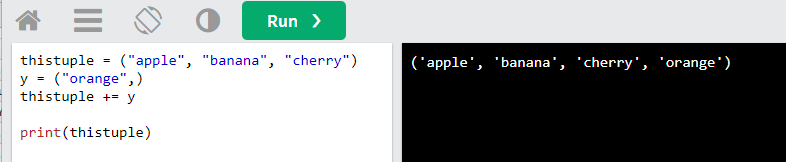
## Update Tuples

### Change Tuple Values

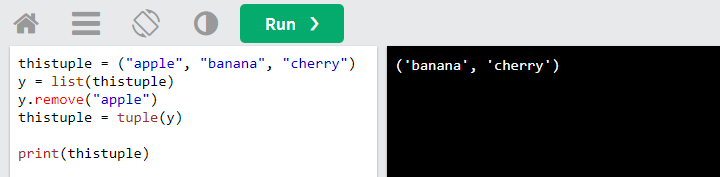


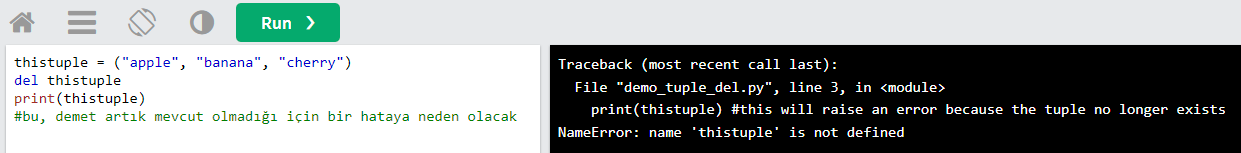
### Add Items





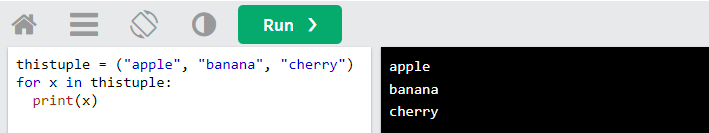
### Remove Items



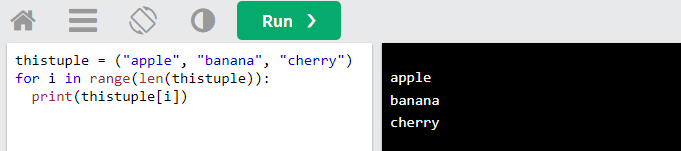


## Loop Tuples

### Loop Through a Tuple

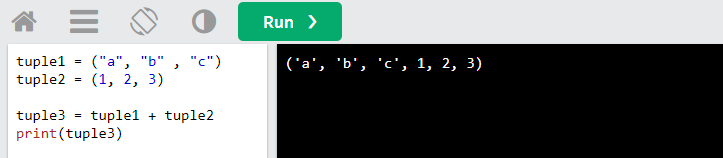


### Loop Through the Index Numbers

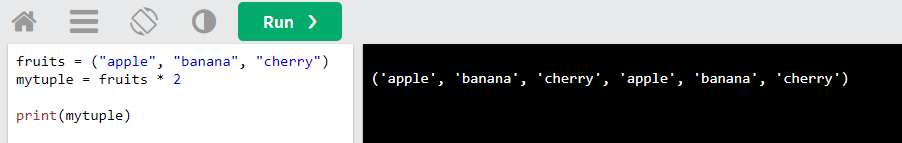


## Join Tuples

### Join Two Tuples



### Multiply Tuples



## Tuple Methods

[count()](https://www.w3schools.com/python/ref_tuple_count.asp) - Bir tanımlama grubunda belirtilen bir değerin kaç kez oluştuğunu döndürür

[index()](https://www.w3schools.com/python/ref_tuple_index.asp) - Tuple'ı belirtilen bir değer için arar ve bulunduğu yerin konumunu döndürür

#### **Tersten yazdırma:** print ([::-1])

#### **Üçüncü, dördüncü ve beşinci öğeyi döndür:** print(thistuple[2:5])

#### **Kadar döndür ama .. dahil etme :** print(thistuple[:4])

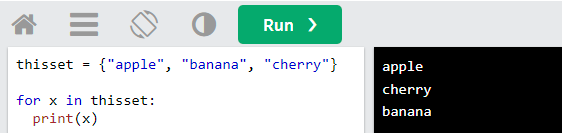
#### **… dan ve sonuna kadar döndür :** print(thistuple[2:])

#### **dizin -4'ten (dahil) dizin -1'e (hariç tutulan) öğeleri döndürür :** print(thistuple[-4:-1])

# Sets – Küme

myset = {"apple", "banana", "cherry"}

## Access Set Items

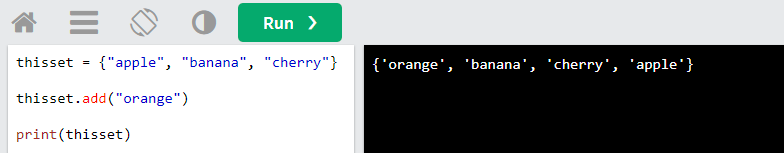


## Change Items

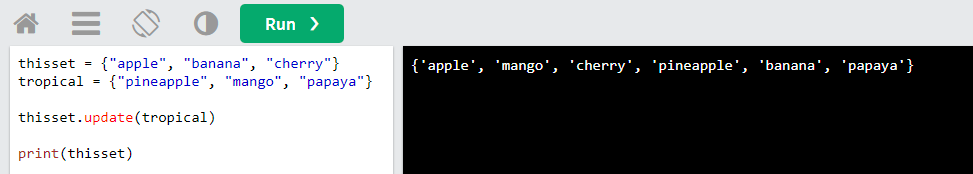
Bir set oluşturulduktan sonra öğelerini değiştiremezsiniz, ancak yeni öğeler ekleyebilirsiniz.

## Add Set Items

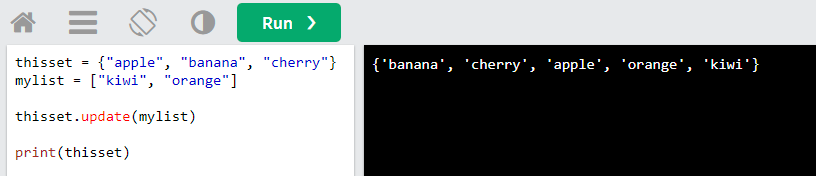
### Add Items



### Add Set – Update

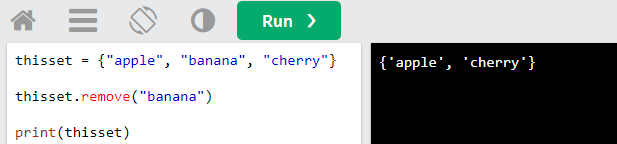


### Add Any Iterable

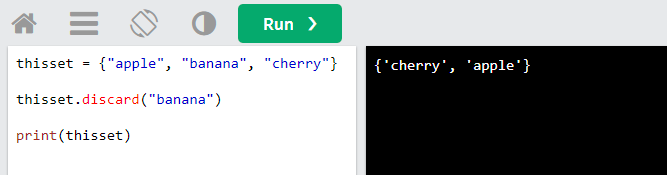


## Remove Set Items

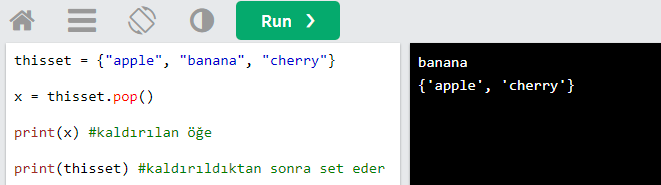
### Remove Item



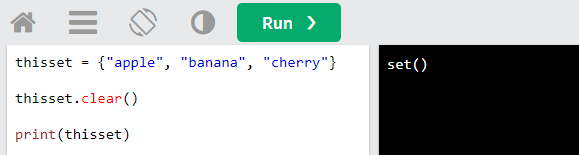
### Discard Item



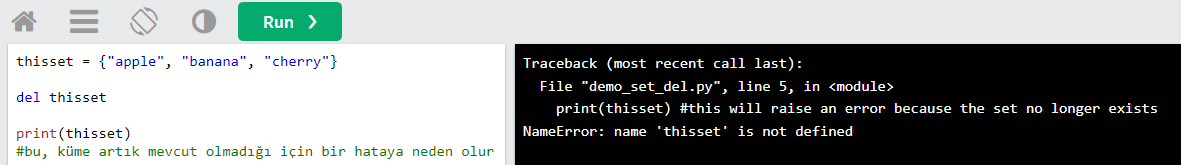
### Pop Item



### Clear Item

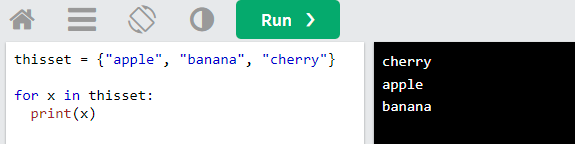


### Delete Item



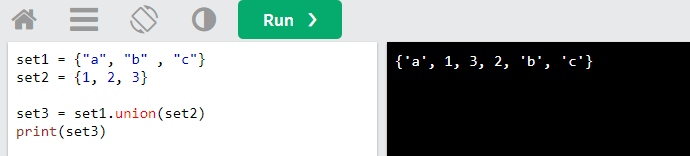
## Loop Sets

### Loop Items

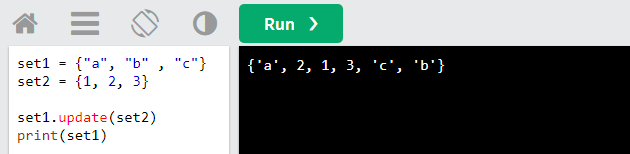


## Join Sets

### union() - Birleştirme



### update()



## Set Methods

|  |  |
| --- | --- |
| [add()](https://www.w3schools.com/python/ref_set_add.asp) | Kümeye bir eleman ekler |
| [clear()](https://www.w3schools.com/python/ref_set_clear.asp) | Kümedeki tüm öğeleri kaldırır |
| [copy()](https://www.w3schools.com/python/ref_set_copy.asp) | Setin bir kopyasını döndürür |
| [difference()](https://www.w3schools.com/python/ref_set_difference.asp) | İki veya daha fazla küme arasındaki farkı içeren bir küme döndürür |
| [difference\_update()](https://www.w3schools.com/python/ref_set_difference_update.asp) | Bu kümedeki, ayrıca belirtilen başka bir kümeye dahil olan öğeleri kaldırır |
| [discard()](https://www.w3schools.com/python/ref_set_discard.asp) | Belirtilen öğeyi kaldır |
| [intersection()](https://www.w3schools.com/python/ref_set_intersection.asp) | Diğer iki kümenin kesişimi olan bir küme döndürür |
| [intersection\_update()](https://www.w3schools.com/python/ref_set_intersection_update.asp) | Bu sette, belirtilen diğer set(ler)de bulunmayan öğeleri kaldırır |
| [isdisjoint()](https://www.w3schools.com/python/ref_set_isdisjoint.asp) | İki kümenin kesişimi olup olmadığını döndürür |
| [issubset()](https://www.w3schools.com/python/ref_set_issubset.asp) | Başka bir kümenin bu kümeyi içerip içermediğini döndürür |
| [issuperset()](https://www.w3schools.com/python/ref_set_issuperset.asp) | Bu kümenin başka bir küme içerip içermediğini döndürür |
| [pop()](https://www.w3schools.com/python/ref_set_pop.asp) | Kümeden bir öğeyi kaldırır |
| [remove()](https://www.w3schools.com/python/ref_set_remove.asp) | Belirtilen öğeyi kaldırır |
| [symmetric\_difference()](https://www.w3schools.com/python/ref_set_symmetric_difference.asp) | İki kümenin simetrik farklarına sahip bir küme döndürür |
| [symmetric\_difference\_update()](https://www.w3schools.com/python/ref_set_symmetric_difference_update.asp) | bu kümeden ve diğerinden simetrik farkları ekler |
| [union()](https://www.w3schools.com/python/ref_set_union.asp) | Kümelerin birleşimini içeren bir küme döndür |
| [update()](https://www.w3schools.com/python/ref_set_update.asp) | Seti bu setin ve diğerlerinin birleşimiyle güncelleyin |

## Fatih Ercan \_ Set (Küme Anlatımı)

mySet1 = {1,2,3,4,78,213,1,46,78,"asdasd"} # setlerde(kümelerde) aynı olan verileri almaz  
print(type(mySet1))  
mySet1.add("Erdem") # ekler listeye  
print(mySet1)  
  
mySet1.remove("Erdem") # siler listeye ancak aradığını bulamadığında hata verir  
# ama remove yerine discard kullanıldığında hataya düşmeden siler yada silemez.  
mySet1.add(5) #   
mySet1.add("Erdem") # ekler listeye  
print(mySet1)

##### issubset = alt küme

##### defference = farklı olanlar

##### intersection = kesişim

###### Soru2=cevap

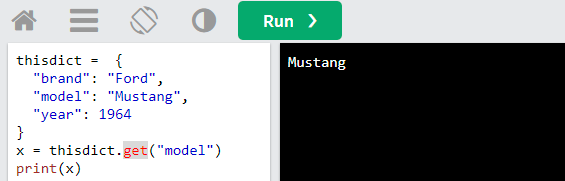
setA = input("Birinci küme elamanını giriniz: ")  
setB = input("İkinci küme elamanını giriniz: ")  
kume1 = set(setA)  
kume2 = set(setB)  
print(setA)  
print(setB)  
print(kume1)  
print(kume2)  
  
kesisim = kume1.intersection(kume2)  
print("kesisim : ", kesisim)  
  
fark = kume2.difference(kume1)  
print("farklıolan : ", fark)

# Dictionaries – Sözlük

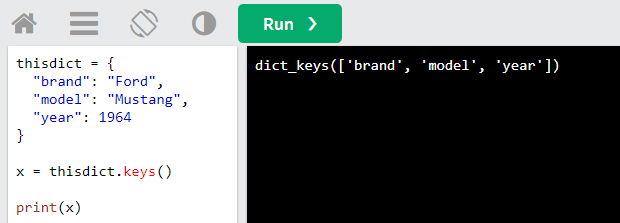
thisdict = {  
  "brand": "Ford",  
  "model": "Mustang",  
  "year": 1964  
}  
x = thisdict["model"]

## ACCESSİNG ITEMS

### Get()



### Keys()



## CHANGE ITEM

### Change Values

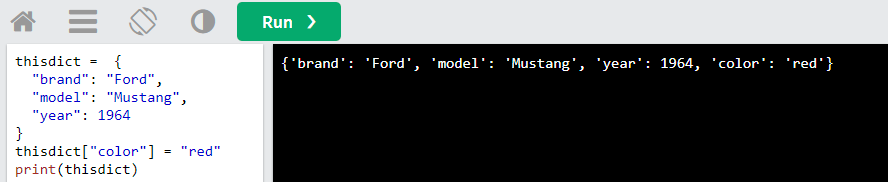


### Update Dictionary

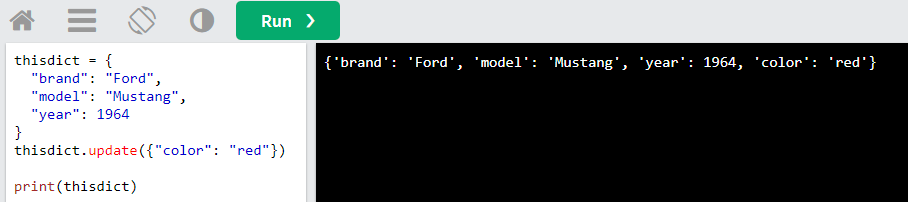


## ADD ITEM

### Adding Items

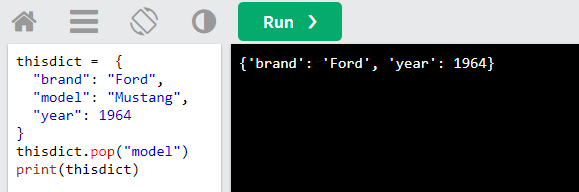


### Update Dictionary

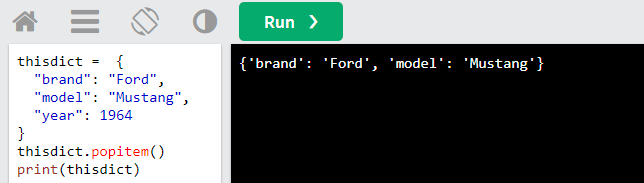


## REMOVE ITEM

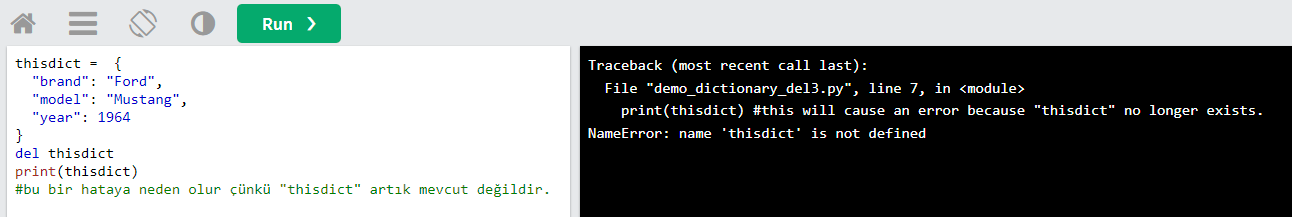
### Pop()



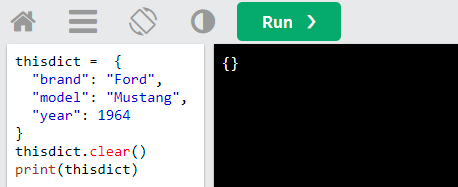
### Popitem()



### Del

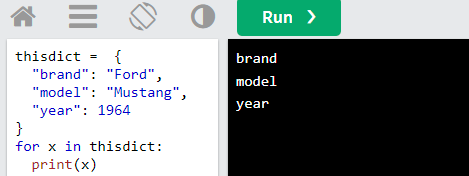


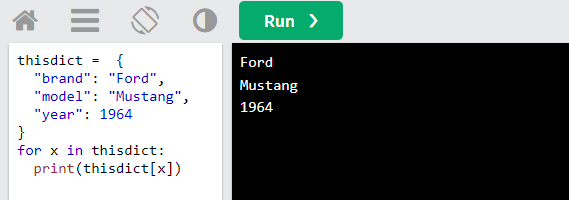
### Clear()



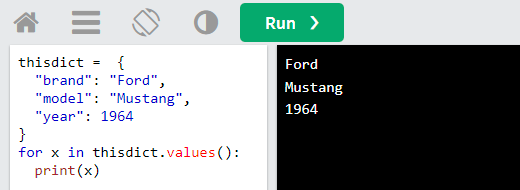
## LOOP DİCTİONERİES

### Loop Through A Dictionary

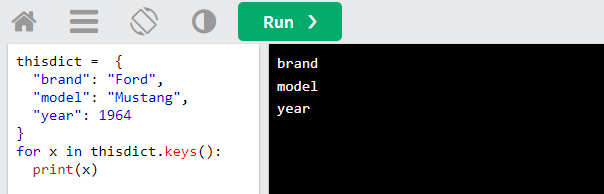




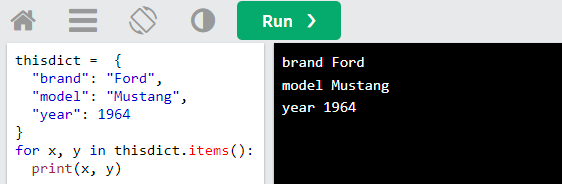
### Values()



### Keys()

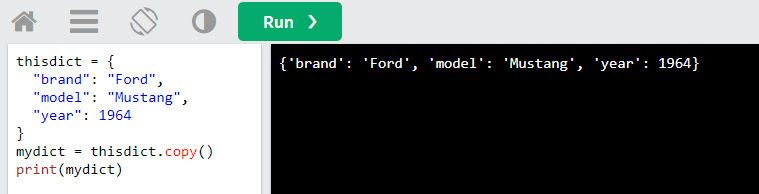


### İtems()

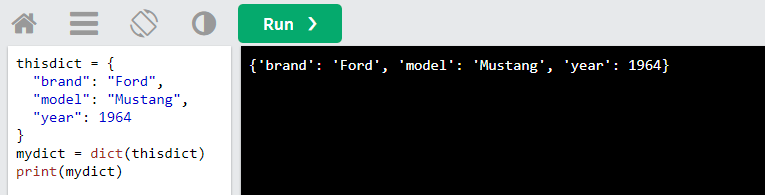


## COPY DICTIONERIES

### Copy()



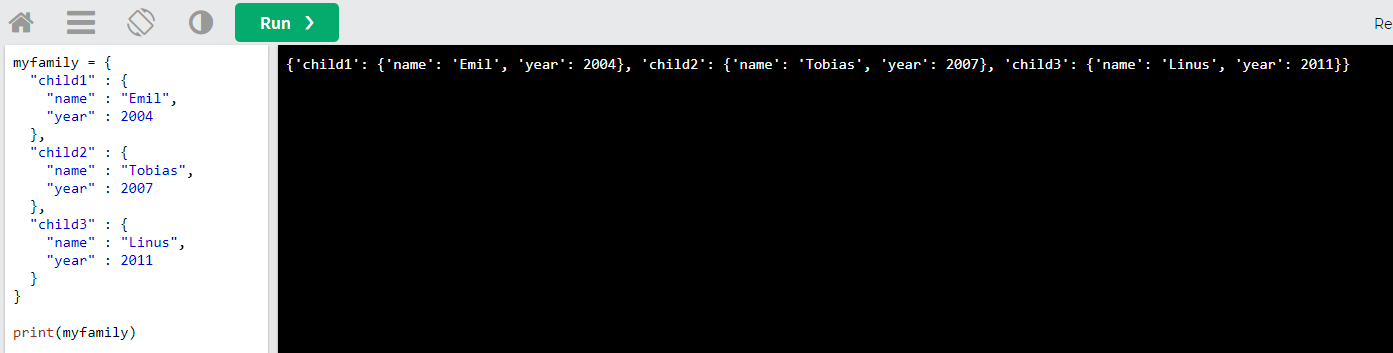
### Dict()



## NESTED DICTIONERIES – İç içe Sözlükler

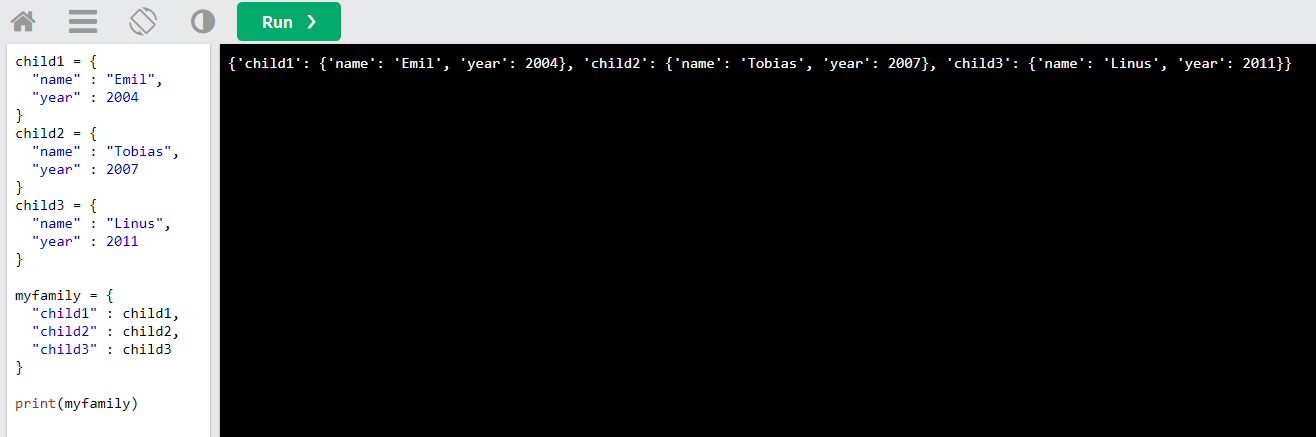
Örnek 1:

myfamily = {  
  "child1" : {  
    "name" : "Emil",  
    "year" : 2004  
  },  
  "child2" : {  
    "name" : "Tobias",  
    "year" : 2007  
  },  
  "child3" : {  
    "name" : "Linus",  
    "year" : 2011  
  }  
}



Örnek 2:

child1 = {  
  "name" : "Emil",  
  "year" : 2004  
}  
child2 = {  
  "name" : "Tobias",  
  "year" : 2007  
}  
child3 = {  
  "name" : "Linus",  
  "year" : 2011  
}  
  
myfamily = {  
  "child1" : child1,  
  "child2" : child2,  
  "child3" : child3  
}



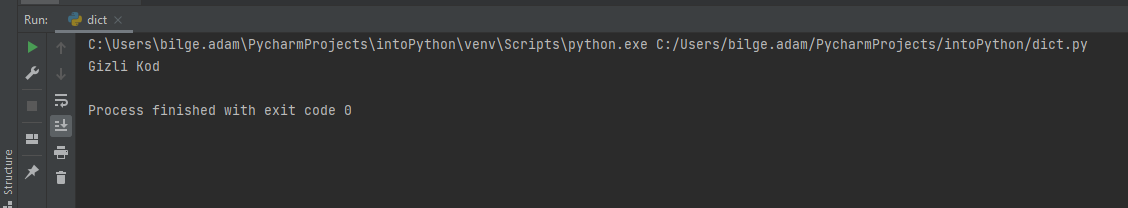
## DICTIONERAY METHODS

|  |  |
| --- | --- |
| [clear()](https://www.w3schools.com/python/ref_dictionary_clear.asp) | Tüm öğeleri sözlükten kaldırır |
| [copy()](https://www.w3schools.com/python/ref_dictionary_copy.asp) | Sözlüğün bir kopyasını döndürür |
| [fromkeys()](https://www.w3schools.com/python/ref_dictionary_fromkeys.asp) | Belirtilen anahtarlara ve değere sahip bir sözlük döndürür |
| [get()](https://www.w3schools.com/python/ref_dictionary_get.asp) | Belirtilen anahtarın değerini döndürür |
| [items()](https://www.w3schools.com/python/ref_dictionary_items.asp) | Her anahtar değer çifti için bir tanımlama grubu içeren bir liste döndürür |
| [keys()](https://www.w3schools.com/python/ref_dictionary_keys.asp) | Sözlüğün anahtarlarını içeren bir liste döndürür |
| [pop()](https://www.w3schools.com/python/ref_dictionary_pop.asp) | Belirtilen anahtarla öğeyi kaldırır |
| [popitem()](https://www.w3schools.com/python/ref_dictionary_popitem.asp) | Son eklenen anahtar/değer çiftini kaldırır |
| [setdefault()](https://www.w3schools.com/python/ref_dictionary_setdefault.asp) | Belirtilen anahtarın değerini döndürür. Anahtar yoksa: anahtarı belirtilen değerle girin |
| [update()](https://www.w3schools.com/python/ref_dictionary_update.asp) | Sözlüğü belirtilen anahtar/değer çiftleriyle günceller |
| [values()](https://www.w3schools.com/python/ref_dictionary_values.asp) | Sözlükteki tüm değerlerin bir listesini döndürür |

## Fatih Ercan – Dictionery (Sözlük) Anlatımı

users = {  
 "username" :{  
 "password" : "123456",  
 "mail" : "ad@asd.com",  
 "name" : "erdem",  
 "roles": ["admin","guest"],  
 "special" : {  
 "motherName" : "Anne",  
 212 : "Gizli Kod"  
 }  
 }  
}  
print(users["username"]["special"][212])

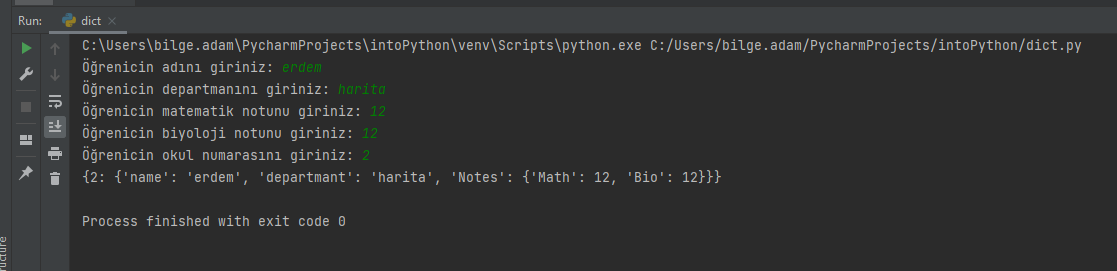
sonuç:



### Soru=cevap

var1 = input("Öğrenicin adını giriniz: ")  
var2 = input("Öğrenicin departmanını giriniz: ")  
var3 = input("Öğrenicin matematik notunu giriniz: ")  
var4 = input("Öğrenicin biyoloji notunu giriniz: ")  
var5 = input("Öğrenicin okul numarasını giriniz: ")  
  
  
studentname = str(var1)  
studentdep = str(var2)  
studentmatnote = int(var3)  
studentbionote = int(var4)  
studentcode = int(var5)  
  
students = {  
 studentcode : {  
 "name": studentname,  
 "departmant": studentdep,  
 "Notes": {  
 "Math": studentmatnote,  
 "Bio": studentbionote  
 }  
 }  
}  
  
print(students)

sonuç



### örnek

var1 = input("Öğrenicin adını giriniz: ")  
var2 = input("Öğrenicin departmanını giriniz: ")  
var3 = input("Öğrenicin matematik notunu giriniz: ")  
var4 = input("Öğrenicin biyoloji notunu giriniz: ")  
var5 = input("Öğrenicin okul numarasını giriniz: ")  
  
  
studentname = str(var1)  
studentdep = str(var2)  
studentmatnote = int(var3)  
studentbionote = int(var4)  
studentcode = int(var5)  
  
students = {  
 studentcode : {  
 "name": studentname,  
 "departmant": studentdep,  
 "Notes": {  
 "Math": studentmatnote,  
 "Bio": studentbionote  
 }  
 }  
}  
  
print(students)  
  
ort = ((studentmatnote + studentbionote)/2)  
  
if ort >= 50:  
 print("Geçetiniz" )  
else :  
 print("Kaldınız" )  
  
  
#yada alttaki gibi yapılabilir.  
studentcode = int(input("Öğrenci No:"))  
localMath = students[studentcode]["Notes"]["Math"]  
localBio = students[studentcode]["Notes"]["Bio"]  
print((localMath+localBio)/2>=50)

# IF...ELSE

Python, matematikteki olağan mantıksal koşulları destekler:

* Equals: a == b
* Not Equals: a != b
* Less than: a < b
* Less than or equal to: a <= b
* Greater than: a > b
* Greater than or equal to: a >= b

if e bir örnek

a = 33  
b = 200  
if b > a:  
  print("b is greater than a")

## Elif

a = 33  
b = 33  
if b > a:  
  print("b is greater than a")  
elif a == b:  
  print("a and b are equal")



## Else

a = 200  
b = 33  
if b > a:  
  print("b is greater than a")  
elif a == b:  
  print("a and b are equal")  
else:  
  print("a is greater than b")



## Short Hand If

a = 200  
b = 33

if a > b: print("a is greater than b")



## Short Hand If ... Else

a = 2  
b = 330  
print("A") if a > b else print("B")



## And

a = 200  
b = 33  
c = 500  
if a > b and c > a:  
  print("Both conditions are True")



## Or

a = 200  
b = 33  
c = 500  
if a > b or a > c:  
  print("At least one of the conditions is True")



## Nested If

x = 41  
  
if x > 10:  
  print("Above ten,")  
  if x > 20:  
    print("and also above 20!")  
  else:  
    print("but not above 20.")



## The pass Statement

a = 33  
b = 200  
  
if b > a:  
  pass

# bunun gibi boş bir if ifadesine sahip olmak, pass ifadesi olmadan bir hataya neden olur

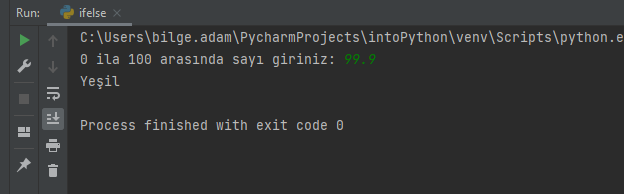
## Fatih Ercan – İf Else

if 60 > 50:  
 print("Geçetiniz" )  
elif 40 < 50:  
 print("Mantıklı")  
else :  
 print("Kaldınız" )

### Soru=Cevap

var1 = float(input("0 ila 100 arasında sayı giriniz: "))  
  
if var1 < 0 or var1 > 100 :  
 print("Aralık dışında bir sayı girdiniz")  
elif var1 <= 40:  
 print("Siyah")  
elif var1 <= 75:  
 print("Turuncu")  
elif var1 <= 100:  
 print("Yeşil")  
else :  
 print("Tekrar Çalıştır")

#### sonuç

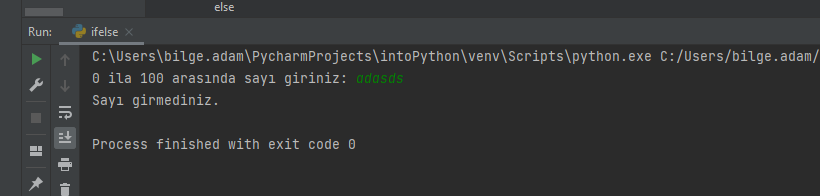


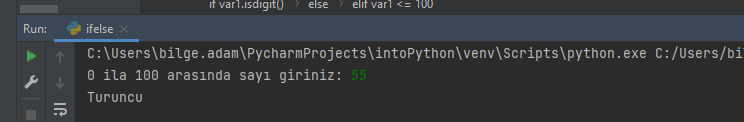
*isdigit*  
Eğer sayı yerine harf girildiğinde programımız patlamasını istiyorsak isdigit özelliği kullanılacak

Örnek:

var1 = input("0 ila 100 arasında sayı giriniz: ")  
if var1.isdigit(): # isdigit özelliği True False şeklinde cevap döndürür  
 var1 = int(var1)  
 if var1 < 0 or var1 > 100:  
 print("Aralık dışında bir sayı girdiniz")  
 else:  
 var1 = int(var1)  
 if var1 <= 40:  
 print("Siyah")  
 elif var1 <= 75:  
 print("Turuncu")  
 elif var1 <= 100:  
 print("Yeşil")  
else:  
 print("Sayı girmediniz.")

sonuç:



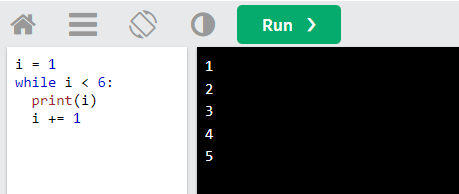


# WHILE LOOPS

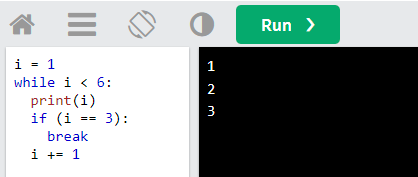
Python'un iki ilkel döngü komutu vardır:

* while loops
* for loops

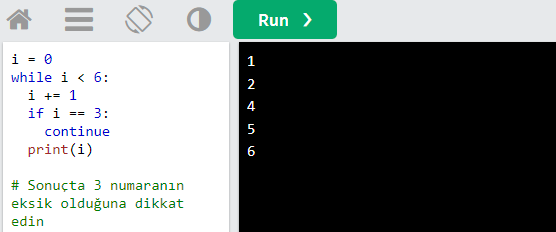
## while



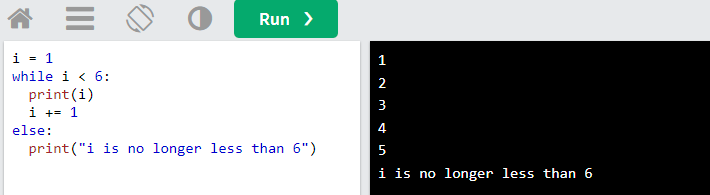
## Break



## Continue



## Else



## Fatih ERCAN Hocanın Anlatımı

#### Örnek 1:

checkPoint = True  
  
while checkPoint:  
 number = input("Sayı Giriniz: ")  
 if number.isdigit():  
 checkPoint = False

#### Örnek 2:

checkPoint = False  
  
while checkPoint:  
 number = input("Sayı Giriniz: ")  
 if number.isdigit():  
 checkPoint = False  
   
else:  
 print("Döngüye Girmedi.")

#### Örnek 3:

Random Örnekleri için altaki siteden bakılabilir.

<https://www.w3schools.com/python/module_random.asp>

import random as \_random  
  
\_random.random()

#### Soru = cevap - tahmin

# 1. Bir sayı üretin (Bu sayı 0-100 arasında oalcak)  
# 2. Kullanıcıdan bir değer girmesini isteyin  
# 3. Eğer Kullanıcının girdiği sayı ile bizim ürettiğimiz sayı aynı ise tebrikler  
# 4. Tahmin ettiği sayı yakınsa yukarı aşağı şeklinde tahmin etmesini isteyeceğiz.  
  
import random as \_random  
sayi = \_random.randint(0,100)  
print(sayi)  
  
  
  
while True:  
 tahminet = int(input("0 ile 100 arasında bir sayı giriniz: "))  
 if sayi == tahminet :  
 print("Tebrikler bildiniz.")  
 break  
 elif tahminet > sayi:  
 print("Yaklaştın! Aşağı")  
 else:  
 print("Yakşlaştın! Yukarı")

#### Soru = cevap - takmin

# 5. beş adet hakkı var. bittiğinde kaybettiniz yazacak  
# 6. her hak 20 puan , 100 den düşecek  
# 7. bildiğinde tebrikler puanınız budur.  
  
import random as \_random  
sayi = \_random.randint(0,100)  
print(sayi)  
  
hak = 5  
puan = 100  
  
while hak >0:  
 tahminet = int(input("0 ile 100 arasında bir sayı giriniz: "))  
 if sayi == tahminet :  
 print(f"Tebrikler bildiniz.... puan: {puan}, ....hak: {hak} ")  
 break  
 elif tahminet > sayi:  
 print("Yaklaştın! Aşağı")  
 else:  
 print("Yakşlaştın! Yukarı")  
  
 hak -=1  
 puan -= 20  
else:  
 print("..........GAME OVER...........")

#### Ödev ?

# Kullanıcı integer girmediği taktirde hakkından düşmeyecek  
# Kullanıcı aynı rakamı girerse eğer hakkından düşmeyecek ve uyaracğız.  
# Kaybettiğinde tahmin etmesi gereken rakamın ne olduğunu ekrana vereceğiz.

#### Soru=cevap \_ bankamatik – erdem’in çözümü

# costmers = {  
# "customerCode" : {  
# "password" : "Metin",  
# "name" : "Metin",  
# "balance" : "float",  
# "addBalance" : "float (2000)"  
# }  
# }  
  
# kullanıcı bize custemercode ve şifresini verecek  
# şifreyi 3 kez deneyebilir. yanlış girerse sistemden atacak  
# Doğru ise içeri girecek  
# Heabında 10000 var 5000 çekecek ve kalan bakiyeyi düşecek  
# Eğer istedi para hesabından büyük ve "Evet" der ise ödeyebileceğim rakam bu diyecek istiyormusun?  
# 2000 TL den küçük ise parayı vereck bakiyeyi güncelleyecek  
  
studentCount = input("Kaç Kullanıcı Bankamıza Kayıt Olacak: ")  
if studentCount.isdigit():  
 studentCount = int(studentCount)  
  
 students = {}  
  
 for x in range(0,studentCount):  
 # sNumber = input("Kullanıcı id'ni tanımla ")  
 sUserName = input("Kullanıcı Adını Tanımla: ")  
 sPassword = input("Parolanı Tanımla: ")  
 students = {  
 "userName" : sUserName,  
 "password" : sPassword  
 }  
 print("--------------------------------------------------")  
  
 while True:  
 kullanici = input("Kullanıcı adınızı giriniz: ")  
 parola = input("Şifrenizi giriniz: ")  
 # ikisi de doğru  
 if kullanici == sUserName and parola == sPassword:  
 print("Hoşgeldiniz")  
 break  
 # kullanıcı adı doğru şifre yanlış  
 elif kullanici == sUserName and parola != sPassword:  
 print("Şifreniz yanlış\n")  
 # şifre doğru kullacı adı yanlış  
 elif kullanici != sUserName and parola == sPassword:  
 print("Kullanıcı adınız yanlış \n")  
 # ikisi de yanlış  
 else:  
 print("Kullanıcı adı ve şifre hatalı\n")  
  
 bakiye = 10000  
  
  
 while True:  
 cekilenpara = int(input("Ne kadar para çekeceksiniz? : "))  
 if bakiye > cekilenpara:  
 print(f"hesabınızdan: {cekilenpara} TL Çektiniz., Kalan bakiyeniz: {bakiye-cekilenpara} ")  
 break  
 elif bakiye < cekilenpara:  
 print(f"Mevcut bakiyenizden fazla çekiyorsunuz. Devam etmek istiyor musunuz? ")  
 cvp = str(input("Evet ise 'e', Hayır ise 'h' yazıp enter'a basın "))  
 if cvp == "e":  
 print(f"hesabınızdan: {cekilenpara} TL Çektiniz., Kalan bakiyeniz: {bakiye-cekilenpara} ")  
 elif cvp == "h":  
 pass  
 else:  
 print("Please enter yes or no.")  
 else:  
 print("dasda")  
  
 # bakiye -= cekilenpara

#### Soru=cevap \_ bankamatik - hoca’nın çözümü

# customers = {  
# "customerCode" : {  
# "password" : "Metin",  
# "name" : "Metin",  
# "balance" : "float",  
# "addBalance" : "float (2000)"  
# }  
# }  
  
# 1 - kullanıcıya customerCode sorulur  
# 2 - eğer kullanıcı var ise şifre sorulaca  
# 3 - şifre hatalı ise en fazla 3 kez deneme yapmasına izin verilcek  
# 4 - sisteme giriş yaptığında hoşgeldiniz.. (name) diyecek ve çekmek istediği rakam sorulacak  
# 5- çekilen rakam bakiyeden küçük yada eşitse pra verilip bakiye güncellencek  
# 6 - değil ise ek hesap kullanmak istemisiniz diye soracak  
# 7 - evet ise ve kalan para 2000 TL den küçük yada eşitse direk verilecek  
# 8 - değil ise en fazla ... verebilirim istermisiniz diye sorulacak  
  
customers = {}  
  
for x in range(0,2):  
  
 cCode = input("Müşteri Numarası Giriniz: ")  
 cName = input("Müşteri Adı Giriniz: ")  
 cBalance = float(input("Müşteri Bakiyesi Giriniz: "))  
 cPassword = input("Müşteri Şifresi Giriniz: ")  
  
 customers[cCode] = {  
 "password" : cPassword,  
 "name" : cName,  
 "balance" : cBalance,  
 "addBalance" : float(2000)  
 }  
print("------------------------------------------------------------")  
print(" ")  
  
customerCode = input("Lütfen Müşteri Numaranızı Giriniz: ")  
if customerCode in customers.keys():  
 customerPassword = input(f"Merhaba {customers[customerCode]['name']}... Şifreniz: ")  
 if customerPassword == customers[customerCode]["password"]:  
 amount = float(input(f"Hoş Geldiniz {customers[customerCode]['name']}... Çekilecek Turar: "))  
 if amount <= customers[customerCode]['balance']:  
 print(f" Mr/Mrs {customers[customerCode]['name']} Hesabınızdan {amount} TL Çektiniz.")  
 customers[customerCode]['balance'] -= amount  
 else:  
 cAnswer = input("Ek Hesap Kulllanmak İster misiniz : Y/N : ")  
 if cAnswer.upper() == "Y":  
 if amount <= customers[customerCode]['balance'] + customers[customerCode]['addBalance']:  
 print(f" Mr/Mrs {customers[customerCode]['name']} Hesabınızdan {amount} TL Çektiniz.")  
 customers[customerCode]["addBalance"] = amount - customers[customerCode]['balance']  
 customers[customerCode]['balance'] = float(0)  
 else:  
 awabAmount = customers[customerCode]['balance'] + customers[customerCode]['addBalance']  
 cAnswer2 = input(f"İstenilen Tutar Toplam Bakiyeden Büyük! "  
 f"{awabAmount} TL Çekmek istermisiniz? Y/N ")  
 if cAnswer2.upper() == "Y":  
 print(f" Mr/Mrs {customers[customerCode]['name']} Hesabınızdan {awabAmount} TL Çektiniz.")  
 customers[customerCode]["addBalance"] = 0.0  
 customers[customerCode]['balance'] = float(0)  
 else:  
 print("İyi Günler...")  
  
 else:  
 print("İyi Günler...")  
  
else:  
 print("Müşteri Numarası Bulunamadı!...")

# FOR LOOPS

fruits = ["apple", "banana", "cherry"]  
for x in fruits:  
  print(x)

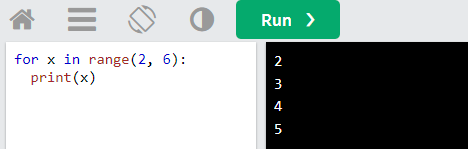
## Break

fruits = ["apple", "banana", "cherry"]  
for x in fruits:  
  print(x)  
  if x == "banana":  
    break

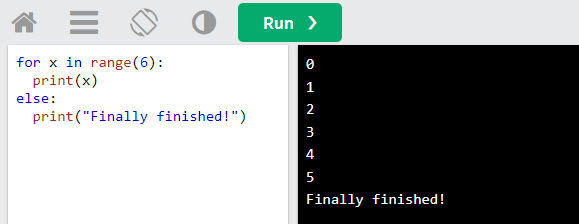
## Continue

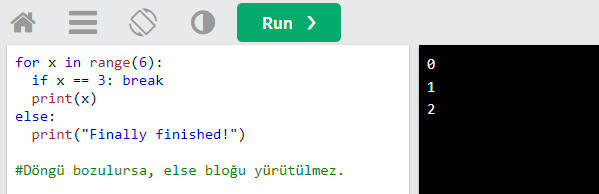
fruits = ["apple", "banana", "cherry"]  
for x in fruits:  
  if x == "banana":  
    continue  
  print(x)

## Range()



## Else in For Loop





## Nested Loop – iç içe for döngüsü



## The pass Statement

for x in [0, 1, 2]:  
  pass

# bunun gibi boş bir for döngüsüne sahip olmak, pass ifadesi olmadan bir hataya neden olur

## Fatih Ercan – For Döngüsü

mySet = [1, "Ali", "Veli", 584,555,32515654, "asdasd"]  
for item in mySet:  
 if item == 584:  
 print(item)  
 break  
 else:  
 print("Bulamadım")  
  
print("----------------------")  
  
for item in mySet:  
 if item == 584:  
 continue  
 print("Bulamadım")

### HomeWork

Ödevler için isterler  
 1. Veri Modeli belli  
 2. Kullanıcıdan 5 tane öğrenci bilgilerini alın (code,name,departman)   
 3. Kullanıcıdan öğrenci nosunu soracağız, girmiş olduğu öğrenci nosu bizde var ise; düzgün bir formatta öğrencinin adını,bölümü falan filan budur. Yoksa böyle bir öğrenci bulunamadı hatasını verdirin.

student = {  
 code : {  
 "name" : sName,  
 "departman" : sDepartman  
 }  
}

#### Ödev’in cevabı

for item in range (1,5):  
 studentcode = int(input(f" {item} Öğrenicin numarasını giriniz: "))  
 studentname = input(f"{item} Öğrenicin adını giriniz: ")  
 studentdpt = input(f" {item} öğrencinin bölümünü giriniz: ")  
  
students = {  
 studentcode : {  
 "name": studentname,  
 "departmant": studentdpt  
 }  
}  
  
var4=int(input ("lütfen öğrenci numarası girin"))  
var5= list(students.keys())  
  
if var4 in var5:  
 print("evet öğrenci sistemde var")  
else:  
 print("öğrenci sistemde yok")

### Ödev için Hocanın yaptığı çözüm

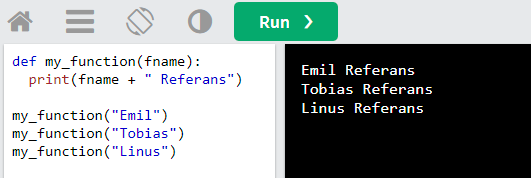
studentCount = input("Girilecel Öğrenci Adedi: ")  
if studentCount.isdigit():  
 studentCount = int(studentCount)  
  
 students = {}  
  
 for x in range(0,studentCount):  
 sNumber = input("Öğrenci No: ")  
 sName = input("Öğrenci Adı: ")  
 sDepart = input("Öğrenci Departmanı: ")  
 students[sNumber] = {  
 "name" : sName,  
 "departmant" : sDepart  
 }  
 print("--------------------------------------------------")  
 studentNo = input("Öğrenci No Girin: ")  
 if studentNo in students.keys():  
 print(f"{studentNo} nolu öğrencinin adı: {students[studentNo]['name']}" \  
 f"bölümü: {students[studentNo]['departmant']}")  
 else:  
 print("Öğrenci no bulunamadı")  
else:  
 print("Geçersiz Bir Sayı Girdiniz.")

# FUNCTİONS

## Fonksiton Oluşturma

def my\_function():  
  print("Hello from a function")

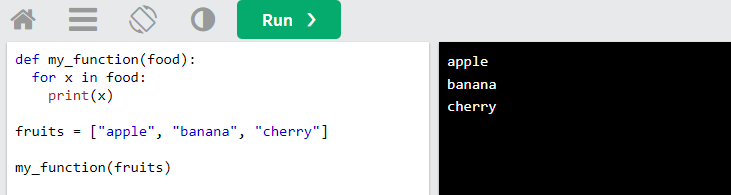
* 1. Fonksiyonu Çağırma

def my\_function(**fname**):  
  print(fname + " Refsnes")  
  
my\_function(**"Emil"**)  
my\_function(**"Tobias"**)  
my\_function(**"Linus"**)

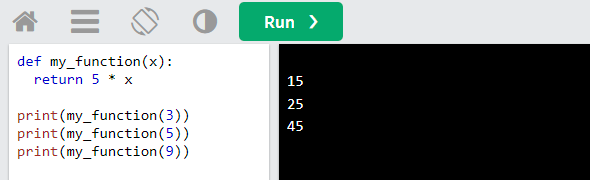
## Bir Listeyi Argüman Olarak Paslama

Bir fonksiyona herhangi bir veri tipi argüman gönderebilirsiniz (dize, sayı, liste, sözlük vb.) ve fonksiyon içinde aynı veri tipi olarak ele alınacaktır.

Örneğin. Argüman olarak bir Liste gönderirseniz, işleve ulaştığında hala bir Liste olacaktır:



## Return Values – Değeri Döndürme



## Fatih Ercan – Funksiyon Anlatımı

### Örnekler:

# ilk örnek  
def sayHello():  
 print(f"Hello")  
sayHello() # burada fonksiyonu çağırıyoruz.  
  
  
# ilk örneğin farklı versiyonu  
def sayHi(name):  
 print(f"Hello {name} ")  
  
sayHi("TKGM") # burada fonksiyonu çağırıyoruz.  
sayHi("Sınıf") # burada fonksiyonu çağırıyoruz.  
sayHi("Erdem") # burada fonksiyonu çağırıyoruz.  
  
# bir başka örnek  
def saySome(param1, param2, param3):  
 print(f"{param1} {param2} {param3}")  
  
saySome("Merhaba", "Ben", "Erdem" ) # burada fonksiyonu çağırıyoruz.  
saySome(param1="Merhaba", param3="Nasılsınız?", param2="TKGM") # burada fonksiyonu çağırıyoruz.  
  
# bir başka örnek (nekadar değer aldığını bilmiyorsak eğer)  
def writeSomething(\*value): # buradaki yıldız(\*) işareti n kadar değer alabilir.  
 for x in value:  
 print(x)  
 print("----------------------------------------------")  
writeSomething("Merhaba", 5, 10, "Ali") # burada fonksiyonu çağırıyoruz.  
writeSomething("Hello") # burada fonksiyonu çağırıyoruz.  
writeSomething("Hello","My","Friend") # burada fonksiyonu çağırıyoruz.  
  
  
def writeDict(\*\*paramDict): # iki yıldız(\*\*) Dictionary (sözlük) olduğunda kullanılur.  
 print(f"Ad: {paramDict['name']} Soyad: {paramDict['surname']}")  
  
writeDict(name="Erdem", surname="ALPAR") # burada fonksiyonu çağırıyoruz.  
  
#-----------------------------------------------------------------------------------------  
#return veren funcition(fonksiyon)  
def getHello(name):  
 return f"Hello {name}"  
  
message = getHello("TKGM")  
print(message)  
  
#farklı bir örneği (matematik hesap versiyonu)  
def getSum(number1, number2):  
 result = number1 + number2  
 return result  
  
 # return number1 + number2 # bu şekilde de yapılabilir.  
  
print(getSum(10,15)) # fonksiyonu print ile de çağırabiliyoruz.  
#----------------------------------------------------------------------------------------

### Doğum Yılı Hesabı\_

*"""  
1- Kullanıcıdan parametre olarak doğum yılı ve ismini alan bir method yazınız.  
2- Bu method kişinin emekli olup olmadığı ekran yazdın  
\* Emeklilik Yaşı 65 Kabul edilir.  
\*\* "Merhaba {ad} emeklisiniz , yada değilsiniz"  
  
3 - Doğum yılının int olup olmaığını ve 1900 ile 2022 arasında olduğunu kontrol ediniz.  
3.1 - Kontrolden geçiyorsa geriye int türünde bir değer döndüren method yazınız.  
  
4 - Kullanıcının girdiği ad bilgisini kontrol ediniz.  
"""*def getYas(dogumYili):  
 return 2022 - dogumYili  
  
  
def getBirth():  
 while True:  
 brithYear = input("Doğum Yılınızı giriniz : ")  
 if not brithYear.isdigit():  
 print("Doğum Yılı Bilgisi Sadece Sayısal Olamlıdır")  
 continue  
 brithYear = int(brithYear)  
 if not (1900 < brithYear < 2022):  
 print("Doğum Yılı 1900 - 2022 aralığında olmallıdır.")  
 continue  
  
 return brithYear  
  
  
def getName():  
 while True:  
 name = input("Lütfen Adınızı Giirniz : ")  
 if not name:  
 print("Ad Alanı Boş Olamaz...")  
 continue  
 if name.isdigit():  
 print("Lütfen İsim Alanına sayısal değer Girmeyiniz...")  
 continue  
  
 return name  
  
  
def emeklilik(dogumYili, isim):  
 yas = getYas(dogumYili)  
 if yas >= 65:  
 print(f"Merhaba {isim} , Emeklisiniz.")  
 else:  
 print(f"Merhaba {isim} , Emekli Değilsiniz.")  
  
  
birthYear = getBirth()  
name = getName()  
  
emeklilik(birthYear, name)

### Asal Sayı \_ Soru=cevap\_

*"""  
1- Gönderilen iki sayı arasındaki asal sayıları ekrana yazdıran bir method (fonksiyon) tanımlayın.  
 1.1. Sayıları kullanıcıdan alarak int kontrolleri ve değerin bir(1) farklı olduğuu kontrol ediniz.  
2-  
"""*def getInt():  
 while True:  
 number = input("Lütfen sayı giriniz: ")  
 if not number:  
 print("sayı alanı boş olamaz")  
 continue  
 if not number.isdigit():  
 print("Sayı girmediniz.")  
 continue  
 number = int(number)  
 if number <= 1:  
 print("Sayı 1'e eşit yada küçük olamaz")  
 continue  
 return int(number)  
  
def numbersControl(number1, number2):  
 if number1 >= number2:  
 raise Exception("ikinci sayı ilk sayıdan büyük olamaz")  
  
def getPrime(number1, number2): # getPrime (prime asal sayı demek)  
 for number in range(number1, number2+1):  
 for x in range(2,number):  
 if number % x == 0:  
 break  
 else:  
 print(number)  
  
  
number1 = getInt()  
  
number2 = 0  
  
while True:  
 try:  
 number2 = getInt()  
 numbersControl(number1,number2)  
 break  
 except Exception as ex:  
 print(ex)  
 continue  
  
getPrime(number1, number2)

# LAMBDA

Bir lambda işlevi, küçük bir anonim işlevdir.

Bir lambda işlevi herhangi bir sayıda argüman alabilir, ancak yalnızca bir ifadeye sahip olabilir.

## Syntax – Yazım şekli

lambda argumanları : ifade

Örnek :

a argümanına 10 ekleyin ve sonucu döndürün:

x = lambda a : a + 10  
print(x(5))

Sonuç: 15 ‘dir

Örnek:

a argümanını b argümanıyla çarpın ve sonucu döndürün:

x = lambda a, b : a \* b  
print(x(5, 6))

Sonuç: 30 ’dur

Örnek:

a, b ve c argümanını özetleyin ve sonucu döndürün:

x = lambda a, b, c : a + b + c  
print(x(5, 6, 2))

Sonuç: 13 ’dür

## Lambda İşlevlerini Niçin Kullanmalı?

Lambda'nın gücü, onları başka bir işlevin içinde anonim bir işlev olarak kullandığınızda daha iyi gösterilir.

Bir argüman alan bir fonksiyon tanımınız olduğunu ve bu argümanın bilinmeyen bir sayı ile çarpılacağını varsayalım:

def myfunc(n):  
  return lambda a : a \* n

Gönderdiğiniz sayıyı her zaman iki katına çıkaran bir işlev yapmak için bu işlev tanımını kullanın:

def myfunc(n):  
  return lambda a : a \* n  
  
mydoubler = myfunc(2)  
  
print(mydoubler(11))

Sonuç: 22 ‘dir

## Fatih Ercan – Lamda Anlatımı

*"""  
lambda  
"""*def square(num): #parametre(alacağı parametre)  
 return num \*\* 2  
print(square(4))  
  
print("----------------------------------")  
lambdaSquare = lambda num: num \*\* 2  
print(lambdaSquare(2))  
  
print("----------------------------------")  
  
totalLambda = lambda num1,num2,num3 : num1+num2+num3  
print(totalLambda(1,2,3))  
  
print("----------------------------------")  
  
def action(n):  
 return lambda a: a \*\* n  
  
multiLambda\_Square = action(2) # karsini aldığımız için parantez içinde 2 yazıyoruz.  
multiLambda\_Cube = action(3) # küpünü aldığımız için parantez içinde 3 yazıyoruz.  
print(type(multiLambda\_Square))  
print(type(multiLambda\_Cube))  
  
print("-----------------yada----------------------")  
print(multiLambda\_Square(2))  
print(multiLambda\_Cube(3))  
  
print("-----------------farklı bir lambda örneği--------------------------")  
myNumbers = [1,2,4,64,55,3]  
resaultList = list(map(lambda a: a \*\* 2, myNumbers))  
print(resaultList)  
  
print("--------------lambda operasyonlarında bir filtre yapma konusuna geçtik--------------------------")  
filterNumbers = [1,25,9,13,20,65,84]  
result = list(filter(lambda x: x % 2 == 0,filterNumbers))  
print(result)  
  
print("--------------yada--------------------------")  
def findEven(\*numbers): #findEven çift sayıları bul demek  
 result = []  
 for x in numbers:  
 if x % 2 == 0:  
 result.append(x)  
 return result  
print(result)  
  
print("--------------yada--------------------------")  
result2 = list(filter(lambda x: x \*\* 2 == 0,filterNumbers))  
print(result)  
  
print("--------------yada--------------------------")  
myStrList = {"Ali","Veli","asdad","asdasd"}  
result3 = list(filter(lambda x: x == "Ali", myStrList))  
print(result3)

# ARRAYS

# CLASSES/OBJECTS

# INHERITANCE

# ITERATORS

# SCOPE

# MODULES

# DATES

# MATH

# JSON

# REGEX

import re  
koordinat = "36.45, <asdasda25.66"  
  
re.sub(r"[^\d\,\.]","", koordinat)  
print(koordinat)  
  
# ^ bu işaret değil ise  
# \d herhangi bir digit demek  
  
# https://www.rexegg.com/regex-quickstart.html

# PIP

# TRY...EXCEPT

try bloğu, bir kod bloğunu hatalar için test etmenizi sağlar.

except blok, hatayı işlemenize izin verir.

else bloğu, hata olmadığında kodu çalıştırmanıza izin verir.

finally bloğu, deneme ve hariç tutma bloklarının sonucundan bağımsız olarak kod yürütmenize izin verir.

## Many Exceptions – Birçok Hata

Try bloğu bir NameError ve diğer hatalar için başka bir mesaj verirse bir mesaj yazdırın:

#x tanımlı olmadığı için try bloğu bir NameError oluşturacaktır:

try:  
  print(x)  
except NameError:  
  print("x değişkeni tanımlı değil")  
except:  
  print("Başka bir şey ters gitti")

## Else – Başka, Yoksa, Aksi Halde, Ayrıca

Herhangi bir hata oluşmadıysa yürütülecek bir kod bloğunu tanımlamak için else anahtar sözcüğünü kullanabilirsiniz:

try:  
  print("Hello")  
except:  
  print("Bir şeyler yanlış gitti")  
else:  
  print("hiçbir şey ters gitmedi")

## Finally

Final bloğu, belirtilirse, try bloğunun bir hata oluşturup oluşturmadığına bakılmaksızın yürütülür.

try:  
  print(x)  
except:  
  print("Bir şeyler yanlış gitti")  
finally:  
  print(" 'try except' bitti")

Örnek:

try:  
  f = open("demofile.txt")  
  try:  
    f.write("Lorum Ipsum")  
  except:  
    print("Dosyaya yazarken bir şeyler ters gitti")  
  finally:  
    f.close()  
except:  
  print("Dosyayı açarken bir şeyler ters gitti ")

## Raise an exception – istisna oluşturma

* Bir Python geliştiricisi olarak, bir koşul oluştuğunda bir istisna oluşturmayı seçebilirsiniz.
* Bir istisna atmak (veya yükseltmek) için yükseltme anahtar sözcüğünü kullanın.

x = -1  
  
if x < 0:  
  raise Exception("Üzgünüm, sıfırın altında sayı yok ")

x = "hello"  
  
if not type(x) is int:  
  raise TypeError("Yalnızca tam sayılara izin verilir")

## Fatih Ercan - Try Exeption

try:  
 sayi1 = input("Sayı Giriniz: ")  
 sayi2 = input("Sayi 2 giriniz: ")  
  
 sayi1 = int(sayi1)  
 sayi2 = int(sayi2)  
except:  
 print(f"Hata : Verdi ")  
  
# --------------------------------------------------------  
  
try:  
 sayi1 = input("Sayı Giriniz: ")  
 sayi2 = input("Sayi 2 giriniz: ")  
  
 sayi1 = int(sayi1)  
 sayi2 = int(sayi2)  
except Exception as ex:  
 print(f"Hata : Verdi {ex}")  
  
# --------------------------------------------------------  
  
try:  
 sayi1 = input("Sayı Giriniz: ")  
 sayi2 = input("Sayi 2 giriniz: ")  
  
 sayi1 = int(sayi1)  
 sayi2 = int(sayi2)  
  
 result = sayi1 / sayi2  
 print(result)  
  
except ValueError as valueE:  
 print(f"Hata : {valueE}")  
except ZeroDivisionError as zero:  
 print(f"Hata : {zero}")  
except:  
 print("Genel Hata...")  
  
  
print("Program Devam Ediyor....")

### Try – Exception \_ Örnek 2

*"""  
  
try:  
 kodlar.....  
except Excetion :* ***:except*** *kod...  
else  
 else kodlar.....  
  
"""*try:  
 number1 = int(input("1. Sayıyı Giriniz: "))  
 number2 = int(input("2. Sayıyı Giriniz: "))  
except ValueError:  
 print("Sadece Sayı Giriniz...")  
else:  
 try:  
 print(number1 / number2)  
 except ZeroDivisionError:  
 print("Sayı sıfıra bölünmez...")

### Try – exception \_ örnek 3

*"""  
  
try:  
 kod  
except  
 kod  
except2  
 kod  
finally:  
 kod  
  
"""*try:  
 int1 = int(input("İlk Sayı :"))  
 int2 = int(input("İkinci Sayı : "))  
 print(int1//int2)  
except:  
 print(" Error Alındı")  
finally:  
 print("Finally Blok çalıştı")

### Try – Exception \_ Örnek 4

*"""  
raise  
"""*list1 = []  
list2 = []  
  
value1 = input("İlk Veri : ")  
value2 = input("İkinci Veri : ")  
value3 = input("Üçüncü Değer :")  
value4 = input("Dördüncğ değer : ")  
  
list1.append(value1)  
list1.append(value2)  
  
list2.append(value3)  
list2.append(value4)  
  
try:  
 for x in list1:  
 if not x.isdigit():  
 raise Exception("İlk Dizi için sasdece sayısal değer girilmeli")  
 else:  
 print("1. Doğru...")  
  
 for x in list2:  
 if not x.isalpha():  
 raise ValueError("İkici Dizede sadece metinsel ifadeler olmalı")  
 else:  
 print("2. Doğru...")  
except Exception as ex:  
 print(f"Hata : {ex}")

### Try – exception \_ örnek 5

class VectorError(Exception):  
 pass  
  
  
list1 = (1, 5, 2)  
list2 = (2, 5, 4)  
  
if len(list1) != len(list2):  
 raise VectorError("Dizilerin uzunluğu eşit olmalı...")  
else:  
 result = sum(a \* b for (a, b) in zip(list1, list2))  
 print(result)  
  
print("Program Biti...")

### TE1\_Try\_Demo\_1

*"""  
Kullanıcıdan 5 adet sayı alarak ekrana yazdırın ve  
bunların ortalamasını tam sayı şeklinde ekrana yazdırın  
"""*numbers = []  
  
for x in range(1, 6):  
 try:  
 number = input(f"{x}. Sayıyı Giriniz:")  
 number = int(number)  
 numbers.append(number)  
 except:  
 print("Hatalı Bir Giriş Yaptınız...")  
 continue  
  
print(numbers)  
  
  
total = 0  
  
for x in numbers: total += x # total = total + x  
  
print(f"Ortalama {total / len(numbers)}")

### TE1\_Try\_Demo\_2

*"""  
Kullanıcıdan 5 adet sayı alarak ekrana yazdırın ve  
bunların ortalamasını tam sayı şeklinde ekrana yazdırın  
"""*numbers = []  
for x in range(1, 6):  
 while True:  
 try:  
 number = input(f"{x}. Sayıyı Giriniz:")  
 number = int(number)  
 numbers.append(number)  
 break  
 except:  
 print("Hatalı Bir Giriş Yaptınız...")  
 continue  
  
print(numbers)  
total = 0  
for x in numbers: total += x # total = total + x  
print(f"Ortalama {total // len(numbers)}")

### TE1\_Try\_Demo\_3

*"""  
  
Kullanıcıdan alınan 2 değeri bölen bir uygulama geliştiriniz.  
her hataya karşı farklı bir uyarı ekrana yazdırsın  
2 değeri alana kadar tekrar etsin  
  
(Value , Zero)  
  
"""*# try:  
# number1 = int(input("İlk Sayıyı Giriniz : "))  
# number2 = int(input("İkinci Sayıyı Giriniz : "))  
# print(number1 // number2)  
# except ValueError:  
# print("Lütfen Sadece Sayı Giriniz...")  
# except ZeroDivisionError:  
# print("Bölen Sayı 0 olamaz...")  
# except:  
# print("Genel Hata")  
# ---------------------------------------------------------  
  
numbersA = []  
  
for x in range(1, 3):  
 while True:  
 try:  
 numberA = int(input(f"lütfen {x}. Sayıyı Giriniz..."))  
 numbersA.append(numberA)  
 except ValueError:  
 print("Lütfen Sadece Sayı Giirniz...")  
 continue  
 break  
  
try:  
 result = numbersA[0] // numbersA[1]  
 print(result)  
except ZeroDivisionError as Zero:  
 print("Sıfıra Bölünemez..")

### TE2\_Try\_Demo\_1

number1 = 0  
number2 = 0  
  
while True:  
 try:  
 number1 = int(input("1. Sayı Giriniz: "))  
 except ValueError:  
 print("Sadece Sayı Giriniz...")  
 continue  
 else:  
 try:  
 number2 = int(input("2. Sayı Girniz: "))  
 print(f" Sonuç : {number1 // number2}")  
 break  
 except ValueError:  
 print("Sadece Sayı Giriniz...")  
 except ZeroDivisionError:  
 print("Error Sayı 0 a Bölünmez.....")  
  
print("Program Bitti....")

### TE4\_Try\_Demo\_2

*"""  
Kullanıcdan 5 tane değer girdireceğiz  
Bu değerler arasında sayısal veri olmayacak hata verdireceğiz.  
Sonunda da her halükarda ekrana listeyi yazdıracağız.  
"""*values = []  
  
for x in range(0, 5):  
 value = input("Lütfen Değer Giriniz: ")  
 values.append(value)  
  
try:  
 for item in values:  
 if item.isdigit():  
 raise ValueError("Sayısal Bir Değer Girilemez.")  
except Exception as ex:  
 print(f"Hata : {ex}")  
finally:  
 print(values)  
 # del values

### Tekrar

#### Try – Exception – Hata Fırlatma

Stage\_Hocanın yaptığı

#### Soru = Cevap Erdem’in yaptığı çözüm

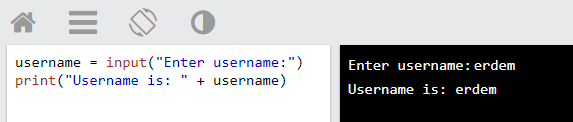
*"""  
  
  
1. 3 adet öğrenci bilgisi alınız.  
 1.1. Boş değer girilemez.  
2. sırası ile öğrenci notlarını sorunu (0-100)  
 2.1. Hatlı girişlerde hata fırlatarak ekrana tekrar sorulacak.  
3. Girilen öğrenci notlarına göre derslerin durumunu ekrana yazdırınız.  
 3.1. 0-39:1 , 40-49:2 , 50-49:3 , 70-84:4 , 85-100:5  
4.  
"""*students ={}  
  
# while True:  
for x in range(1,3):  
 while True:  
 try:  
 cOgrno = int(input(f"{x}. Öğrencinin Nosunu Giriniz: "))  
 if not cOgrno:  
 raise ValueError("Öğrenci Nosunu Boş Giremezsiniz")  
 except Exception as cOgrnoEx:  
 print(f"{x}. Öğrenci Nosunu Tekrar Giriniz. Çünkü İstenilen Değer Aralığında Bir Giriş Yapmadınız.")  
  
 try:  
 cName = str(input(f"{x}. Öğrencinin Adını Giriniz: "))  
 if not cName:  
 raise ValueError("Öğrenci Adını Boş Giremezsiniz")  
 except Exception as cNameEx:  
 print(f"{x}. Öğrenci Adını Tekrar Giriniz. Çünkü İstenilen Değer Aralığında Bir Giriş Yapmadınız.")  
  
 try:  
 cclassName = str(input(f"{x}. Öğrencinin Sınıfını Giriniz: "))  
 if not cclassName:  
 raise ValueError("Öğrenci Sınıfını Boş Giremezsiniz")  
 except Exception as cclassNameEx:  
 print(f"{x}. Öğrenci Sınıfını Tekrar Giriniz. Çünkü İstenilen Değer Aralığında Bir Giriş Yapmadınız.")  
  
 try:  
 cmath = int(input(f"{x}. Öğrencinin Matematik Notunu Giriniz: "))  
 if not cmath:  
 raise ValueError("Öğrenci Matematik Notunu Boş Giremezsiniz")  
 except Exception as cmathEx:  
 print(f"{x}. Öğrenci Matematik Notunu Tekrar Giriniz. Çünkü İstenilen Değer Aralığında Bir Giriş Yapmadınız.")  
  
 try:  
 cbio = int(input(f"{x}. Öğrencinin Biyoloji Notunu Giriniz: "))  
 if not cbio:  
 raise ValueError("Öğrenci Biyoloji Notunu Boş Giremezsiniz")  
 except Exception as cbioEx:  
 print(f"{x}. Öğrenci Matematik Notunu Tekrar Giriniz. Çünkü İstenilen Değer Aralığında Bir Giriş Yapmadınız.")  
  
 try:  
 cphsysic = int(input(f"{x}. Öğrencinin Fizik Notunu Giriniz: "))  
 if not cphsysic:  
 raise ValueError("Öğrenci Fizik Notunu Boş Giremezsiniz")  
 except Exception as cphsysicEX:  
 print(f"{x}. Öğrenincinin Fizik Notunu Tekrar Girniz. Çünkü İstenilen Değer Aralığında Bir Giriş Yapmadınız.")  
  
  
 students[cOgrno] = {  
 "Ogrno": {  
 "name": cName,  
 "className": cclassName,  
 "notes": {  
 "math": cmath,  
 "bio": cbio,  
 "phsysic": cphsysic  
 }  
 }  
 }  
 break  
 students[x]["name"]  
print(students)  
print(f"--------------------------------------------")

#### GeneralDemo\_1

*"""  
  
  
1. 3 adet öğrenci bilfisi alınız.  
 1.1. Boş değer girilemez.  
2. sırası ile öğrenci notlarını sorunu (0-100)  
 2.1. Hatlı girişlerde hata fırlatarak ekrana tekrar sorulacak.  
3. Girilen öğrenci notlarına göre derslerin durumunu ekrana yazdırınız.  
 3.1. 0-39:1 , 40-49:2 , 50-49:3 , 70-84:4 , 85-100:5  
4.  
"""*students ={}  
  
# while True:  
for x in range(1,3):  
 while True:  
 try:  
 cOgrno = int(input(f"{x}. Öğrencinin Nosunu Giriniz: "))  
 if not cOgrno:  
 raise ValueError("Öğrenci Nosunu Boş Giremezsiniz")  
 except Exception as cOgrnoEx:  
 print(f"{x}. Öğrenci Nosunu Tekrar Giriniz. Çünkü İstenilen Değer Aralığında Bir Giriş Yapmadınız.")  
  
 try:  
 cName = str(input(f"{x}. Öğrencinin Adını Giriniz: "))  
 if not cName:  
 raise ValueError("Öğrenci Adını Boş Giremezsiniz")  
 except Exception as cNameEx:  
 print(f"{x}. Öğrenci Adını Tekrar Giriniz. Çünkü İstenilen Değer Aralığında Bir Giriş Yapmadınız.")  
  
 try:  
 cclassName = str(input(f"{x}. Öğrencinin Sınıfını Giriniz: "))  
 if not cclassName:  
 raise ValueError("Öğrenci Sınıfını Boş Giremezsiniz")  
 except Exception as cclassNameEx:  
 print(f"{x}. Öğrenci Sınıfını Tekrar Giriniz. Çünkü İstenilen Değer Aralığında Bir Giriş Yapmadınız.")  
  
 try:  
 cmath = int(input(f"{x}. Öğrencinin Matematik Notunu Giriniz: "))  
 if not cmath:  
 raise ValueError("Öğrenci Matematik Notunu Boş Giremezsiniz")  
 except Exception as cmathEx:  
 print(f"{x}. Öğrenci Matematik Notunu Tekrar Giriniz. Çünkü İstenilen Değer Aralığında Bir Giriş Yapmadınız.")  
  
 try:  
 cbio = int(input(f"{x}. Öğrencinin Biyoloji Notunu Giriniz: "))  
 if not cbio:  
 raise ValueError("Öğrenci Biyoloji Notunu Boş Giremezsiniz")  
 except Exception as cbioEx:  
 print(f"{x}. Öğrenci Matematik Notunu Tekrar Giriniz. Çünkü İstenilen Değer Aralığında Bir Giriş Yapmadınız.")  
  
 try:  
 cphsysic = int(input(f"{x}. Öğrencinin Fizik Notunu Giriniz: "))  
 if not cphsysic:  
 raise ValueError("Öğrenci Fizik Notunu Boş Giremezsiniz")  
 except Exception as cphsysicEX:  
 print(f"{x}. Öğrenincinin Fizik Notunu Tekrar Girniz. Çünkü İstenilen Değer Aralığında Bir Giriş Yapmadınız.")  
  
  
 students[cOgrno] = {  
 "Ogrno": {  
 "name": cName,  
 "className": cclassName,  
 "notes": {  
 "math": cmath,  
 "bio": cbio,  
 "phsysic": cphsysic  
 }  
 }  
 }  
 break  
 students[x]["name"]  
print(students)  
print(f"--------------------------------------------")

# USER INPUT

username = input("Enter username:")  
print("Username is: " + username)



# STRING FORMATTİNG

* format() yöntemi, bir dizenin seçili bölümlerini biçimlendirmenize olanak tanır.
* Bazen bir metnin kontrol etmediğiniz kısımları olabilir, bunlar bir veritabanından veya kullanıcı girdisinden gelebilir mi?
* Bu tür değerleri kontrol etmek için metne yer tutucular (kıvrımlı parantezler {}) ekleyin ve değerleri format() yöntemiyle çalıştırın:

Örnek:

fiyatı görüntülemek istediğiniz yere bir yer tutucu ekleyin:

price = 49  
txt = "The price is {} dollars"  
print(txt.format(price))



Örnek:

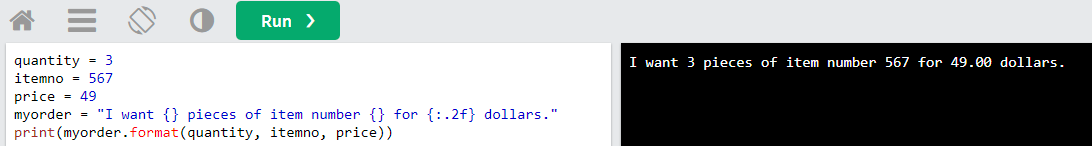
price = 49

txt = "The price is {:.2f} dollars"

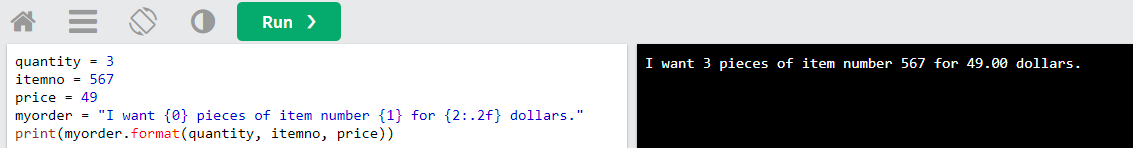
print(txt.format(price))



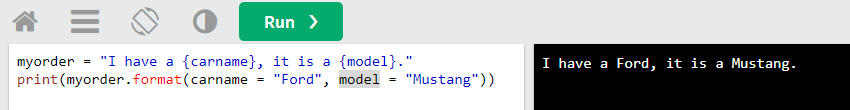
Çoklu Değer Getirme

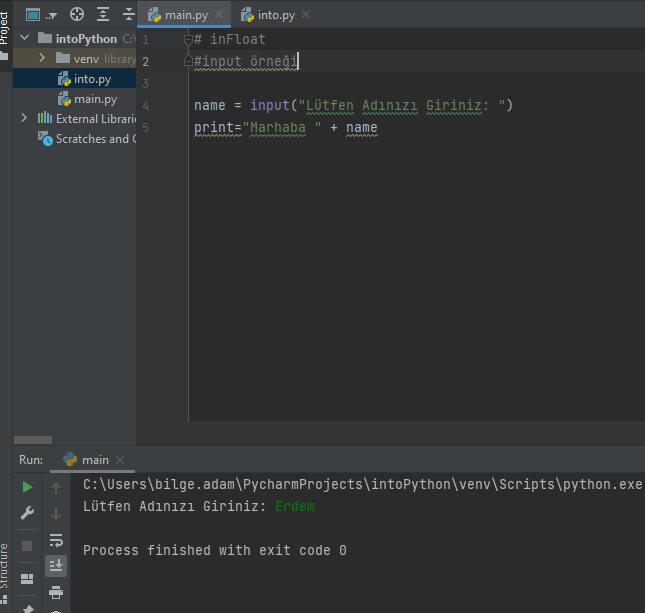


Index Numarasına Göre Getirme

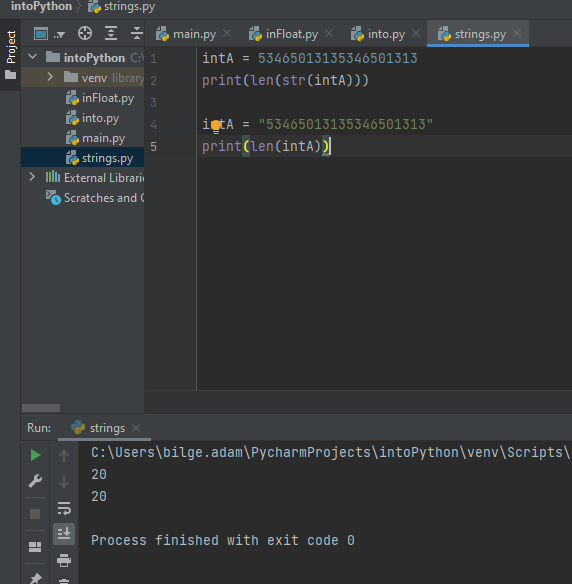


Named Indexes – İç içe indeks





#### Karakter saydırma

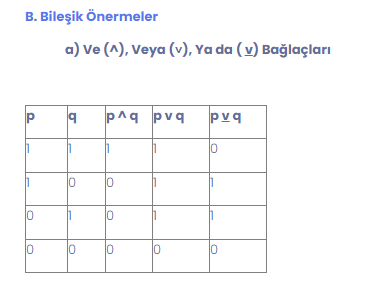


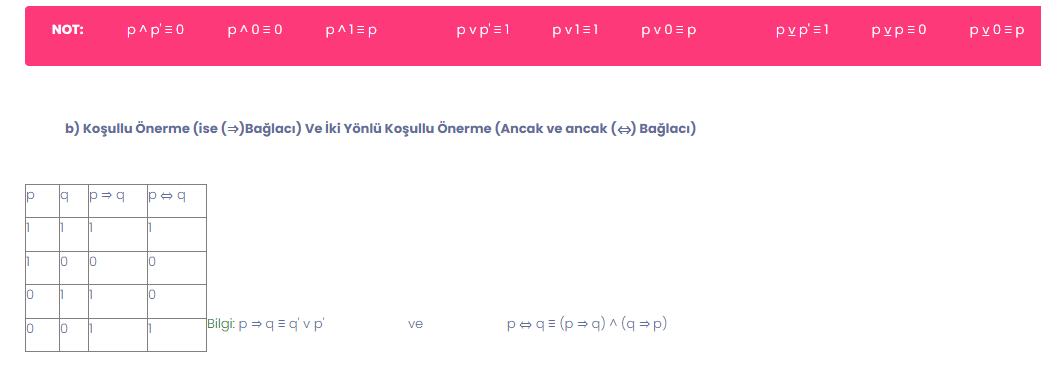
### Dersteki ödev

*"""  
1- "Merhaba" yazısını tersten yazdır  
2- "Merhaba TKGM Hoş Geldiniz" yazısındaki "Hoş" yazısını ekrana yazdırınız  
3- "Merhaba TKGM Hoş Geldiniz" hepsini büyük yapınız  
4- "Merhaba TKGM Hoş Geldiniz" boşluk itibari ile diziye alın.  
5- "Merhaba TKGM Hoş Geldiniz" i harflerini ı ile değiştir.  
6- "Merhaba TKGM Hoş Geldiniz" hepsini küçük harfe çevir ve sadece ilk harfini büyük yap  
7- "Merhaba TKGM Hoş Geldiniz" 2 şe 2 şer atlayarak yazdır  
8- "Merhaba TKGM Hoş Geldiniz" kaç adet "a" harfinin olduğunu yazınız.  
"""*#1 "Merhaba" yazısını tersten yazdır  
Atama = "Merhaba"  
result1 = Atama[::-1]  
print(result1)  
  
#2 "Merhaba TKGM Hoş Geldiniz" yazısındaki "Hoş" yazısını ekrana yazdırınız  
Atama = "Merhaba TKGM Hoş Geldiniz"  
result2 = Atama[13:16]  
print(result2)  
  
#3 "Merhaba TKGM Hoş Geldiniz" hepsini büyük yapınız  
Atama = "Merhaba TKGM Hoş Geldiniz"  
result3 = Atama.upper()  
print(result3)  
  
#4 "Merhaba TKGM Hoş Geldiniz" boşluk itibari ile diziye alın.  
Atama = "Merhaba TKGM Hoş Geldiniz"  
result4 = Atama.split()  
print(result4)  
  
#5 "Merhaba TKGM Hoş Geldiniz" i harflerini ı ile değiştir.  
Atama = "Merhaba TKGM Hoş Geldiniz"  
result5 = Atama.replace("i","ı")  
print(result5)  
  
#6 "Merhaba TKGM Hoş Geldiniz" hepsini küçük harfe çevir ve sadece ilk harfini büyük yap  
Atama = "Merhaba TKGM Hoş Geldiniz"  
result6 = Atama.capitalize()  
print(result6)  
  
#7 "Merhaba TKGM Hoş Geldiniz" 2 şe 2 şer atlayarak yazdır  
Atama = "Merhaba TKGM Hoş Geldiniz"  
result7 = Atama[::2]  
print(result7)  
  
#8 "Merhaba TKGM Hoş Geldiniz" kaç adet "a" harfinin olduğunu yazınız.  
Atama = "Merhaba TKGM Hoş Geldiniz"  
result8 = Atama.count("a")  
print(result8)

### Mantık Operatörleri

*"""  
== eşit  
!= Eşit değildir  
<  
>  
=<  
>=  
  
in (içinde)  
and (ve)  
or (veya)  
not  
"""*var1 = 5 == 7  
var2 = 5 != 7  
print(var1)  
print(type(var2))  
print("a" in "Merhaba") # içinde a var mı ? varsa true döner yoksa false döner  
print("a" in "Merhaba" and 3>2) # ve üç ikiden büyük mü? true döner  
print("a" in "Merhaba" and 3>2 and var1)  
print("a" in "Merhaba" and 3>2 or var1)





### SORU: 5 karakterli bir kelime girdirin, baştan ikinci ile sondan ikinci karakterin eşit olup olmadığını kıyasla

var3 = input("5 karakterli kelime giriniz: ")  
var4 = var3[1] == var3[3]  
print(var4)

Sonuc : False gelir

### Atama operatörlerine örnek

var6 = 5  
var6 += 10  
print(var6)

# EĞİTİM VİDEOLARI

4.Gün 1. Videosu

<https://bulut.tkgm.gov.tr/s/Y3AxKXyogKG8a5E>

4.gün 2.Videosu

<https://bulut.tkgm.gov.tr/s/YxWe9YQJnYiYGww>

5.gün 1.Video \_Sadece Ödevin Satage 4’ü

<https://bulut.tkgm.gov.tr/s/ctgApdcnA9EF3F4>

5.gün 2. Videosu \_ Fonksiyonlar

<https://bulut.tkgm.gov.tr/s/3GoGZKT626TYXQi>

5.gün 3.Videosu \_ Asal Sayı Örneği

<https://bulut.tkgm.gov.tr/s/DHMpXDJWp4boHca>

6.gün 1.Video\_konu tekrarı

<https://bulut.tkgm.gov.tr/s/y4G46Aj7fdH564n>

6.gün 2.Video\_konu tekrarı\_matematik operatörleri

<https://bulut.tkgm.gov.tr/s/eKGmZrMYS4SrpRT>

6.gün 3.Video\_konu tekrarı\_array\_diziler

<https://bulut.tkgm.gov.tr/s/kryTdtsQBamCMBT>

6.gün 4.Video\_konu tekrarı\_tuple ve set

<https://bulut.tkgm.gov.tr/s/k5eXdQ6gLcWTQ7z>

6.gün 5.Video\_konu tekrarı\_brand\_marka\_örneği

<https://bulut.tkgm.gov.tr/s/GNsXHeGQSNPjnSM>

6.gün 6.Video\_konu tekrarı\_operatörler

<https://bulut.tkgm.gov.tr/s/iQJ5F2c268MoDxK>

# 7.GÜN – 28.10.2022 – CUMA \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Online eğitim

# 8.GÜN – 29.10.2022 – CUMARTESİ \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Online eğitim

# 9.GÜN – 30.10.2022 – PAZAR \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Online eğitim

# 10.GÜN – 04.11.2022 – CUMA \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

## Koordinat Oku

import re  
p10 = "10.54,34.59"  
p11 = "10.54,34.59"  
p12 = "10.54a , 34.59"  
p13 = "10.5x4 , 34.59"  
p14 = "10.54\n, 34.59"  
  
  
def koordinatOku(metin:str) ->list:  
 metin=[x for x in metin if x.isdigit() or x==',' or x=='.']  
 metin="".join(metin)  
  
 metin=metin.replace(" ", "")  
 metin=metin.replace('\n' , "")  
 parcalanmis=metin.split(",")  
 parcalanmis=[float(parca) for parca in parcalanmis]  
 return parcalanmis  
  
print("p10", koordinatOku(p10))  
print("p11", koordinatOku(p11))  
print("p12", koordinatOku(p12))  
print("p13", koordinatOku(p13))  
print("p14", koordinatOku(p14))  
  
  
print(re.search("[A-Za-z]", ))

## Open Street Map – OSMPythonTools

from OSMPythonTools.api import Api  
a=Api()  
  
sonuc=a.query("way/421469056")  
print(sonuc)  
gelenGeom=sonuc.geometry()  
  
for c in gelenGeom:  
 print(c, gelenGeom[c])  
 print(sonuc.tags())  
  
  
  
print("------------------------------")  
# s= ["print(sonuc." + a + "())" for a in dir(sonuc) if a [0] != "\_"]  
s= ["print('" + a + "', sonuc." + a + "())" for a in dir(sonuc) if a [0] != "\_"]  
print("\n".join(s))  
  
  
print('apiVersion', sonuc.apiVersion())  
print('areaId', sonuc.areaId())  
print('attribution', sonuc.attribution())  
print('cacheTimestamp', sonuc.cacheTimestamp())  
print('cacheVersion', sonuc.cacheVersion())  
print('centerLat', sonuc.centerLat())  
print('centerLon', sonuc.centerLon())  
print('changeset', sonuc.changeset())  
print('copyright', sonuc.copyright())  
print('countMembers', sonuc.countMembers())  
print('countNodes', sonuc.countNodes())  
print('fromId', sonuc.fromId())  
print('generator', sonuc.generator())  
print('geometry', sonuc.geometry())  
print('history', sonuc.history())  
print('id', sonuc.id())  
print('isValid', sonuc.isValid())  
print('lat', sonuc.lat())  
print('license', sonuc.license())  
print('lon', sonuc.lon())  
print('members', sonuc.members())  
print('nodes', sonuc.nodes())  
print('queryString', sonuc.queryString())  
print('tag', sonuc.tag())  
print('tags', sonuc.tags())  
print('timestamp', sonuc.timestamp())  
print('toXML', sonuc.toXML())  
print('type', sonuc.type())  
print('typeId', sonuc.typeId())  
print('typeIdShort', sonuc.typeIdShort())  
print('uid', sonuc.uid())  
print('user', sonuc.user())  
print('userid', sonuc.userid())  
print('version', sonuc.version())  
print('visible', sonuc.visible())

### method

from OSMPythonTools.api import Api  
a=Api()  
  
sonuc=a.query("way/421469056")  
  
  
info = {}  
for a in dir(sonuc):  
 if a[0] != "\_" and a not in ["fromId", "tag"]:  
 method = getattr(sonuc, a)  
 info[a] = method()  
  
print((info))

### farklı kullanımı

from OSMPythonTools.api import Api  
ap=Api()  
  
sonuc=ap.query("way/421469056")  
  
  
def bilgiSorgula(id):  
 sonuc = ap.query(id)  
 info = {}  
 for a in dir(sonuc):  
 if a[0] != "\_" and a not in ["fromId", "tag"]:  
 method = getattr(sonuc, a)  
 info[a] = method()  
  
 return info  
  
print(bilgiSorgula("way/473223865"))

## Open Street Map verisini CSV’ye yazdırma

from OSMPythonTools.api import Api  
  
ap=Api()  
  
sonuc=ap.query("way/421469056")  
  
idler = [473223865,144231683,168719557]  
  
def bilgiSorgula(id):  
 sonuc = ap.query(id)  
 info = {}  
 for a in dir(sonuc):  
 if a[0] != "\_" and a not in ["fromId", "tag","toXML"]:  
 method = getattr(sonuc, a)  
 info[a] = method()  
  
 return info  
  
  
  
print("-------------------------")  
  
import pandas as pd  
sample = bilgiSorgula("way/473223865")  
df = pd.DataFrame(columns= sample.keys())  
print(df)  
  
for id in idler:  
 df.loc[len(df)] = bilgiSorgula(f"way/{id}").values()  
 # print(bilgiSorgula(f"way/{id}"))  
  
print("--------------------------")  
print(df)  
  
df.to\_csv("erdem.csv")

# 11.GÜN – 05.11.2022 – CUMARTESİ \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

print("-------------OSMPythonTools - Nominatim ------------------")  
print("------------- 05.11.2022 - Cumartesi ------------------")  
from OSMPythonTools.nominatim import Nominatim  
nominatim = Nominatim()  
location = nominatim.query("Cumhuriyet, Üsküdar", wkt=True)  
print("NAME", location.displayName())  
print("ARID", location.areaId())  
print("ADRS", location.address())  
print("TYPE", location.type())  
print("WKT", location.wkt())  
print("JSON", location.toJSON()[0])  
  
  
  
print("-------------wktPoligonOkuuyucu ------------------")  
def wktPoligonOkuuyucu( metin : str) -> list:  
 metin = metin.replace("POLYGON((","")  
 metin = metin.replace("))","")  
 koordinatlar = metin.split(",")  
  
 koordinatlar = [tuple([float(c) for c in k.split(" ")]) for k in koordinatlar]  
 # üst satırdakini teker teker yapma stili  
 # koordinatlar = [k.split(" ") for k in koordinatlar]  
 # koordinatlar = [[float(k) for k in koordinatlar] for k in koordinatlar]  
 # koordinatlar = [tuple(k) for k in koordinatlar]  
 return koordinatlar  
  
poligon\_metin = "POLYGON((29.0726392 41.0056665,29.0738694 41.005781,29.07535 41.005969,29.076774 41.0063719,29.0775729 41.0066838,29.0785461 41.0070638,29.0814741 41.0082072,29.08346 41.0088827,29.0851591 41.0092204,29.0870841 41.0094598,29.0888092 41.0094893,29.0890691 41.0094836,29.0894421 41.0094753,29.0893821 41.0100059,29.0893227 41.0102341,29.0892849 41.0103792,29.0892807 41.0104834,29.0892748 41.0106322,29.0892579 41.0109075,29.0893658 41.0110954,29.0893737 41.0112263,29.0893546 41.0113694,29.0892576 41.0115952,29.0891316 41.0119404,29.0886524 41.0119708,29.0881341 41.0119357,29.0876894 41.0136003,29.087473 41.0140839,29.0873369 41.0143716,29.0870082 41.0150678,29.085803 41.0147199,29.0852201 41.0145594,29.0834767 41.0141223,29.0820926 41.0139442,29.0815401 41.0140332,29.081068 41.0144946,29.0808213 41.0147092,29.0798235 41.0150006,29.0791154 41.015203,29.0784019 41.014863,29.0772378 41.0143854,29.0775865 41.0137499,29.076701 41.0137941,29.0753174 41.0138632,29.0750211 41.0116305,29.0748212 41.0111856,29.0746415 41.0108253,29.074569 41.0103517,29.0745181 41.0100906,29.0744081 41.0096089,29.0742579 41.0092324,29.0740567 41.0087082,29.0736946 41.0080787,29.0734237 41.0076294,29.0730777 41.0069615,29.0729275 41.0066194,29.0728497 41.0063036,29.0727049 41.0058604,29.0726392 41.0056665))"  
print(wktPoligonOkuuyucu(poligon\_metin))

## Nominatim

#: Imports  
import sys  
from OSMPythonTools.overpass import overpassQueryBuilder  
from OSMPythonTools.api import Api  
from OSMPythonTools.nominatim import Nominatim  
from OSMPythonTools.overpass import Overpass  
  
overpass = Overpass()  
nominatim = Nominatim()  
api = Api()  
  
def özellikÇıkar( nesne: object, hariç: list = [] ) -> dict:  
 info = {}  
 for a in dir( nesne ):  
 if a[0] != "\_" and a not in hariç:  
 method = getattr( nesne, a )  
 info[a] = method()  
 return info  
  
def sorgula( girdi, ekstra = 10 ):  
 if type(girdi) is int:  
 sonuc = api.query(f"way/{girdi}")  
 return özellikÇıkar( sonuc, ["fromId", "tag", "toXML"] )  
 if type(girdi) is str:  
 sonuc = nominatim.query(girdi, wkt=True)  
 return özellikÇıkar( sonuc )  
 if type(girdi) is tuple:  
 if type(girdi[0]) is float:  
 sonuc = nominatim.query(girdi[0], girdi[1], reverse=True, zoom= ekstra)  
 return özellikÇıkar( sonuc )  
 if type(girdi[0]) is str:  
 loc = nominatim.query( girdi[0] )  
 query = overpassQueryBuilder(area=loc, elementType='node', selector=girdi[1], out='body')  
 sonuc = overpass.query(query)  
 #!!!! problem  
 return özellikÇıkar( sonuc, ["timestamp\_area\_base"] )  
 raise Exception("Bilinmeyen format")  
  
  
print("1", sorgula(473223865) )  
print("2", sorgula("Cumhuriyet, Üsküdar") )  
print("3", sorgula(( 49.4093582, 8.694724 )) )  
print("4", sorgula(( "Cumhuriyet, Üsküdar", '"highway"="bus\_stop"' )) )

s1 = sorgula(473223865)  
s2 = sorgula("Cumhuriyet, Üsküdar")  
s3 = sorgula(( 49.4093582, 8.694724 ))  
s4 = sorgula(( "Cumhuriyet, Üsküdar", '"highway"="bus\_stop"' ))  
  
s1 = list(s1.keys())  
s2 = list(s2.keys())  
s3 = list(s3.keys())  
s4 = list(s4.keys())  
  
  
birleştirilmiş = s1 + s2 + s3 +s4  
birleştirilmiş = sorted(birleştirilmiş)  
print("birleştirilmiş hali :",birleştirilmiş)  
yegane = set(birleştirilmiş)  
  
print(len(yegane))  
  
# kaç tane keys den olduğunu bulmak için  
import collections  
# using Counter to find frequency of elements  
frequency = dict(collections.Counter(birleştirilmiş))  
# printing the frequency  
print(frequency)  
  
# bunlardan 4 tane geçenleri getir  
for i in frequency:  
 if frequency[i] == 4:  
 print(f"4 olanlar : {frequency.keys()}")  
 else:  
 pass  
 # print(f"İçin de 4 olan yok: {frequency.keys()} ")

## ???????

#: Kütüphaneleri import etme  
import sys  
from OSMPythonTools.overpass import overpassQueryBuilder  
from OSMPythonTools.api import Api  
from OSMPythonTools.nominatim import Nominatim  
from OSMPythonTools.overpass import Overpass  
  
#: Api'leri başlatma  
overpass = Overpass()  
nominatim = Nominatim()  
api = Api()  
  
#  
def özellikÇıkar( nesne: object, hariç: list = [] ) -> dict:  
 info = {}  
 for a in dir( nesne ):  
 if a[0] != "\_" and a not in hariç:  
 method = getattr( nesne, a )  
 info[a] = method()  
 return info  
  
def sorgula( girdi, ekstra = 10 ):  
 if type(girdi) is int:  
 sonuc = api.query(f"way/{girdi}")  
 return özellikÇıkar( sonuc, ["fromId", "tag", "toXML"] )  
 if type(girdi) is str:  
 sonuc = nominatim.query(girdi, wkt=True)  
 return özellikÇıkar( sonuc )  
 if type(girdi) is tuple:  
 if type(girdi[0]) is float:  
 sonuc = nominatim.query(girdi[0], girdi[1], reverse=True, zoom= ekstra)  
 return özellikÇıkar( sonuc )  
 if type(girdi[0]) is str:  
 loc = nominatim.query( girdi[0] )  
 query = overpassQueryBuilder(area=loc, elementType='node', selector=girdi[1], out='body')  
 sonuc = overpass.query(query)  
 return özellikÇıkar( sonuc, ["timestamp\_area\_base"] )  
 raise Exception("Bilinmeyen format")  
  
print( "1", sorgula(473223865) )  
print( "2", sorgula("Cumhuriyet, Üsküdar") )  
print( "3", sorgula(( 49.4093582, 8.694724 )) )  
print( "4", sorgula(( "Cumhuriyet, Üsküdar", '"highway"="bus\_stop"' )) )  
  
  
s1 = sorgula(473223865)  
s2 = sorgula("Cumhuriyet, Üsküdar")  
s3 = sorgula(( 49.4093582, 8.694724 ))  
s4 = sorgula(( "Cumhuriyet, Üsküdar", '"highway"="bus\_stop"' ))  
  
s1 = list(s1.keys())  
s2 = list(s2.keys())  
s3 = list(s3.keys())  
s4 = list(s4.keys())  
  
birleşmiş = s1 + s2 + s3 + s4  
birleşmiş = sorted(birleşmiş)  
yegane = set(birleşmiş)  
  
print(len(yegane))  
  
import collections  
frequency = dict(collections.Counter(birleşmiş))  
print(frequency)  
  
for item in frequency:  
 if frequency[item] == 2:  
 print(item)  
  
  
import pandas as pd  
df = pd.DataFrame(columns = list(yegane))  
  
  
df.loc[len(df)] = {s:1 for s in s1}  
df.loc[len(df)] = {s:1 for s in s2}  
df.loc[len(df)] = {s:1 for s in s3}  
df.loc[len(df)] = {s:1 for s in s4}  
print(df)  
  
  
# df.to\_csv("şablon.csv", sep=";")  
  
  
# polygon içindeki...  
# iki nokta arasında  
  
# kaydet  
# oku  
# networkx  
# dönüştür (xml, json, polygon, KML, GML, SHP, OSM)  
# parse (polygon, json, dict ....)  
  
  
  
  
  
  
temelSorgu = pd.DataFrame(columns =  
 ["temelSorguId", "cacheTimestamp", "cacheVersion", "queryString", "other"])  
alan = pd.DataFrame(columns =  
 ["alanId", "temelSorguId", "id", "areaId", "typeIdShort", "type", "typeId"])  
adres = pd.DataFrame(columns =  
 ["adresId", "alanId", "address", "displayName", "isReverse", "wkt"])  
node = pd.DataFrame(columns =  
 ["nodeId", "alanId", "countNodes", "generator", "nodes", "version",  
 "valid", "copyright"])  
  
# Auto Identity = Auto Increment  
son\_idler = { "temelSorgu": 0, "alan": 0, "adres": 0,"node": 0}  
  
def idver( tablo: str ) -> int:  
 sonid = son\_idler[ tablo ] + 1  
 son\_idler[tablo] = sonid  
 return sonid  
  
sorgular = [  
 473223865,  
 "Cumhuriyet, Üsküdar",  
 ( 49.4093582, 8.694724 ),  
 ( "Cumhuriyet, Üsküdar", '"highway"="bus\_stop"' )  
]  
  
  
#: Her bir sorgu için  
for sorgu in sorgular:  
 #: Sorgulama işlemi yap  
 sonuc = sorgula( sorgu )  
 #: Her bir sorguda, mutlaka temel sorgu vardır!  
 #: Temel sorgu için yeni bir id üret!  
 temelSorguId = idver("temelSorgu")  
 yeni\_kayıt = {  
 "temelSorguId": temelSorguId,  
 "cacheTimestamp": sonuc["cacheTimestamp"],  
 "cacheVersion": sonuc["cacheVersion"],  
 "queryString": sonuc["queryString"]  
 }  
 #: Temel sorgu tablosuna ekle  
 temelSorgu.loc[ len(temelSorgu) ] = yeni\_kayıt  
 #: Alan tablosuna ekleme yap  
 alanId = idver("alan")  
 yeni\_kayıt2 = {}  
 #: Temel bağlantı kolonlarını dolduralım  
 yeni\_kayıt2["alanId"] = alanId  
 yeni\_kayıt2["temelSorguId"] = temelSorguId  
  
 # ["id", "areaId", "typeIdShort", "type", "typeId"]  
  
  
  
  
 # print("!", sonuc)  
 # ["alanId", "temelSorguId", "id", "areaId", "typeIdShort", "type", "typeId"]  
  
"""  
! {'apiVersion': '0.6', 'areaId': 2873223865, 'attribution': 'http://www.openstreetmap.org/copyright', 'cacheTimestamp': '2022-11-04T15:23:49.505200', 'cacheVersion': '1.0', 'centerLat': None, 'centerLon': None, 'changeset': '114040647', 'copyright': 'OpenStreetMap and contributors', 'countMembers': 0, 'countNodes': 23, 'generator': 'CGImap 0.8.8 (4108215 spike-07.openstreetmap.org)', 'geometry': {"coordinates": [[[32.83615, 39.850298], [32.836171, 39.849481], [32.836307, 39.849416], [32.836359, 39.84935], [32.836362, 39.849255], [32.83639, 39.848428], [32.836553, 39.848366], [32.836633, 39.848319], [32.836713, 39.848262], [32.836757, 39.848224], [32.83679, 39.848171], [32.836818, 39.848119], [32.837018, 39.848244], [32.837211, 39.848487], [32.837306, 39.848487], [32.837307, 39.848535], [32.837321, 39.84859], [32.83759, 39.849091], [32.837274, 39.849246], [32.837366, 39.849785], [32.837458, 39.85013], [32.83698, 39.850331], [32.83615, 39.850298]]], "type": "Polygon"}, 'history': None, 'id': 473223865, 'isValid': True, 'lat': None, 'license': 'http://opendatacommons.org/licenses/odbl/1-0/', 'lon': None, 'members': [], 'nodes': [<OSMPythonTools.api.ApiResult object at 0x0000015B5DDE6B60>, <OSMPythonTools.api.ApiResult object at 0x0000015B5DDE6E30>, <OSMPythonTools.api.ApiResult object at 0x0000015B5DDE7040>, <OSMPythonTools.api.ApiResult object at 0x0000015B5DDE7250>, <OSMPythonTools.api.ApiResult object at 0x0000015B5DDE7460>, <OSMPythonTools.api.ApiResult object at 0x0000015B5DDE7670>, <OSMPythonTools.api.ApiResult object at 0x0000015B5DDE7880>, <OSMPythonTools.api.ApiResult object at 0x0000015B5DDE7A90>, <OSMPythonTools.api.ApiResult object at 0x0000015B5DDE7CA0>, <OSMPythonTools.api.ApiResult object at 0x0000015B5DDE7EB0>, <OSMPythonTools.api.ApiResult object at 0x0000015B5DDE7F70>, <OSMPythonTools.api.ApiResult object at 0x0000015B5DDE7F40>, <OSMPythonTools.api.ApiResult object at 0x0000015B5DE20520>, <OSMPythonTools.api.ApiResult object at 0x0000015B5DE20730>, <OSMPythonTools.api.ApiResult object at 0x0000015B5DE20940>, <OSMPythonTools.api.ApiResult object at 0x0000015B5DE20B50>, <OSMPythonTools.api.ApiResult object at 0x0000015B5DE20D60>, <OSMPythonTools.api.ApiResult object at 0x0000015B5DE20F70>, <OSMPythonTools.api.ApiResult object at 0x0000015B5DE21180>, <OSMPythonTools.api.ApiResult object at 0x0000015B5DE21390>, <OSMPythonTools.api.ApiResult object at 0x0000015B5DE215A0>, <OSMPythonTools.api.ApiResult object at 0x0000015B5DE217B0>, <OSMPythonTools.api.ApiResult object at 0x0000015B5DE219C0>], 'queryString': 'way/473223865', 'tags': {'barrier': 'fence', 'landuse': 'residential', 'name': 'TBMM Eski Lojmanları Sitesi'}, 'timestamp': datetime.datetime(2021, 11, 20, 21, 1, 48, tzinfo=tzutc()), 'type': 'way', 'typeId': 'way/473223865', 'typeIdShort': 'w473223865', 'uid': '14474713', 'user': 'pinkyPrii', 'userid': '14474713', 'version': 8, 'visible': 'true'}  
! {'address': None, 'areaId': 3603866327, 'cacheTimestamp': '2022-11-05T10:23:28.504432', 'cacheVersion': '1.0', 'displayName': 'Cumhuriyet Mahallesi, Üsküdar, İstanbul, Marmara Bölgesi, 34697, Türkiye', 'id': 3866327, 'isReverse': False, 'queryString': 'Cumhuriyet, Üsküdar', 'toJSON': [{'place\_id': 298278074, 'licence': 'Data © OpenStreetMap contributors, ODbL 1.0. https://osm.org/copyright', 'osm\_type': 'relation', 'osm\_id': 3866327, 'boundingbox': ['41.0056665', '41.015203', '29.0726392', '29.0894421'], 'lat': '41.0108797', 'lon': '29.0807595', 'display\_name': 'Cumhuriyet Mahallesi, Üsküdar, İstanbul, Marmara Bölgesi, 34697, Türkiye', 'class': 'boundary', 'type': 'administrative', 'importance': 0.45585065784831785, 'icon': 'https://nominatim.openstreetmap.org/ui/mapicons/poi\_boundary\_administrative.p.20.png', 'geotext': 'POLYGON((29.0726392 41.0056665,29.0738694 41.005781,29.07535 41.005969,29.076774 41.0063719,29.0775729 41.0066838,29.0785461 41.0070638,29.0814741 41.0082072,29.08346 41.0088827,29.0851591 41.0092204,29.0870841 41.0094598,29.0888092 41.0094893,29.0890691 41.0094836,29.0894421 41.0094753,29.0893821 41.0100059,29.0893227 41.0102341,29.0892849 41.0103792,29.0892807 41.0104834,29.0892748 41.0106322,29.0892579 41.0109075,29.0893658 41.0110954,29.0893737 41.0112263,29.0893546 41.0113694,29.0892576 41.0115952,29.0891316 41.0119404,29.0886524 41.0119708,29.0881341 41.0119357,29.0876894 41.0136003,29.087473 41.0140839,29.0873369 41.0143716,29.0870082 41.0150678,29.085803 41.0147199,29.0852201 41.0145594,29.0834767 41.0141223,29.0820926 41.0139442,29.0815401 41.0140332,29.081068 41.0144946,29.0808213 41.0147092,29.0798235 41.0150006,29.0791154 41.015203,29.0784019 41.014863,29.0772378 41.0143854,29.0775865 41.0137499,29.076701 41.0137941,29.0753174 41.0138632,29.0750211 41.0116305,29.0748212 41.0111856,29.0746415 41.0108253,29.074569 41.0103517,29.0745181 41.0100906,29.0744081 41.0096089,29.0742579 41.0092324,29.0740567 41.0087082,29.0736946 41.0080787,29.0734237 41.0076294,29.0730777 41.0069615,29.0729275 41.0066194,29.0728497 41.0063036,29.0727049 41.0058604,29.0726392 41.0056665))'}], 'type': 'relation', 'typeId': 'relation/3866327', 'typeIdShort': 'r3866327', 'wkt': 'POLYGON((29.0726392 41.0056665,29.0738694 41.005781,29.07535 41.005969,29.076774 41.0063719,29.0775729 41.0066838,29.0785461 41.0070638,29.0814741 41.0082072,29.08346 41.0088827,29.0851591 41.0092204,29.0870841 41.0094598,29.0888092 41.0094893,29.0890691 41.0094836,29.0894421 41.0094753,29.0893821 41.0100059,29.0893227 41.0102341,29.0892849 41.0103792,29.0892807 41.0104834,29.0892748 41.0106322,29.0892579 41.0109075,29.0893658 41.0110954,29.0893737 41.0112263,29.0893546 41.0113694,29.0892576 41.0115952,29.0891316 41.0119404,29.0886524 41.0119708,29.0881341 41.0119357,29.0876894 41.0136003,29.087473 41.0140839,29.0873369 41.0143716,29.0870082 41.0150678,29.085803 41.0147199,29.0852201 41.0145594,29.0834767 41.0141223,29.0820926 41.0139442,29.0815401 41.0140332,29.081068 41.0144946,29.0808213 41.0147092,29.0798235 41.0150006,29.0791154 41.015203,29.0784019 41.014863,29.0772378 41.0143854,29.0775865 41.0137499,29.076701 41.0137941,29.0753174 41.0138632,29.0750211 41.0116305,29.0748212 41.0111856,29.0746415 41.0108253,29.074569 41.0103517,29.0745181 41.0100906,29.0744081 41.0096089,29.0742579 41.0092324,29.0740567 41.0087082,29.0736946 41.0080787,29.0734237 41.0076294,29.0730777 41.0069615,29.0729275 41.0066194,29.0728497 41.0063036,29.0727049 41.0058604,29.0726392 41.0056665))'}  
! {'address': {'city': 'Heidelberg', 'state': 'Baden-Württemberg', 'ISO3166-2-lvl4': 'DE-BW', 'country': 'Deutschland', 'country\_code': 'de'}, 'areaId': 3600285864, 'cacheTimestamp': '2022-11-05T11:42:06.557488', 'cacheVersion': '1.0', 'displayName': 'Heidelberg, Baden-Württemberg, Deutschland', 'id': 285864, 'isReverse': True, 'queryString': [49.4093582, 8.694724], 'toJSON': [{'place\_id': 298086493, 'licence': 'Data © OpenStreetMap contributors, ODbL 1.0. https://osm.org/copyright', 'osm\_type': 'relation', 'osm\_id': 285864, 'lat': '49.4093582', 'lon': '8.694724', 'display\_name': 'Heidelberg, Baden-Württemberg, Deutschland', 'address': {'city': 'Heidelberg', 'state': 'Baden-Württemberg', 'ISO3166-2-lvl4': 'DE-BW', 'country': 'Deutschland', 'country\_code': 'de'}, 'boundingbox': ['49.3520029', '49.4596927', '8.5731788', '8.7940496']}], 'type': 'relation', 'typeId': 'relation/285864', 'typeIdShort': 'r285864', 'wkt': None}  
! {'areas': [], 'cacheTimestamp': '2022-11-05T12:05:16.223676', 'cacheVersion': '1.0', 'copyright': 'The data included in this document is from www.openstreetmap.org. The data is made available under ODbL.', 'countAreas': 0, 'countElements': 20, 'countNodes': 20, 'countRelations': 0, 'countWays': 0, 'elements': [<OSMPythonTools.element.Element object at 0x0000015B5DDE6A10>, <OSMPythonTools.element.Element object at 0x0000015B5DDE7CA0>, <OSMPythonTools.element.Element object at 0x0000015B5DDE7A90>, <OSMPythonTools.element.Element object at 0x0000015B5DDE4250>, <OSMPythonTools.element.Element object at 0x0000015B5DDE6B00>, <OSMPythonTools.element.Element object at 0x0000015B5DDE7040>, <OSMPythonTools.element.Element object at 0x0000015B5DDE56C0>, <OSMPythonTools.element.Element object at 0x0000015B5DE20F70>, <OSMPythonTools.element.Element object at 0x0000015B5DE21A20>, <OSMPythonTools.element.Element object at 0x0000015B5DE21DE0>, <OSMPythonTools.element.Element object at 0x0000015B5DE22050>, <OSMPythonTools.element.Element object at 0x0000015B5DE21CF0>, <OSMPythonTools.element.Element object at 0x0000015B5DE217B0>, <OSMPythonTools.element.Element object at 0x0000015B5DE20520>, <OSMPythonTools.element.Element object at 0x0000015B5DE21BD0>, <OSMPythonTools.element.Element object at 0x0000015B5DE21EA0>, <OSMPythonTools.element.Element object at 0x0000015B5DE21F60>, <OSMPythonTools.element.Element object at 0x0000015B5DE21D80>, <OSMPythonTools.element.Element object at 0x0000015B5DE21390>, <OSMPythonTools.element.Element object at 0x0000015B5DE20730>], 'generator': 'Overpass API 0.7.59 e21c39fe', 'isValid': True, 'nodes': [<OSMPythonTools.element.Element object at 0x0000015B5DDE6A10>, <OSMPythonTools.element.Element object at 0x0000015B5DDE7CA0>, <OSMPythonTools.element.Element object at 0x0000015B5DDE7A90>, <OSMPythonTools.element.Element object at 0x0000015B5DDE4250>, <OSMPythonTools.element.Element object at 0x0000015B5DDE6B00>, <OSMPythonTools.element.Element object at 0x0000015B5DDE7040>, <OSMPythonTools.element.Element object at 0x0000015B5DDE56C0>, <OSMPythonTools.element.Element object at 0x0000015B5DE20F70>, <OSMPythonTools.element.Element object at 0x0000015B5DE21A20>, <OSMPythonTools.element.Element object at 0x0000015B5DE21DE0>, <OSMPythonTools.element.Element object at 0x0000015B5DE22050>, <OSMPythonTools.element.Element object at 0x0000015B5DE21CF0>, <OSMPythonTools.element.Element object at 0x0000015B5DE217B0>, <OSMPythonTools.element.Element object at 0x0000015B5DE20520>, <OSMPythonTools.element.Element object at 0x0000015B5DE21BD0>, <OSMPythonTools.element.Element object at 0x0000015B5DE21EA0>, <OSMPythonTools.element.Element object at 0x0000015B5DE21F60>, <OSMPythonTools.element.Element object at 0x0000015B5DE21D80>, <OSMPythonTools.element.Element object at 0x0000015B5DE21390>, <OSMPythonTools.element.Element object at 0x0000015B5DE20730>], 'queryString': '[timeout:25][out:json];area(3603866327)->.searchArea;(node["highway"="bus\_stop"](area.searchArea);); out body;', 'relations': [], 'remark': None, 'timestamp\_osm\_base': datetime.datetime(2022, 11, 5, 9, 2, 44, tzinfo=tzutc()), 'toJSON': {'version': 0.6, 'generator': 'Overpass API 0.7.59 e21c39fe', 'osm3s': {'timestamp\_osm\_base': '2022-11-05T09:02:44Z', 'timestamp\_areas\_base': '2022-11-05T08:44:26Z', 'copyright': 'The data included in this document is from www.openstreetmap.org. The data is made available under ODbL.'}, 'elements': [{'type': 'node', 'id': 927955126, 'lat': 41.0105176, 'lon': 29.0838191, 'tags': {'bus': 'yes', 'highway': 'bus\_stop', 'public\_transport': 'platform'}}, {'type': 'node', 'id': 927955128, 'lat': 41.0118129, 'lon': 29.0880248, 'tags': {'bus': 'yes', 'highway': 'bus\_stop', 'name': 'Açan Sokak', 'public\_transport': 'platform'}}, {'type': 'node', 'id': 4486289891, 'lat': 41.0106672, 'lon': 29.08703, 'tags': {'bus': 'yes', 'highway': 'bus\_stop', 'name': 'Üçyol', 'public\_transport': 'platform'}}, {'type': 'node', 'id': 5239669821, 'lat': 41.0137177, 'lon': 29.0753145, 'tags': {'bus': 'yes', 'highway': 'bus\_stop', 'name': 'Çeşme', 'public\_transport': 'platform'}}, {'type': 'node', 'id': 7132896793, 'lat': 41.0119592, 'lon': 29.0880413, 'tags': {'bus': 'yes', 'highway': 'bus\_stop', 'name': 'Açan Sokak', 'public\_transport': 'platform'}}, {'type': 'node', 'id': 7132896810, 'lat': 41.0107082, 'lon': 29.0872894, 'tags': {'bus': 'yes', 'highway': 'bus\_stop', 'name': 'Üçyol', 'public\_transport': 'platform'}}, {'type': 'node', 'id': 7132896836, 'lat': 41.0146126, 'lon': 29.0856502, 'tags': {'bus': 'yes', 'highway': 'bus\_stop', 'name': 'Dörtyol', 'public\_transport': 'platform'}}, {'type': 'node', 'id': 7138816397, 'lat': 41.014198, 'lon': 29.0812956, 'tags': {'bus': 'yes', 'highway': 'bus\_stop', 'name': 'Cumhuriyet Mahallesi Muhtarlığı', 'public\_transport': 'platform'}}, {'type': 'node', 'id': 7138854975, 'lat': 41.0127702, 'lon': 29.0822821, 'tags': {'bus': 'yes', 'highway': 'bus\_stop', 'name': 'Özanakent', 'public\_transport': 'platform'}}, {'type': 'node', 'id': 7138854976, 'lat': 41.0127089, 'lon': 29.0824285, 'tags': {'bus': 'yes', 'highway': 'bus\_stop', 'name': 'Özanakent', 'public\_transport': 'platform'}}, {'type': 'node', 'id': 7138854983, 'lat': 41.0091346, 'lon': 29.0800034, 'tags': {'bus': 'yes', 'highway': 'bus\_stop', 'name': 'Şehit Recep Büyük', 'public\_transport': 'platform'}}, {'type': 'node', 'id': 7138854984, 'lat': 41.0091184, 'lon': 29.0801536, 'tags': {'bus': 'yes', 'highway': 'bus\_stop', 'name': 'Şehit Recep Büyük', 'public\_transport': 'platform'}}, {'type': 'node', 'id': 7138855286, 'lat': 41.0103895, 'lon': 29.0791987, 'tags': {'bus': 'yes', 'highway': 'bus\_stop', 'name': 'Karlıdere Caddesi', 'public\_transport': 'platform'}}, {'type': 'node', 'id': 7138855287, 'lat': 41.0104077, 'lon': 29.0793073, 'tags': {'bus': 'yes', 'highway': 'bus\_stop', 'name': 'Karlıdere Caddesi', 'public\_transport': 'platform'}}, {'type': 'node', 'id': 7138855296, 'lat': 41.0071632, 'lon': 29.0776645, 'tags': {'bus': 'yes', 'highway': 'bus\_stop', 'name': 'Deniz Sokak', 'public\_transport': 'platform'}}, {'type': 'node', 'id': 7138855297, 'lat': 41.0072199, 'lon': 29.0781527, 'tags': {'bus': 'yes', 'highway': 'bus\_stop', 'name': 'Deniz Sokak', 'public\_transport': 'platform'}}, {'type': 'node', 'id': 7138855300, 'lat': 41.0105838, 'lon': 29.0747141, 'tags': {'bus': 'yes', 'highway': 'bus\_stop', 'name': 'Kırklar', 'public\_transport': 'platform'}}, {'type': 'node', 'id': 7138862587, 'lat': 41.0077486, 'lon': 29.0736317, 'tags': {'bench': 'no', 'bus': 'yes', 'highway': 'bus\_stop', 'name': 'Devlet Su İşleri', 'public\_transport': 'platform', 'shelter': 'yes'}}, {'type': 'node', 'id': 7138862589, 'lat': 41.0063833, 'lon': 29.0750894, 'tags': {'bus': 'yes', 'highway': 'bus\_stop', 'name': 'Baraj Yolu Sokak', 'public\_transport': 'platform'}}, {'type': 'node', 'id': 7138862590, 'lat': 41.006325, 'lon': 29.0752541, 'tags': {'bus': 'yes', 'highway': 'bus\_stop', 'name': 'Baraj Yolu Sokak', 'public\_transport': 'platform'}}]}, 'version': 0.6, 'ways': []}  
  
  
temelSorgu = pd.DataFrame(columns =  
 ["temelSorguId", "cacheTimestamp", "cacheVersion", "queryString", "other"])  
alan = pd.DataFrame(columns =  
 ["alanId", "temelSorguId", "id", "areaId", "typeIdShort", "type", "typeId"])  
adres = pd.DataFrame(columns =  
 ["adresId", "alanId", "address", "displayName", "isReverse", "wkt"])  
node = pd.DataFrame(columns =  
 ["nodeId", "alanId", "countNodes", "generator", "nodes", "version",  
 "valid", "copyright"])  
"""

## UML - **Unified Modeling Language** (Birleşik Modelleme Dili

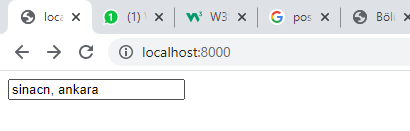
### Normalizasyon

Diyagram oluşturma sitesi: <https://app.diagrams.net/>

TUCBS Diagram : <https://tucbs-public-api.csb.gov.tr/tucbs/tucbs_uml_model/index.htm>

## Web Sitesi Yapımı

localhost:8000



import uvicorn  
from fastapi import FastAPI  
app = FastAPI()  
  
#: Kütüphaneleri import etme  
import sys  
from OSMPythonTools.overpass import overpassQueryBuilder  
from OSMPythonTools.api import Api  
from OSMPythonTools.nominatim import Nominatim  
from OSMPythonTools.overpass import Overpass  
  
#: Api'leri başlatma  
overpass = Overpass()  
nominatim = Nominatim()  
api = Api()  
  
#  
def özellikÇıkar( nesne: object, hariç: list = [] ) -> dict:  
 info = {}  
 for a in dir( nesne ):  
 if a[0] != "\_" and a not in hariç:  
 method = getattr( nesne, a )  
 info[a] = method()  
 return info  
  
def sorgula( girdi, ekstra = 10 ):  
 if type(girdi) is int:  
 sonuc = api.query(f"way/{girdi}")  
 return özellikÇıkar( sonuc, ["fromId", "tag", "toXML", "nodes", "timestamp"] )  
 if type(girdi) is str:  
 sonuc = nominatim.query(girdi, wkt=True)  
 return özellikÇıkar( sonuc )  
 if type(girdi) is tuple:  
 if type(girdi[0]) is float:  
 sonuc = nominatim.query(girdi[0], girdi[1], reverse=True, zoom= ekstra)  
 return özellikÇıkar( sonuc )  
 if type(girdi[0]) is str:  
 loc = nominatim.query( girdi[0] )  
 query = overpassQueryBuilder(area=loc, elementType='node', selector=girdi[1], out='body')  
 sonuc = overpass.query(query)  
 return özellikÇıkar( sonuc, ["timestamp\_area\_base"] )  
 raise Exception("Bilinmeyen format")  
  
  
  
from fastapi.responses import HTMLResponse  
  
@app.get("/", response\_class=HTMLResponse)  
async def root():  
 return "<form action='sorgula' method='get'><input type='text' name='sorgu' id='sorgu'></form>"  
 #return sorgula(473223865)  
  
@app.get("/sorgula")  
async def xdsfgdsfgfsd(sorgu):  
 if isNumeric(sorgu):  
 return sorgula(int(sorgu))  
 elif isCoordinate(sorgu):  
 return sorgula(tuple([float(k) for k in sorgu.split(" ")]))  
 else:  
 return sorgula(sorgu)  
  
import re  
def isNumeric(metin: str) -> bool:  
 return re.match("^[0-9]+$", metin)  
  
def isCoordinate(metin: str) -> bool:  
 return re.match("^[0-9\,\s\.]+$", metin)  
  
  
uvicorn.run(app, host="0.0.0.0", port=8000)

# 12.GÜN – 07.11.2022 – PAZAR \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

## POSTGRESQL

### Pgadmin

Pycharm da veri tabanı için random veri üretme

import random  
for \_ in range(250):  
 id = random.randint(1, 100000)  
 alan = random.random() \* 1000  
 no = random.randint(1, 100)  
 ilid = random.randint(1, 81)  
 print(f"INSERT INTO parsel(id, alan, no, ilid) VALUES ({id}, {alan}, {no}, {ilid});")

### pgadmin nasıl yüklenir ?

<https://www.youtube.com/watch?v=iTNn4u8gPdA>

### Veritabanı oluşturma

CREATE DATABASE *databasename*;

### Tablo oluşturma

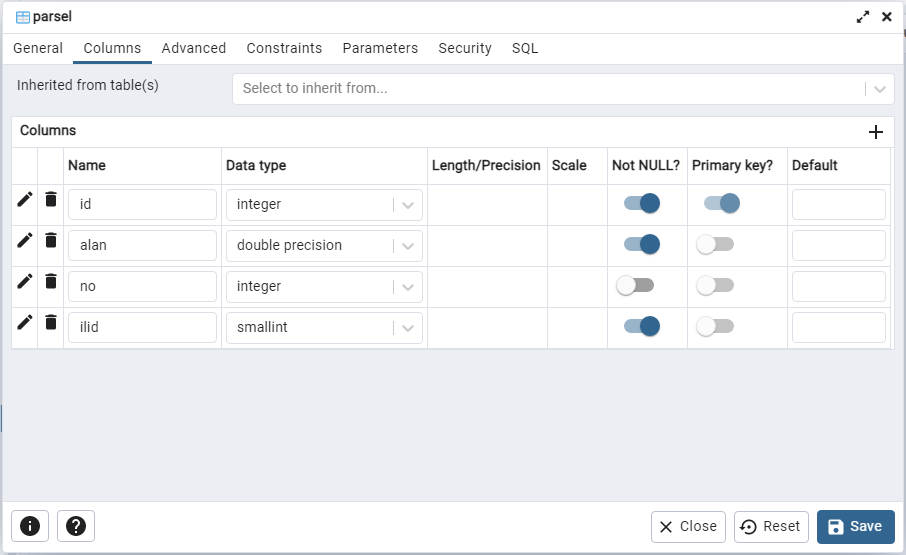
CREATE TABLE *table\_name*(  
*column1 datatype*,  
*column2 datatype*,  
*column3 datatype*,  
   ....  
);

Bizim örneğimizde

Aşağıdaki tabloyu oluşturmak için

CREATE TABLE *parsel*(  
*id integer*,  
*alan double*,  
*no integer*,

*ilid smallint*     
);



# 13.GÜN – 11.11.2022 – CUMA \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

## Yedek alma (veritabanı)

Pgadmin4 de veritabanı yedeğini alma

<https://turnateknoloji.com/tr/atolye/postgresql-yedek-alma-%28backup%29-ve-kurma-%28restore%29>

## Postgresql de Sanal Tablo Oluşturma

create temporary table tablo3

as

select \* from cd.bookings b where b.slots >3

# 14.GÜN – 12.11.2022 – CUMARTESİ \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

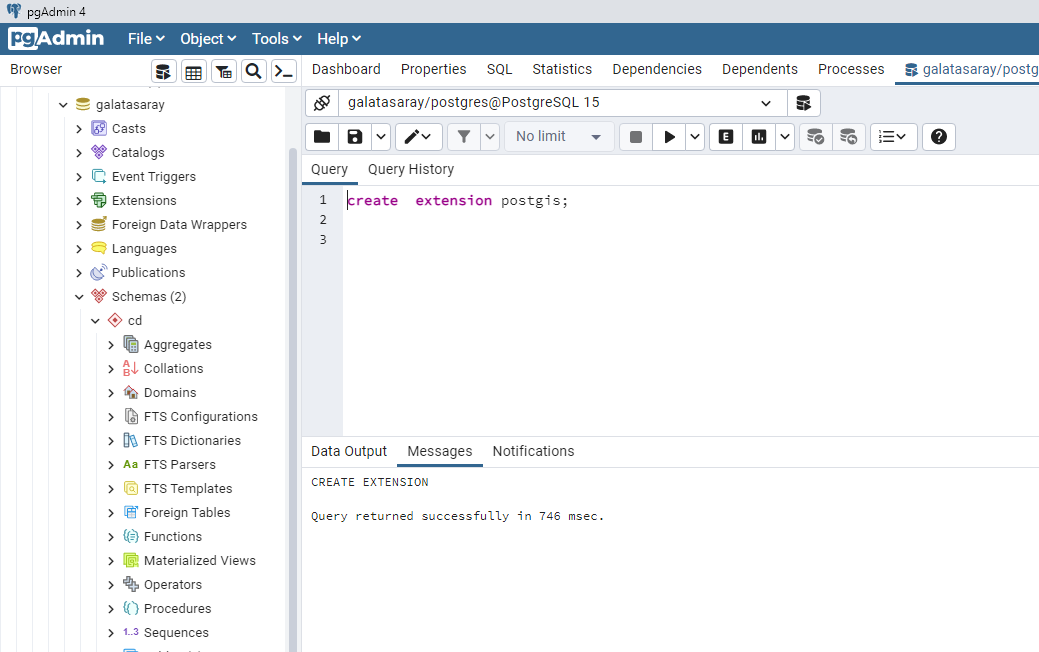
## POSTGIS

### Kurulumu

Application stack builder programını indirip kurun

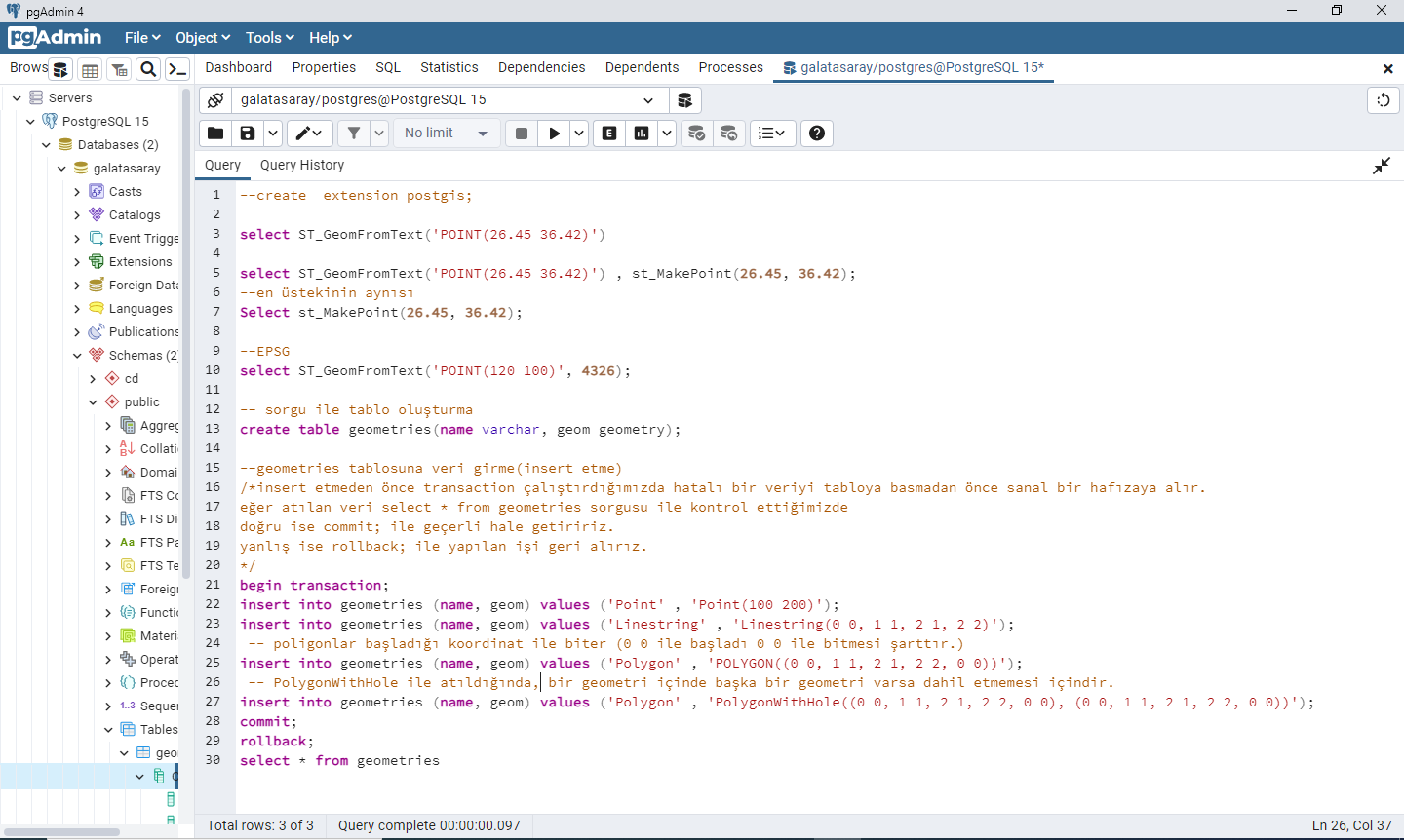
### Kurulumdan sonra

pgAdmin programını çalıştırıp query (sorgu yazdığımız ekran ) kısmına alttaki “create extension postgis;” sorgusunu bir kereliğine çalıştırın.



## EPSG – Konumsal Referans Sistem ID’leri - 4326

<https://epsg.io/?q=turef>



### Polygon – Poligon örnekleri

insert into geometries (name, geom) values ('Point' , 'Point(100 200)');

insert into geometries (name, geom) values ('Linestring' , 'Linestring(0 0, 1 1, 2 1, 2 2)');

-- poligonlar başladığı koordinat ile biter (0 0 ile başladı 0 0 ile bitmesi şarttır.)

insert into geometries (name, geom) values ('Polygon' , 'POLYGON((0 0, 1 1, 2 1, 2 2, 0 0))');

-- PolygonWithHole ile atıldığında, bir geometri içinde başka bir geometri varsa dahil etmemesi içindir.

insert into geometries (name, geom) values ('Polygon' , 'PolygonWithHole((0 0, 1 1, 2 1, 2 2, 0 0), (0 0, 1 1, 2 1, 2 2, 0 0))');

## Veri Gönderme - PostgreSQL veri tabanımıza Python kodları ile veri gönderme

import psycopg2  
  
conn=psycopg2.connect("dbname=galatasaray user=postgres password=123")  
cur=conn.cursor()  
  
  
import random  
def noktauret() ->str:  
 boylam = 26+random.random()\*19  
 enlem = 36+random.random()\*6  
 return f"POINT({boylam} {enlem})"  
  
for \_ in range (1000):  
 r = noktauret()  
 print(r)  
 cur.execute(f"insert into geometries values ('Point', '{r}');")  
  
conn.commit()  
cur.close()  
conn.close()

## Pycharm (Python) programındayken postgresql veri tabanımızda bulunan tablolarımızı nasıl çekeriz

import psycopg2  
conn = psycopg2.connect("dbname=galatasaray user=postgres password=123")  
cur = conn.cursor()  
cur.execute("Select \* from cd.bookings limit 3")  
  
print("---------------------------tablodaki kolanları ve tiplerini getirir------------------------------")  
sonuclar = cur.fetchall()  
for s in sonuclar:  
 print(s)  
  
print("----------------------------------------------------------------------------------")  
cur.execute("select \* from cd.bookings limit 1")  
sonuclar = cur.fetchall()  
for s in sonuclar:  
 for a in s :  
 print(a, type(a))  
 print(s)  
  
print("---------------------------------Kaçtane sonuç geldi---------------------------------")  
cur.execute("select \* from cd.bookings")  
sonuclar = cur.fetchall()  
#: Kaçtane sonuç geldi.  
print(len(sonuclar))  
  
print("--------------------------------------------------------------------------------------")  
print(type(cur.description), cur.description)  
  
print("---------------------sözlük olarak getirdiğimizde------------------------------------")  
def getir(sorgu:str) -> dict:  
 cur.execute(sorgu)  
 sonuclar = cur.fetchall()  
 columns = [c[0] for c in cur.description]  
 return {"sonuçlar" : sonuclar,  
 "kolonlar" : columns,  
 "adeti" : len(sonuclar)  
 }  
r1 = getir("Select \* from cd.bookings limit 1;")  
r2 = getir("Select \* from cd.facilities limit 1;")  
r3 = getir("Select \* from cd.members limit 1;")  
print("sonuçlar: ", r1["sonuçlar"][0])  
print("kolonlar: ", r2["kolonlar"])  
print("adedi: ",r3["adeti"])  
  
print("---count-------------------------------------------------------------------------------------")  
print(getir("select count(b.bookid) from cd.bookings b"))  
print("----------------------------------------------------------------------------------------")

## Polygon insert ederken

import random  
import psycopg2  
  
conn=psycopg2.connect("dbname=galatasaray user=postgres password=123")  
cur=conn.cursor()  
  
  
def çokgenÜret() -> str:  
 def koordinatÜret() -> str:  
 b = 26 + random.random() \* 19 # 26 ile 45 arası  
 e = 36 + random.random() \* 6 # 36 ile 42  
 return f"{b} {e}"  
  
 poly = []  
 for \_ in range(random.randint(3, 20)):  
 poly.append( koordinatÜret() )  
  
 poly.append(poly[0])  
 return "POLYGON((" + ",".join(poly) + "))"  
  
  
  
for \_ in range(1000):  
 p = çokgenÜret()  
 s = f"INSERT INTO geometries VALUES ('Polygon', '{p}');"  
 cur.execute(s)  
## INSERT INTO geometries VALUES ('Polygon', 'POLYGON((0 0, 1 0, 1 1, 0 1, 0 0))');  
  
  
  
conn.commit()

## Point insert ederken

import psycopg2  
import random  
  
conn=psycopg2.connect("dbname=galatasaray user=postgres password=123")  
cur=conn.cursor()  
  
  
  
def noktauret() ->str:  
 boylam = 26+random.random()\*19  
 enlem = 36+random.random()\*6  
 return f"POINT({boylam} {enlem})"  
  
for \_ in range (1000):  
 r = noktauret()  
 print(r)  
 cur.execute(f"insert into geometries values ('Point', '{r}');")  
  
conn.commit()

## Linestring insert ederken

import psycopg2  
import random  
  
conn=psycopg2.connect("dbname=galatasaray user=postgres password=123")  
cur=conn.cursor()  
  
  
def doğrultuÜret() -> str:  
 def kooordinatÜret() -> str:  
 boylam = 26 + random.random() \* 19  
 enlem = 36 + random.random() \* 6  
 return f"{boylam} {enlem}"  
  
 k1 = kooordinatÜret()  
 k2 = kooordinatÜret()  
 k3 = kooordinatÜret()  
 k4 = kooordinatÜret()  
 return f"LINESTRING({k1},{k2},{k3},{k4})"  
  
for \_ in range(1000):  
 r = doğrultuÜret()  
 cur.execute(f"insert into geometries values ('Linestring', '{r}');")  
  
conn.commit()

## Uzaklık bulma

1-- Anakra'nın 10 birimden yakın noktaları getir

select \* from geometries g where ST\_Distance(ST\_MakePoint(39.9221783 , 32.8125787), g.geom) < 10

2-- Öyle nokatalar bulacağız ki Ankara ve İstanbul'a eşit uzaklıkta olacak 41.0066009,28.9728604

select \* from geometries g

where ABS(ST\_Distance(ST\_MakePoint(41.0066009,28.9728604), g.geom) - ST\_Distance(ST\_MakePoint(39.9221783 , 32.8125787), g.geom) ) <2

# PostGIS - ST\_ SÖZLÜK

## PostGIS Geometri/Coğrafya/Kutu Veri Tipleri

Bu bölüm, uzamsal verileri temsil etmek için PostGIS tarafından kurulan özel PostgreSQL veri türlerini listeler.

Her veri türü, kendi tür döküm davranışını tanımlar. Bir [tür dökümü](https://www.postgresql.org/docs/current/sql-expressions.html#SQL-SYNTAX-TYPE-CASTS) , bir veri türünün değerlerini başka bir türe dönüştürür. PostgreSQL, tür değerlerini dönüştürmek için kullanılan işlevlerle birlikte özel türler için döküm davranışını tanımlamaya izin verir. Yayınlar, bir işlev bağımsız değişkeninin işlev tarafından desteklenen bir türe otomatik olarak dönüştürülmesine izin veren **otomatik davranışa sahip olabilir.**

Bazı yayınların **açıkCAST(myval As sometype)** davranışı vardır; bu, yayının veya sözdizimi kullanılarak belirtilmesi gerektiği anlamına gelir **myval::sometype**. Açık döküm, belirli bir türü desteklemeyen aşırı yüklenmiş bir işlev kullanıldığında ortaya çıkabilecek belirsiz yayın sorununu önler. Örneğin, bir işlev bir box2d veya bir box3d'yi kabul edebilir, ancak bir geometriyi kabul edemez. Geometrinin her iki kutu tipine de otomatik bir dökümü olduğundan, bu bir "belirsiz işlev" hatası üretir. Hatayı önlemek için, istenen kutu tipine açık bir döküm kullanın.

Tüm veri türleri için yayınlanabilir **text**, bu nedenle bunun açıkça belirtilmesi gerekmez.

[box2d](https://postgis.net/docs/box2d_type.html) — 2 boyutlu bir sınırlayıcı kutuyu temsil eden tür.

[box3d](https://postgis.net/docs/box3d_type.html) — 3 boyutlu bir sınırlayıcı kutuyu temsil eden tür.

[geometri](https://postgis.net/docs/geometry.html) — Düzlemsel koordinat sistemleriyle uzamsal özellikleri temsil eden tip.

[geometri\_dump](https://postgis.net/docs/geometry_dump.html) — Karmaşık geometrinin parçalarını tanımlamak için kullanılan bir bileşik tür.

[coğrafya](https://postgis.net/docs/geography.html) — Jeodezik (elipsoidal) koordinat sistemleriyle uzamsal özellikleri temsil eden tip.

### Tablo Yönetim Fonksiyonları

Bu işlevler, geometri sütunları içeren tabloların tanımlanmasına yardımcı olur.

[AddGeometryColumn](https://postgis.net/docs/AddGeometryColumn.html) — Mevcut bir tabloya bir geometri sütunu ekler.

[DropGeometryColumn](https://postgis.net/docs/DropGeometryColumn.html) — Uzamsal tablodan bir geometri sütununu kaldırır.

[DropGeometryTable](https://postgis.net/docs/DropGeometryTable.html) — Bir tabloyu ve tüm referanslarını geometri\_sütunlar içine bırakır.

[Find\_SRID](https://postgis.net/docs/Find_SRID.html) — Bir geometri sütunu için tanımlanan SRID'yi döndürür.

[Populate\_Geometry\_Columns](https://postgis.net/docs/Populate_Geometry_Columns.html) — Geometri sütunlarının tür değiştiricilerle tanımlanmasını veya uygun uzamsal kısıtlamalara sahip olmasını sağlar.

[UpdateGeometrySRID](https://postgis.net/docs/UpdateGeometrySRID.html) — Bir geometri sütunundaki tüm özelliklerin SRID'sini ve tablo meta verilerini günceller.

### Geometri Oluşturucuları

[ST\_Collect](https://postgis.net/docs/ST_Collect.html) — Bir dizi geometriden bir GeometryCollection veya Multi\* geometri oluşturur.

[ST\_LineFromMultiPoint](https://postgis.net/docs/ST_LineFromMultiPoint.html) — Bir MultiPoint geometrisinden bir LineString oluşturur.

[ST\_MakeEnvelope](https://postgis.net/docs/ST_MakeEnvelope.html) — Minimum ve maksimum koordinatlardan dikdörtgen bir Çokgen oluşturur.

[ST\_MakeLine](https://postgis.net/docs/ST_MakeLine.html) — Point, MultiPoint veya LineString geometrilerinden bir LineString oluşturur.

[ST\_MakePoint](https://postgis.net/docs/ST_MakePoint.html) — 2B, 3BZ veya 4B Nokta oluşturur.

[ST\_MakePointM](https://postgis.net/docs/ST_MakePointM.html) — X, Y ve M değerlerinden bir Nokta oluşturur.

[ST\_MakePolygon](https://postgis.net/docs/ST_MakePolygon.html) — Bir kabuktan ve isteğe bağlı delikler listesinden bir Çokgen oluşturur.

[ST\_Point](https://postgis.net/docs/ST_Point.html) — X, Y ve SRID değerlerine sahip bir Nokta oluşturur.

[ST\_PointZ](https://postgis.net/docs/ST_PointZ.html) — X, Y, Z ve SRID değerlerine sahip bir Nokta oluşturur.

[ST\_PointM](https://postgis.net/docs/ST_PointM.html) — X, Y, M ve SRID değerlerine sahip bir Nokta oluşturur.

[ST\_PointZM](https://postgis.net/docs/ST_PointZM.html) — X, Y, Z, M ve SRID değerlerine sahip bir Nokta oluşturur.

[ST\_Polygon](https://postgis.net/docs/ST_Polygon.html) — Belirtilen SRID'ye sahip bir LineString'den bir Çokgen oluşturur.

[ST\_TileEnvelope](https://postgis.net/docs/ST_TileEnvelope.html) — XYZ döşeme sistemini kullanarak Web Mercator'da (SRID:3857) dikdörtgen bir Çokgen oluşturur.

[ST\_HexagonGrid](https://postgis.net/docs/ST_HexagonGrid.html) — Geometri argümanının sınırlarını tamamen kapsayan bir dizi altıgen ve hücre indeksi döndürür.

[ST\_Hexagon](https://postgis.net/docs/ST_Hexagon.html) — Sağlanan kenar boyutunu ve altıgen ızgara alanı içindeki hücre koordinatını kullanarak tek bir altıgen döndürür.

[ST\_SquareGrid](https://postgis.net/docs/ST_SquareGrid.html) — Geometri argümanının sınırlarını tamamen kapsayan bir dizi ızgara kare ve hücre indeksi döndürür.

[ST\_Square](https://postgis.net/docs/ST_Square.html) — Kare ızgara alanı içinde sağlanan kenar boyutunu ve hücre koordinatını kullanarak tek bir kare döndürür.

[ST\_Letters](https://postgis.net/docs/ST_Letters.html) — Başlangıç ​​konumunda varsayılan başlangıç ​​konumu ve varsayılan metin yüksekliği 100 olan geometri olarak oluşturulan giriş harflerini döndürür.

### Geometri Aksesuarlar

[GeometryType](https://postgis.net/docs/GeometryType.html) — Bir geometrinin türünü metin olarak döndürür.

[ST\_Boundary](https://postgis.net/docs/ST_Boundary.html) — Bir geometrinin sınırını döndürür.

[ST\_BoundingDiagonal](https://postgis.net/docs/ST_BoundingDiagonal.html) — Bir geometrinin sınırlayıcı kutusunun köşegenini döndürür.

[ST\_CoordDim](https://postgis.net/docs/ST_CoordDim.html) — Bir geometrinin koordinat boyutunu döndürür.

[ST\_Dimension](https://postgis.net/docs/ST_Dimension.html) — Bir geometrinin topolojik boyutunu döndürür.

[ST\_Dump](https://postgis.net/docs/ST_Dump.html) — Bir geometrinin bileşenleri için bir dizi satır döndürür .**geometry\_dump**

[ST\_DumpPoints](https://postgis.net/docs/ST_DumpPoints.html) — Bir geometrideki koordinatlar için bir dizi satır döndürür .**geometry\_dump**

[ST\_DumpSegments](https://postgis.net/docs/ST_DumpSegments.html) — Bir geometrideki segmentler için bir dizi satır döndürür .**geometry\_dump**

[ST\_DumpRings](https://postgis.net/docs/ST_DumpRings.html) — Bir Çokgenin dış ve iç halkaları için bir dizi satır döndürür.**geometry\_dump**

[ST\_EndPoint](https://postgis.net/docs/ST_EndPoint.html) — LineString veya CircularLineString öğesinin son noktasını döndürür.

[ST\_Zarf](https://postgis.net/docs/ST_Envelope.html) — Bir geometrinin sınırlayıcı kutusunu temsil eden bir geometri döndürür.

[ST\_ExteriorRing](https://postgis.net/docs/ST_ExteriorRing.html) — Bir Çokgenin dış halkasını temsil eden bir LineString döndürür.

[ST\_GeometryN](https://postgis.net/docs/ST_GeometryN.html) — Bir geometri koleksiyonunun bir öğesini döndürür.

[ST\_GeometryType](https://postgis.net/docs/ST_GeometryType.html) — Bir geometrinin SQL-MM türünü metin olarak döndürür.

[ST\_HasArc](https://postgis.net/docs/ST_HasArc.html) — Bir geometrinin dairesel bir yay içerip içermediğini test eder

[ST\_InteriorRingN](https://postgis.net/docs/ST_InteriorRingN.html) — Bir Çokgenin N. iç halkasını (delik) döndürür.

[ST\_IsClosed](https://postgis.net/docs/ST_IsClosed.html) — LineStrings'in başlangıç ​​ve bitiş noktalarının çakışıp çakışmadığını test eder. Kapalı (hacimsel) ise ÇokyüzlüYüzey testleri için.

[ST\_IsCollection](https://postgis.net/docs/ST_IsCollection.html) — Bir geometrinin bir geometri koleksiyonu türü olup olmadığını test eder.

[ST\_IsEmpty](https://postgis.net/docs/ST_IsEmpty.html) — Bir geometrinin boş olup olmadığını test eder.

[ST\_IsPolygonCCW](https://postgis.net/docs/ST_IsPolygonCCW.html)— Çokgenlerin saat yönünün tersine yönlendirilmiş dış halkaları ve saat yönünde yönlendirilmiş iç halkaları olup olmadığını test eder.

[ST\_IsPolygonCW](https://postgis.net/docs/ST_IsPolygonCW.html)— Çokgenlerin saat yönünde yönlendirilmiş dış halkalara ve saat yönünün tersine yönlendirilmiş iç halkalara sahip olup olmadığını test eder.

[ST\_IsRing](https://postgis.net/docs/ST_IsRing.html) — LineString'in kapalı ve basit olup olmadığını test eder.

[ST\_IsSimple](https://postgis.net/docs/ST_IsSimple.html) — Bir geometrinin kendi kendine kesişen veya kendinden teğet noktası olup olmadığını test eder.

[ST\_M](https://postgis.net/docs/ST_M.html) — Bir Noktanın M koordinatını döndürür.

[ST\_MemSize](https://postgis.net/docs/ST_MemSize.html) — Bir geometrinin kapladığı bellek alanı miktarını döndürür.

[ST\_NDims](https://postgis.net/docs/ST_NDims.html) — Bir geometrinin koordinat boyutunu döndürür.

[ST\_NPoints](https://postgis.net/docs/ST_NPoints.html) — Bir geometrideki noktaların (köşelerin) sayısını verir.

[ST\_NRings](https://postgis.net/docs/ST_NRings.html) — Çokgen geometrideki halka sayısını döndürür.

[ST\_NumGeometries](https://postgis.net/docs/ST_NumGeometries.html) — Bir geometri koleksiyonundaki öğelerin sayısını döndürür.

[ST\_NumInteriorRings](https://postgis.net/docs/ST_NumInteriorRings.html) — Bir Çokgenin iç halkalarının (deliklerinin) sayısını verir.

[ST\_NumInteriorRing](https://postgis.net/docs/ST_NumInteriorRing.html) — Bir Çokgenin iç halkalarının (deliklerinin) sayısını verir. ST\_NumInteriorRings için Aias

[ST\_NumPatches](https://postgis.net/docs/ST_NumPatches.html) — Çokyüzlü Yüzeydeki yüzlerin sayısını döndürür. Çokyüzlü olmayan geometriler için null değerini döndürür.

[ST\_NumPoints](https://postgis.net/docs/ST_NumPoints.html) — LineString veya CircularString içindeki nokta sayısını döndürür.

[ST\_PatchN](https://postgis.net/docs/ST_PatchN.html) — Bir ÇokyüzlüYüzeyin N. geometrisini (yüzünü) döndürür.

[ST\_PointN](https://postgis.net/docs/ST_PointN.html) — Bir geometrideki ilk LineString veya dairesel LineString'deki N. noktayı döndürür.

[ST\_Points](https://postgis.net/docs/ST_Points.html) — Bir geometrinin koordinatlarını içeren bir Çoklu Nokta döndürür.

[ST\_StartPoint](https://postgis.net/docs/ST_StartPoint.html) — LineString'in ilk noktasını döndürür.

[ST\_Summary](https://postgis.net/docs/ST_Summary.html) — Bir geometrinin içeriğinin metin özetini döndürür.

[ST\_X](https://postgis.net/docs/ST_X.html) — Bir Noktanın X koordinatını döndürür.

[ST\_Y](https://postgis.net/docs/ST_Y.html) — Bir Noktanın Y koordinatını döndürür.

[ST\_Z](https://postgis.net/docs/ST_Z.html) — Bir Noktanın Z koordinatını döndürür.

[ST\_Zmflag](https://postgis.net/docs/ST_Zmflag.html) — Bir geometrinin ZM koordinat boyutunu gösteren bir kod döndürür.

### Geometri Editörleri

Bu işlevler, tür, yapı veya köşeleri değiştirerek değiştirilmiş geometriler oluşturur.

[ST\_AddPoint](https://postgis.net/docs/ST_AddPoint.html) — LineString'e bir nokta ekleyin.

[ST\_CollectionExtract](https://postgis.net/docs/ST_CollectionExtract.html) — Verilen bir geometri koleksiyonu, yalnızca belirtilen türdeki öğeleri içeren bir çoklu geometri döndürür.

[ST\_CollectionHomogenize](https://postgis.net/docs/ST_CollectionHomogenize.html) — Bir geometri koleksiyonunun en basit temsilini döndürür.

[ST\_CurveToLine](https://postgis.net/docs/ST_CurveToLine.html) — Eğriler içeren bir geometriyi doğrusal bir geometriye dönüştürür.

[ST\_Scroll](https://postgis.net/docs/ST_Scroll.html) — Kapalı bir LineString'in başlangıç ​​noktasını değiştirin.

[ST\_FlipCoordinates](https://postgis.net/docs/ST_FlipCoordinates.html) — X ve Y ekseni ters çevrilmiş bir geometri sürümünü döndürür.

[ST\_Force2D](https://postgis.net/docs/ST_Force2D.html) — Geometrileri "2 boyutlu moda" zorla.

[ST\_Force3D](https://postgis.net/docs/ST_Force_3D.html) — Geometrileri XYZ moduna zorlayın. Bu, ST\_Force3DZ için bir takma addır.

[ST\_Force3DZ](https://postgis.net/docs/ST_Force_3DZ.html) — Geometrileri XYZ moduna zorla.

[ST\_Force3DM](https://postgis.net/docs/ST_Force_3DM.html) — Geometrileri XYM moduna zorlayın.

[ST\_Force4D](https://postgis.net/docs/ST_Force_4D.html) — Geometrileri XYZM moduna zorlayın.

[ST\_ForcePolygonCCW](https://postgis.net/docs/ST_ForcePolygonCCW.html)— Tüm dış halkaları saat yönünün tersine ve tüm iç halkaları saat yönünde yönlendirir.

[ST\_ForceCollection](https://postgis.net/docs/ST_Force_Collection.html) — Geometriyi bir GEOMETRYCOLLECTION'a dönüştürün.

[ST\_ForcePolygonCW](https://postgis.net/docs/ST_ForcePolygonCW.html)— Tüm dış halkaları saat yönünde ve tüm iç halkaları saat yönünün tersine yönlendirir.

[ST\_ForceSFS](https://postgis.net/docs/ST_ForceSFS.html) — Geometrileri yalnızca SFS 1.1 geometri türlerini kullanmaya zorlayın.

[ST\_ForceRHR](https://postgis.net/docs/ST_ForceRHR.html) — Sağ El Kuralını takip etmek için bir çokgendeki köşelerin oryantasyonunu zorlayın.

[ST\_ForceCurve](https://postgis.net/docs/ST_ForceCurve.html) — Uygunsa, bir geometriyi kavisli tipine yükseltin.

[ST\_LineToCurve](https://postgis.net/docs/ST_LineToCurve.html) — Doğrusal bir geometriyi eğri bir geometriye dönüştürür.

[ST\_Multi](https://postgis.net/docs/ST_Multi.html) — Geometriyi MULTI\* geometrisi olarak döndürür.

[ST\_Normalize](https://postgis.net/docs/ST_Normalize.html) — Geometriyi kurallı biçiminde döndürür.

[ST\_QuantizeCoordinates](https://postgis.net/docs/ST_QuantizeCoordinates.html)— En az anlamlı koordinat bitlerini sıfıra ayarlar

[ST\_RemovePoint](https://postgis.net/docs/ST_RemovePoint.html) — Çizgi dizisinden bir noktayı kaldırın.

[ST\_RemoveRepeatedPoints](https://postgis.net/docs/ST_RemoveRepeatedPoints.html) — Yinelenen noktaları kaldırılmış bir geometri sürümünü döndürür.

[ST\_Reverse](https://postgis.net/docs/ST_Reverse.html) — Geometriyi köşe sırası ters çevrilmiş olarak döndürür.

[ST\_Segmentize](https://postgis.net/docs/ST_Segmentize.html) — Verilen mesafeden daha uzun segmenti olmayan değiştirilmiş bir geometri/coğrafya döndürür.

[ST\_SetPoint](https://postgis.net/docs/ST_SetPoint.html) — Bir çizgi dizisinin noktasını belirli bir noktayla değiştirin.

[ST\_ShiftLongitude](https://postgis.net/docs/ST_Shift_Longitude.html) — Bir geometrinin boylam koordinatlarını -180..180 ile 0..360 arasında kaydırır.

[ST\_WrapX](https://postgis.net/docs/ST_WrapX.html) — Bir geometriyi bir X değeri etrafına sarın.

[ST\_SnapToGrid](https://postgis.net/docs/ST_SnapToGrid.html) — Giriş geometrisinin tüm noktalarını normal bir ızgaraya yasla.

[ST\_Snap](https://postgis.net/docs/ST_Snap.html) — Giriş geometrisinin parçalarını ve tepe noktalarını bir referans geometrinin tepe noktalarına yasla.

[ST\_SwapOrdinates](https://postgis.net/docs/ST_SwapOrdinates.html) — Verilen ordinat değerleri değiştirilmiş olarak verilen geometrinin bir sürümünü döndürür.

### Geometri Doğrulama

Bu işlevler, geometrilerin OGC SFS standardına göre geçerli olup olmadığını test eder. Ayrıca geçersizliğin mahiyeti ve yeri hakkında bilgi verirler. Geçersiz bir geometriden geçerli bir geometri yaratma işlevi de vardır.

[ST\_IsValid](https://postgis.net/docs/ST_IsValid.html) — Bir geometrinin 2B'de iyi oluşturulup oluşturulmadığını test eder.

[ST\_IsValidDetail](https://postgis.net/docs/ST_IsValidDetail.html) — Bir geometrinin geçerli olup olmadığını veya değilse bir neden ve konumu belirten bir satır döndürür .**valid\_detail**

[ST\_IsValidReason](https://postgis.net/docs/ST_IsValidReason.html) — Bir geometrinin geçerli olup olmadığını veya geçersizlik nedenini belirten metni döndürür.

[ST\_MakeValid](https://postgis.net/docs/ST_MakeValid.html) — Geçersiz bir geometriyi köşeleri kaybetmeden geçerli kılmaya çalışır.

### Mekansal Referans Sistemi Fonksiyonları

**spatial\_ref\_sys**Bu işlevler, tabloda tanımlandığı gibi geometrilerin Uzamsal Referans Sistemi ile çalışır .

[ST\_SetSRID](https://postgis.net/docs/ST_SetSRID.html) — SRID'yi bir geometriye ayarlayın.

[ST\_SRID](https://postgis.net/docs/ST_SRID.html) — Bir geometri için uzaysal referans tanımlayıcısını döndürür.

[ST\_Transform](https://postgis.net/docs/ST_Transform.html) — Koordinatları farklı bir uzaysal referans sistemine dönüştürülmüş yeni bir geometri döndürün.

## Geometri Girişi

Bu işlevler, çeşitli metinsel veya ikili biçimlerden geometri nesneleri oluşturur.

[Tanınmış Metin (WKT)](https://postgis.net/docs/reference.html#idm8242)

[İyi Bilinen İkili Dosya (WKB)](https://postgis.net/docs/reference.html#idm8875)

[Diğer Formatlar](https://postgis.net/docs/reference.html#idm9195)

### Tanınmış Metin (WKT)

[ST\_BdPolyFromText](https://postgis.net/docs/ST_BdPolyFromText.html) — Bir MultiLineString İyi Bilinen metin temsili olarak rastgele bir kapalı çizgi dizileri koleksiyonu verilen bir Çokgen oluşturun.

[ST\_BdMPolyFromText](https://postgis.net/docs/ST_BdMPolyFromText.html) — Bir MultiLineString metin temsili olarak keyfi bir kapalı çizgi dizileri koleksiyonu verilen bir MultiPolygon oluşturun İyi Bilinen metin temsili.

[ST\_GeogFromText](https://postgis.net/docs/ST_GeogFromText.html) — İyi Bilinen Metin gösteriminden veya genişletilmişten (WKT) belirtilen bir coğrafya değeri döndürür.

[ST\_GeographyFromText](https://postgis.net/docs/ST_GeographyFromText.html) — İyi Bilinen Metin gösteriminden veya genişletilmişten (WKT) belirtilen bir coğrafya değeri döndürür.

[ST\_GeomCollFromText](https://postgis.net/docs/ST_GeomCollFromText.html) — Verilen SRID ile WKT koleksiyonundan bir Geometri koleksiyonu yapar. SRID verilmezse, varsayılan olarak 0 olur.

[ST\_GeomFromEWKT](https://postgis.net/docs/ST_GeomFromEWKT.html) — Genişletilmiş İyi Bilinen Metin gösteriminden (EWKT) belirtilen bir ST\_Geometry değeri döndürür.

[ST\_GeomFromMARC21](https://postgis.net/docs/ST_GeomFromMARC21.html) — MARC21/XML coğrafi verilerini girdi olarak alır ve bir PostGIS geometri nesnesi döndürür.

[ST\_GeometryFromText](https://postgis.net/docs/ST_GeometryFromText.html) — İyi Bilinen Metin gösteriminden (WKT) belirtilen bir ST\_Geometry değeri döndürür. Bu, ST\_GeomFromText için bir takma addır

[ST\_GeomFromText](https://postgis.net/docs/ST_GeomFromText.html) — İyi Bilinen Metin gösteriminden (WKT) belirtilen bir ST\_Geometry değeri döndürür.

[ST\_LineFromText](https://postgis.net/docs/ST_LineFromText.html) — Verilen SRID ile WKT temsilinden bir Geometri oluşturur. SRID verilmezse, varsayılan olarak 0 olur.

[ST\_MLineFromText](https://postgis.net/docs/ST_MLineFromText.html) — WKT gösteriminden belirtilen bir ST\_MultiLineString değeri döndürür.

[ST\_MPointFromText](https://postgis.net/docs/ST_MPointFromText.html) — Verilen SRID ile WKT'den bir Geometri oluşturur. SRID verilmezse, varsayılan olarak 0 olur.

[ST\_MPolyFromText](https://postgis.net/docs/ST_MPolyFromText.html) — Verilen SRID ile WKT'den bir MultiPolygon Geometrisi oluşturur. SRID verilmezse, varsayılan olarak 0 olur.

[ST\_PointFromText](https://postgis.net/docs/ST_PointFromText.html) — Verilen SRID ile WKT'den bir nokta Geometrisi yapar. SRID verilmezse, varsayılan olarak bilinmeyen olur.

[ST\_PolygonFromText](https://postgis.net/docs/ST_PolygonFromText.html) — Verilen SRID ile WKT'den bir Geometri oluşturur. SRID verilmezse, varsayılan olarak 0 olur.

[ST\_WKTToSQL](https://postgis.net/docs/ST_WKTToSQL.html) — İyi Bilinen Metin gösteriminden (WKT) belirtilen bir ST\_Geometry değeri döndürür. Bu, ST\_GeomFromText için bir takma addır

### İyi Bilinen İkili Dosya (WKB)

[ST\_GeogFromWKB](https://postgis.net/docs/ST_GeogFromWKB.html) — İyi Bilinen İkili geometri temsilinden (WKB) veya genişletilmiş İyi Bilinen İkiliden (EWKB) bir coğrafya örneği oluşturur.

[ST\_GeomFromEWKB](https://postgis.net/docs/ST_GeomFromEWKB.html) — Genişletilmiş İyi Bilinen İkili gösterimden (EWKB) belirtilen bir ST\_Geometry değeri döndürür.

[ST\_GeomFromWKB](https://postgis.net/docs/ST_GeomFromWKB.html) — İyi Bilinen İkili geometri temsilinden (WKB) ve isteğe bağlı SRID'den bir geometri örneği oluşturur.

[ST\_LineFromWKB](https://postgis.net/docs/ST_LineFromWKB.html) — **LINESTRING**Verilen SRID ile WKB'den bir a yapar

[ST\_LinestringFromWKB](https://postgis.net/docs/ST_LinestringFromWKB.html) — Verilen SRID ile WKB'den bir geometri oluşturur.

[ST\_PointFromWKB](https://postgis.net/docs/ST_PointFromWKB.html) — Verilen SRID ile WKB'den bir geometri oluşturur

[ST\_WKBToSQL](https://postgis.net/docs/ST_WKBToSQL.html) — İyi Bilinen İkili gösterimden (WKB) belirtilen bir ST\_Geometry değeri döndürür. Bu, srid almayan ST\_GeomFromWKB için bir takma addır

### Diğer Formatlar

[ST\_Box2dFromGeoHash](https://postgis.net/docs/ST_Box2dFromGeoHash.html) — GeoHash dizesinden bir BOX2D döndürür.

[ST\_GeomFromGeoHash](https://postgis.net/docs/ST_GeomFromGeoHash.html) — GeoHash dizesinden bir geometri döndürür.

[ST\_GeomFromGML](https://postgis.net/docs/ST_GeomFromGML.html) — Geometrinin GML gösterimini girdi olarak alır ve bir PostGIS geometri nesnesi çıkarır

[ST\_GeomFromGeoJSON](https://postgis.net/docs/ST_GeomFromGeoJSON.html) — Girdi olarak bir geometrinin geojson temsilini alır ve bir PostGIS geometri nesnesi çıktısı verir

[ST\_GeomFromKML](https://postgis.net/docs/ST_GeomFromKML.html) — Girdi olarak geometrinin KML temsilini alır ve bir PostGIS geometri nesnesi çıktısı verir

[ST\_GeomFromTWKB](https://postgis.net/docs/ST_GeomFromTWKB.html) — Bir TWKB ("Tiny Well-Known Binary") geometri temsilinden bir geometri örneği oluşturur.

[ST\_GMLToSQL](https://postgis.net/docs/ST_GMLToSQL.html) — GML gösteriminden belirtilen bir ST\_Geometry değeri döndürür. Bu, ST\_GeomFromGML için bir takma addır

[ST\_LineFromEncodedPolyline](https://postgis.net/docs/ST_LineFromEncodedPolyline.html) — Kodlanmış Bir Çoklu Çizgiden bir LineString oluşturur.

[ST\_PointFromGeoHash](https://postgis.net/docs/ST_PointFromGeoHash.html) — GeoHash dizesinden bir nokta döndürür.

[ST\_FromFlatGeobufToTable](https://postgis.net/docs/ST_FromFlatGeobufToTable.html) — FlatGeobuf verilerinin yapısını temel alan bir tablo oluşturur.

[ST\_FromFlatGeobuf](https://postgis.net/docs/ST_FromFlatGeobuf.html) — FlatGeobuf verilerini okur.

## Geometri Çıkışı

Bu işlevler, geometri nesnelerini çeşitli metinsel veya ikili biçimlere dönüştürür.

[Tanınmış Metin (WKT)](https://postgis.net/docs/reference.html#idm9594)

[İyi Bilinen İkili Dosya (WKB)](https://postgis.net/docs/reference.html#idm9762)

[Diğer Formatlar](https://postgis.net/docs/reference.html#idm9952)

### Tanınmış Metin (WKT)

[ST\_AsEWKT](https://postgis.net/docs/ST_AsEWKT.html) — SRID meta verileriyle geometrinin İyi Bilinen Metin (WKT) temsilini döndürür.

[ST\_AsText](https://postgis.net/docs/ST_AsText.html) — SRID meta verileri olmadan geometri/coğrafyanın İyi Bilinen Metin (WKT) temsilini döndürür.

### İyi Bilinen İkili Dosya (WKB)

[ST\_AsBinary](https://postgis.net/docs/ST_AsBinary.html) — SRID meta verileri olmadan geometrinin/coğrafyanın OGC/ISO İyi Bilinen İkili (WKB) temsilini döndürür.

[ST\_AsEWKB](https://postgis.net/docs/ST_AsEWKB.html) — SRID meta verileriyle geometrinin Genişletilmiş İyi Bilinen İkili (EWKB) temsilini döndürür.

[ST\_AsHEXEWKB](https://postgis.net/docs/ST_AsHEXEWKB.html) — Little-endian (NDR) veya big-endian (XDR) kodlamasını kullanarak HEXEWKB biçiminde (metin olarak) bir Geometri döndürür.

### Diğer Formatlar

[ST\_AsEncodedPolyline](https://postgis.net/docs/ST_AsEncodedPolyline.html) — LineString geometrisinden bir Kodlanmış Çoklu Çizgi döndürür.

[ST\_AsFlatGeobuf](https://postgis.net/docs/ST_AsFlatGeobuf.html) — Bir dizi satırın FlatGeobuf temsilini döndürür.

[ST\_AsGeobuf](https://postgis.net/docs/ST_AsGeobuf.html) — Bir dizi satırın Geobuf temsilini döndürür.

[ST\_AsGeoJSON](https://postgis.net/docs/ST_AsGeoJSON.html) — GeoJSON öğesi olarak bir geometri döndürür.

[ST\_AsGML](https://postgis.net/docs/ST_AsGML.html) — Geometriyi GML sürüm 2 veya 3 öğesi olarak döndürür.

[ST\_AsKML](https://postgis.net/docs/ST_AsKML.html) — Geometriyi bir KML öğesi olarak döndürür.

[ST\_AsLatLonText](https://postgis.net/docs/ST_AsLatLonText.html) — Verilen noktanın Derece, Dakika, Saniye temsilini döndürür.

[ST\_AsMARC21](https://postgis.net/docs/ST_AsMARC21.html) — Geometriyi, coğrafi veri alanı (034) ile bir MARC21/XML kaydı olarak döndürür.

[ST\_AsMVTGeom](https://postgis.net/docs/ST_AsMVTGeom.html) — Bir geometriyi bir MVT döşemesinin koordinat uzayına dönüştürür.

[ST\_AsMVT](https://postgis.net/docs/ST_AsMVT.html) — Bir dizi satırın MVT temsilini döndüren toplama işlevi.

[ST\_AsSVG](https://postgis.net/docs/ST_AsSVG.html) — Bir geometri için SVG yol verilerini döndürür.

[ST\_AsTWKB](https://postgis.net/docs/ST_AsTWKB.html) — Geometriyi TWKB, namı diğer "Küçük İyi Bilinen İkili" olarak döndürür

[ST\_AsX3D](https://postgis.net/docs/ST_AsX3D.html) — X3D xml düğüm öğesi biçiminde bir Geometri döndürür: ISO-IEC-19776-1.2-X3DEncodings-XML

[ST\_GeoHash](https://postgis.net/docs/ST_GeoHash.html) — Geometrinin GeoHash temsilini döndürür.

## operatörler

[8.10.1. Sınırlayıcı Kutu Operatörleri](https://postgis.net/docs/reference.html#operators-bbox)

[8.10.2. Mesafe Operatörleri](https://postgis.net/docs/reference.html#operators-distance)

### Sınırlayıcı Kutu Operatörleri

[&&](https://postgis.net/docs/geometry_overlaps.html) — A'nın 2B sınırlayıcı kutusu B'nin 2B sınırlayıcı kutusuyla kesişirse döndürür .**TRUE**

[&&(geometry,box2df)](https://postgis.net/docs/overlaps_geometry_box2df.html) — **TRUE**Bir geometrinin (önbelleğe alınmış) 2B sınırlama kutusu bir 2B kayan nokta hassas sınırlama kutusuyla (BOX2DF) kesişirse döndürür.

[&&(box2df,geometry)](https://postgis.net/docs/overlaps_box2df_geometry.html) — **TRUE**2B kayan nokta hassas sınırlama kutusu (BOX2DF) bir geometrinin (önbelleğe alınmış) 2B sınırlama kutusuyla kesişirse döndürür.

[&&(box2df,box2df)](https://postgis.net/docs/overlaps_box2df_box2df.html) — **TRUE**İki 2B kayan nokta hassas sınırlayıcı kutu (BOX2DF) birbiriyle kesişirse döndürür.

[&&&](https://postgis.net/docs/geometry_overlaps_nd.html) — A'nın nD sınırlayıcı kutusu B'nin nD sınırlayıcı kutusuyla kesişirse döndürür .**TRUE**

[&&&(geometry,gidx)](https://postgis.net/docs/overlaps_nd_geometry_gidx.html) — **TRUE**Bir geometrinin (önbelleğe alınmış) nD sınırlayıcı kutusu bir nD kayan noktalı hassas sınırlayıcı kutu (GIDX) ile kesişirse döndürür.

[&&&(gidx,geometry)](https://postgis.net/docs/overlaps_nd_gidx_geometry.html) — **TRUE**Bir nD kayan nokta hassas sınırlama kutusu (GIDX) bir geometrinin (önbelleğe alınmış) nD sınırlama kutusuyla kesişirse döndürür.

[&&&(gidx,gidx)](https://postgis.net/docs/overlaps_nd_gidx_gidx.html) — **TRUE**İki nD kayan nokta hassas sınırlayıcı kutu (GIDX) birbiriyle kesişirse döndürür.

[&<](https://postgis.net/docs/ST_Geometry_Overleft.html) — A'nın sınırlayıcı kutusu örtüşüyorsa veya B'nin solundaysa döndürür .**TRUE**

[&<|](https://postgis.net/docs/ST_Geometry_Overbelow.html)— **TRUE**A'nın sınırlayıcı kutusu B'nin sınırlayıcı kutusuyla örtüşüyorsa veya altındaysa döner.

[&>](https://postgis.net/docs/ST_Geometry_Overright.html) — A' sınırlayıcı kutusu örtüşüyorsa veya B'lerin sağındaysa döndürür .**TRUE**

[<<](https://postgis.net/docs/ST_Geometry_Left.html) — **TRUE**A'nın sınırlayıcı kutusu kesinlikle B'nin solundaysa döner.

[<<|](https://postgis.net/docs/ST_Geometry_Below.html)— **TRUE**A'nın sınırlayıcı kutusu kesinlikle B'nin altındaysa döner.

[=](https://postgis.net/docs/ST_Geometry_EQ.html) — Koordinatlar ve koordinat düzeni geometri/coğrafya A, geometri/coğrafya B'nin koordinatları ve koordinat sırası ile aynıysa döndürür .**TRUE**

[>>](https://postgis.net/docs/ST_Geometry_Right.html) — **TRUE**A'nın sınırlayıcı kutusu kesinlikle B'nin sağındaysa döner.

[@](https://postgis.net/docs/ST_Geometry_Contained.html) — **TRUE**A'nın sınırlayıcı kutusunun B'ler tarafından içerilip içermediğini döndürür.

[@(geometry,box2df)](https://postgis.net/docs/is_contained_geometry_box2df.html) — **TRUE**Bir geometrinin 2B sınırlayıcı kutusunun bir 2B kayan nokta hassas sınırlayıcı kutusu (BOX2DF) içinde olup olmadığını döndürür.

[@(box2df,geometry)](https://postgis.net/docs/is_contained_box2df_geometry.html) — **TRUE**Bir geometrinin 2B sınırlayıcı kutusunda bir 2B kayan nokta hassas sınırlama kutusu (BOX2DF) varsa döner.

[@(box2df,box2df)](https://postgis.net/docs/is_contained_box2df_box2df.html) — **TRUE**Bir 2B kayan nokta duyarlık sınırlama kutusu (BOX2DF) başka bir 2B kayan nokta duyarlık sınırlayıcı kutu içinde yer alıyorsa döndürür.

[|&>](https://postgis.net/docs/ST_Geometry_Overabove.html) — A'nın sınırlayıcı kutusu B'nin üzerindeyse veya örtüşüyorsa döndürür .**TRUE**

[|>>](https://postgis.net/docs/ST_Geometry_Above.html) — **TRUE**A'nın sınırlayıcı kutusu kesinlikle B'nin üzerindeyse döner.

[~](https://postgis.net/docs/ST_Geometry_Contain.html) — **TRUE**A'nın sınırlayıcı kutusu B'leri içeriyorsa döndürür.

[~(geometry,box2df)](https://postgis.net/docs/contains_geometry_box2df.html) — **TRUE**Bir geometrinin 2B bağlama kutusu bir 2B kayan nokta hassas sınırlama kutusu (GIDX) içeriyorsa döndürür.

[~(box2df,geometry)](https://postgis.net/docs/contains_box2df_geometry.html) — **TRUE**Bir 2B kayan nokta hassas sınırlama kutusu (BOX2DF) bir geometrinin 2B bağlama kutusunu içeriyorsa döndürür.

[~(box2df,box2df)](https://postgis.net/docs/contains_box2df_box2df.html) — **TRUE**Bir 2B kayan nokta duyarlık sınırlama kutusu (BOX2DF) başka bir 2B kayan nokta duyarlık sınırlayıcı kutu (BOX2DF) içeriyorsa döndürür.

[~=](https://postgis.net/docs/ST_Geometry_Same.html) — **TRUE**A'nın sınırlayıcı kutusu B'ninkiyle aynıysa döndürür.

### Mesafe Operatörleri

[<->](https://postgis.net/docs/geometry_distance_knn.html) — A ve B arasındaki 2B mesafeyi döndürür.

[|=|](https://postgis.net/docs/geometry_distance_cpa.html)— En yakın yaklaşma noktalarında A ve B yörüngeleri arasındaki mesafeyi döndürür.

[<#>](https://postgis.net/docs/geometry_distance_box.html) — A ve B sınırlayıcı kutular arasındaki 2B mesafeyi döndürür.

[<<->>](https://postgis.net/docs/geometry_distance_centroid_nd.html) — A ve B sınırlama kutularının ağırlık merkezleri arasındaki nD mesafesini döndürür.

[<<#>>](https://postgis.net/docs/geometry_distance_box_nd.html) — A ve B sınırlayıcı kutular arasındaki nD mesafesini döndürür.

## Mekansal ilişkiler

Bu işlevler, geometriler arasındaki uzamsal ilişkileri belirler.

[8.11.1. Topolojik İlişkiler](https://postgis.net/docs/reference.html#idm12212)

[8.11.2. Mesafe İlişkileri](https://postgis.net/docs/reference.html#idm13387)

### Topolojik İlişkiler

[ST\_3DIntersects](https://postgis.net/docs/ST_3DIntersects.html) — İki geometrinin 3B'de uzamsal olarak kesişip kesişmediğini test eder - yalnızca noktalar, çizgi dizileri, çokgenler, çokyüzlü yüzey (alan) için.

[ST\_Contains](https://postgis.net/docs/ST_Contains.html) — B'nin hiçbir noktasının A'nın dışında olup olmadığını ve A ile B'nin ortak en az bir iç noktasına sahip olup olmadığını test eder.

[ST\_ContainsProperly](https://postgis.net/docs/ST_ContainsProperly.html) — B'nin A'nın içini kesip kesmediğini, ancak sınır veya dışıyla kesişmediğini test eder.

[ST\_CoveredBy](https://postgis.net/docs/ST_CoveredBy.html) — A'daki hiçbir noktanın B'nin dışında olup olmadığını test eder

[ST\_Covers](https://postgis.net/docs/ST_Covers.html) — B'deki hiçbir noktanın A dışında olup olmadığını test eder

[ST\_Crosses](https://postgis.net/docs/ST_Crosses.html) — İki geometrinin bazı iç noktalarının ortak olup olmadığını test eder.

[ST\_Disjoint](https://postgis.net/docs/ST_Disjoint.html) — İki geometrinin ayrık olup olmadığını test eder (ortak noktaları yoktur).

[ST\_Equals](https://postgis.net/docs/ST_Equals.html) — İki geometrinin aynı nokta kümesini içerip içermediğini test eder.

[ST\_Intersects](https://postgis.net/docs/ST_Intersects.html) — İki geometrinin kesişip kesişmediğini test eder (en az bir ortak noktaları vardır).

[ST\_LineCrossingDirection](https://postgis.net/docs/ST_LineCrossingDirection.html) — İki LineString'in geçiş davranışını gösteren bir sayı döndürür.

[ST\_OrderingEquals](https://postgis.net/docs/ST_OrderingEquals.html) — İki geometrinin aynı geometriyi temsil edip etmediğini ve aynı yön düzeninde noktalara sahip olup olmadığını test eder.

[ST\_Overlaps](https://postgis.net/docs/ST_Overlaps.html) — İki geometrinin kesişip kesişmediğini ve aynı boyuta sahip olup olmadığını ancak birbirleri tarafından tamamen kapsanmadığını test eder.

[ST\_Relate](https://postgis.net/docs/ST_Relate.html) — İki geometrinin bir Kesişim Matrisi modeliyle eşleşen topolojik bir ilişkisi olup olmadığını test eder veya Kesişim Matrisini hesaplar

[ST\_RelateMatch](https://postgis.net/docs/ST_RelateMatch.html) — DE-9IM Kavşak Matrisinin bir Kavşak Matrisi modeliyle eşleşip eşleşmediğini test eder

[ST\_Touches](https://postgis.net/docs/ST_Touches.html) — İki geometrinin en az bir ortak noktası olup olmadığını, ancak içlerinin kesişmediğini test eder.

[ST\_Within](https://postgis.net/docs/ST_Within.html) — A'nın hiçbir noktasının B'nin dışında olup olmadığını ve A ve B'nin ortak en az bir iç noktasına sahip olup olmadığını test eder.

### Mesafe İlişkileri

[ST\_3DDWithin](https://postgis.net/docs/ST_3DDWithin.html) — İki 3B geometrinin belirli bir 3B mesafe içinde olup olmadığını test eder

[ST\_3DDFullyWithin](https://postgis.net/docs/ST_3DDFullyWithin.html) — İki 3B geometrinin tamamen belirli bir 3B mesafede olup olmadığını test eder

[ST\_DFullyWithin](https://postgis.net/docs/ST_DFullyWithin.html) — İki geometrinin tamamen belirli bir mesafede olup olmadığını test eder

[ST\_DWithin](https://postgis.net/docs/ST_DWithin.html) — İki geometrinin belirli bir mesafede olup olmadığını test eder

[ST\_PointInsideCircle](https://postgis.net/docs/ST_PointInsideCircle.html) — Nokta geometrisinin bir merkez ve yarıçap tarafından tanımlanan bir daire içinde olup olmadığını test eder.

## Ölçüm Fonksiyonları

Bu işlevler mesafe, alan ve açı ölçümlerini hesaplar. Ölçümlerle belirlenen geometri değerlerini hesaplamak için fonksiyonlar da vardır.

[ST\_Area](https://postgis.net/docs/ST_Area.html) — Çokgen geometrinin alanını döndürür.

[ST\_Azimuth](https://postgis.net/docs/ST_Azimuth.html) — İki nokta arasındaki bir çizginin kuzey tabanlı azimutunu döndürür.

[ST\_Angle](https://postgis.net/docs/ST_Angle.html) — 3 veya 4 nokta veya 2 çizgi ile tanımlanan iki vektör arasındaki açıyı döndürür.

[ST\_ClosestPoint](https://postgis.net/docs/ST_ClosestPoint.html) — g2'ye en yakın olan g1 üzerindeki 2B noktayı döndürür. Bu, bir geometriden diğerine giden en kısa çizginin ilk noktasıdır.

[ST\_3DClosestPoint](https://postgis.net/docs/ST_3DClosestPoint.html) — g2'ye en yakın olan g1 üzerindeki 3B noktayı döndürür. Bu, 3B en kısa çizginin ilk noktasıdır.

[ST\_Distance](https://postgis.net/docs/ST_Distance.html) — İki geometri veya coğrafya değeri arasındaki mesafeyi döndürür.

[ST\_3DDistance](https://postgis.net/docs/ST_3DDistance.html) — Yansıtılan birimlerde iki geometri arasındaki 3B kartezyen minimum mesafeyi (uzaysal referansa dayalı olarak) döndürür.

[ST\_DistanceSphere](https://postgis.net/docs/ST_DistanceSphere.html) — Küresel bir dünya modeli kullanarak iki boylam/enlem geometrisi arasındaki metre cinsinden minimum mesafeyi döndürür.

[ST\_DistanceSpheroid](https://postgis.net/docs/ST_Distance_Spheroid.html) — Küresel bir dünya modeli kullanarak iki boylam/enlem geometrisi arasındaki minimum mesafeyi döndürür.

[ST\_FrechetDistance](https://postgis.net/docs/ST_FrechetDistance.html) — İki geometri arasındaki Fréchet mesafesini döndürür.

[ST\_HausdorffDistance](https://postgis.net/docs/ST_HausdorffDistance.html) — İki geometri arasındaki Hausdorff mesafesini döndürür.

[ST\_Length](https://postgis.net/docs/ST_Length.html) — Doğrusal bir geometrinin 2B uzunluğunu döndürür.

[ST\_Length2D](https://postgis.net/docs/ST_Length2D.html) — Doğrusal bir geometrinin 2B uzunluğunu döndürür. takma ad**ST\_Length**

[ST\_3DLength](https://postgis.net/docs/ST_3DLength.html) — Doğrusal bir geometrinin 3B uzunluğunu döndürür.

[ST\_LengthSpheroid](https://postgis.net/docs/ST_Length_Spheroid.html) — Bir sferoid üzerindeki bir boylam/enlem geometrisinin 2B veya 3B uzunluğunu/çevresini döndürür.

[ST\_LongestLine](https://postgis.net/docs/ST_LongestLine.html) — İki geometri arasındaki en uzun 2B çizgiyi döndürür.

[ST\_3DlongestLine](https://postgis.net/docs/ST_3DLongestLine.html) — İki geometri arasındaki en uzun 3B çizgiyi döndürür

[ST\_MaxDistance](https://postgis.net/docs/ST_MaxDistance.html) — Yansıtılan birimlerde iki geometri arasındaki en büyük 2B mesafeyi döndürür.

[ST\_3DMaxDistance](https://postgis.net/docs/ST_3DMaxDistance.html) — Yansıtılan birimlerde iki geometri arasındaki 3B kartezyen maksimum mesafeyi (uzaysal referansa dayalı olarak) döndürür.

[ST\_MinimumClearance](https://postgis.net/docs/ST_MinimumClearance.html) — Bir geometrinin sağlamlığının bir ölçüsü olan bir geometrinin minimum açıklığını döndürür.

[ST\_MinimumClearanceLine](https://postgis.net/docs/ST_MinimumClearanceLine.html) — Bir geometrinin minimum boşluğunu kapsayan iki noktalı LineString'i döndürür.

[ST\_Perimeter](https://postgis.net/docs/ST_Perimeter.html) — Çokgen geometri veya coğrafyanın sınırının uzunluğunu döndürür.

[ST\_Perimeter2D](https://postgis.net/docs/ST_Perimeter2D.html) — Çokgen geometrinin 2B çevresini döndürür. için takma ad **ST\_Perimeter**.

[ST\_3DPerimeter](https://postgis.net/docs/ST_3DPerimeter.html) — Çokgen geometrinin 3B çevresini döndürür.

[ST\_Project](https://postgis.net/docs/ST_Project.html) — Bir başlangıç ​​noktasından bir mesafe ve yön (azimut) ile yansıtılan bir nokta döndürür.

[ST\_ShortestLine](https://postgis.net/docs/ST_ShortestLine.html) — İki geometri arasındaki en kısa 2B çizgiyi döndürür

[ST\_3DShortestLine](https://postgis.net/docs/ST_3DShortestLine.html) — İki geometri arasındaki en kısa 3B çizgiyi döndürür

## Bindirme Fonksiyonları

Bu işlevler, iki geometrinin üst üste bindirilmesinden kaynaklanan sonuçları hesaplar. Bunlar ayrıca nokta kümeli teorik boole işlemleri olarak da bilinir. Bazı ilgili işlevler de sağlanmaktadır.

[ST\_ClipByBox2D](https://postgis.net/docs/ST_ClipByBox2D.html) — Geometrinin bir dikdörtgenin içine düşen kısmını hesaplar.

[ST\_Difference](https://postgis.net/docs/ST_Difference.html) — A geometrisinin B geometrisiyle kesişmeyen kısmını temsil eden bir geometri hesaplar.

[ST\_Intersection](https://postgis.net/docs/ST_Intersection.html) — A ve B geometrilerinin paylaşılan bölümünü temsil eden bir geometri hesaplar.

[ST\_MemUnion](https://postgis.net/docs/ST_MemUnion.html) — Geometrileri bellek açısından verimli ancak daha yavaş bir şekilde birleştiren toplama işlevi

[ST\_Node](https://postgis.net/docs/ST_Node.html) — Bir satır koleksiyonunu düğümler.

[ST\_Split](https://postgis.net/docs/ST_Split.html) — Bir geometriyi başka bir geometriye bölerek oluşturulan bir geometri koleksiyonu döndürür.

[ST\_Subdivide](https://postgis.net/docs/ST_Subdivide.html) — Bir geometrinin doğrusal bir alt bölümünü hesaplar.

[ST\_SymDifference](https://postgis.net/docs/ST_SymDifference.html) — A ve B geometrilerinin kesişmeyen kısımlarını temsil eden bir geometri hesaplar.

[ST\_UnaryUnion](https://postgis.net/docs/ST_UnaryUnion.html) — Tek bir geometrinin bileşenlerinin birleşimini hesaplar.

[ST\_Union](https://postgis.net/docs/ST_Union.html) — Giriş geometrilerinin nokta kümesi birleşimini temsil eden bir geometri hesaplar.

## Geometri İşleme

Bu işlevler geometrik yapıları hesaplar veya geometri boyutunu veya şeklini değiştirir.

[ST\_Buffer](https://postgis.net/docs/ST_Buffer.html) — Bir geometriden belirli bir mesafedeki tüm noktaları kapsayan bir geometri hesaplar.

[ST\_BuildArea](https://postgis.net/docs/ST_BuildArea.html) — Bir geometrinin çizgisinden oluşan çokgen bir geometri oluşturur.

[ST\_Centroid](https://postgis.net/docs/ST_Centroid.html) — Bir geometrinin geometrik merkezini döndürür.

[ST\_ChaikinSmoothing](https://postgis.net/docs/ST_ChaikinSmoothing.html) — Chaikin algoritmasını kullanarak bir geometrinin düzleştirilmiş bir sürümünü döndürür

[ST\_ConcaveHull](https://postgis.net/docs/ST_ConcaveHull.html) — Tüm giriş geometrisi köşelerini kapsayan olası bir içbükey geometriyi hesaplar

[ST\_ConvexHull](https://postgis.net/docs/ST_ConvexHull.html) — Bir geometrinin dışbükey gövdesini hesaplar.

[ST\_DelaunayTriangles](https://postgis.net/docs/ST_DelaunayTriangles.html) — Bir geometrinin köşelerinin Delaunay üçgenlemesini döndürür.

[ST\_FilterByM](https://postgis.net/docs/ST_FilterByM.html) — M değerlerine göre köşeleri kaldırır

[ST\_GeneratePoints](https://postgis.net/docs/ST_GeneratePoints.html) — Bir Poligon veya MultiPolygon içinde bulunan rastgele noktalar üretir.

[ST\_GeometricMedian](https://postgis.net/docs/ST_GeometricMedian.html)— Bir MultiPoint'in geometrik medyanını döndürür.

[ST\_LineMerge](https://postgis.net/docs/ST_LineMerge.html) — Bir MultiLineString'i birbirine dikerek oluşturulan çizgileri döndürür.

[ST\_MaximumInscribedCircle](https://postgis.net/docs/ST_MaximumInscribedCircle.html) — Bir geometride bulunan en büyük daireyi hesaplar.

[ST\_MinimumBoundingCircle](https://postgis.net/docs/ST_MinimumBoundingCircle.html) — Bir geometri içeren en küçük daire poligonunu döndürür.

[ST\_MinimumBoundingRadius](https://postgis.net/docs/ST_MinimumBoundingRadius.html) — Geometri içeren en küçük dairenin merkez noktasını ve yarıçapını döndürür.

[ST\_OrientedEnvelope](https://postgis.net/docs/ST_OrientedEnvelope.html) — Bir geometri içeren minimum alanlı bir dikdörtgen döndürür.

[ST\_OffsetCurve](https://postgis.net/docs/ST_OffsetCurve.html) — Bir giriş satırından belirli bir mesafede ve kenarda bir ofset çizgisi döndürür.

[ST\_PointOnSurface](https://postgis.net/docs/ST_PointOnSurface.html) — Bir poligonda veya bir geometride olması garanti edilen bir noktayı hesaplar.

[ST\_Polygonize](https://postgis.net/docs/ST_Polygonize.html) — Bir dizi geometrinin çizgisinden oluşturulmuş bir çokgen koleksiyonunu hesaplar.

[ST\_ReducePrecision](https://postgis.net/docs/ST_ReducePrecision.html) — Noktaları bir ızgara toleransına yuvarlatılmış geçerli bir geometri döndürür.

[ST\_SharedPaths](https://postgis.net/docs/ST_SharedPaths.html) — İki giriş satır dizisi/çok satırlı dize tarafından paylaşılan yolları içeren bir koleksiyon döndürür.

[ST\_Simplify](https://postgis.net/docs/ST_Simplify.html) — Douglas-Peucker algoritmasını kullanarak bir geometrinin basitleştirilmiş bir sürümünü döndürür.

[ST\_SimplifyPreserveTopology](https://postgis.net/docs/ST_SimplifyPreserveTopology.html) — Douglas-Peucker algoritmasını kullanarak bir geometrinin basitleştirilmiş ve geçerli bir sürümünü döndürür.

[ST\_SimplifyPolygonHull](https://postgis.net/docs/ST_SimplifyPolygonHull.html) — Bir poligonal geometrinin dış veya iç gövdesini koruyan basitleştirilmiş bir topolojiyi hesaplar.

[ST\_SimplifyVW](https://postgis.net/docs/ST_SimplifyVW.html) — Visvalingam- Whyatt algoritmasını kullanarak bir geometrinin basitleştirilmiş bir sürümünü döndürür

[ST\_SetEffectiveArea](https://postgis.net/docs/ST_SetEffectiveArea.html) — Visvalingam- Whyatt algoritmasını kullanarak her köşe için etkili alanı ayarlar.

[ST\_TriangulatePolygon](https://postgis.net/docs/ST_TriangulatePolygon.html) — Çokgenlerin kısıtlı Delaunay üçgenlemesini hesaplar

[ST\_VoronoiLines](https://postgis.net/docs/ST_VoronoiLines.html) — Bir geometrinin köşelerinin Voronoi diyagramının sınırlarını döndürür.

[ST\_VoronoiPolygons](https://postgis.net/docs/ST_VoronoiPolygons.html) — Bir geometrinin köşelerinin Voronoi diyagramının hücrelerini döndürür.

## Afin Dönüşümler

[Bu işlevler, afin dönüşümler](https://en.wikipedia.org/wiki/Affine_transformation) kullanarak geometrilerin konumunu ve şeklini değiştirir .

[ST\_Affine](https://postgis.net/docs/ST_Affine.html) — Bir geometriye 3B afin dönüşümü uygulayın.

[ST\_Rotate](https://postgis.net/docs/ST_Rotate.html) — Geometriyi bir başlangıç ​​noktası etrafında döndürür.

[ST\_RotateX](https://postgis.net/docs/ST_RotateX.html) — Bir geometriyi X ekseni etrafında döndürür.

[ST\_RotateY](https://postgis.net/docs/ST_RotateY.html) — Bir geometriyi Y ekseni etrafında döndürür.

[ST\_RotateZ](https://postgis.net/docs/ST_RotateZ.html) — Bir geometriyi Z ekseni etrafında döndürür.

[ST\_Scale](https://postgis.net/docs/ST_Scale.html) — Verilen faktörlere göre bir geometriyi ölçekler.

[ST\_Translate](https://postgis.net/docs/ST_Translate.html) — Verilen ofsetlere göre bir geometriyi öteler.

[ST\_TransScale](https://postgis.net/docs/ST_TransScale.html) — Verilen ötelemelere ve faktörlere göre bir geometriyi öteler ve ölçekler.

## Kümeleme İşlevleri

Bu işlevler, geometri kümeleri için kümeleme algoritmaları uygular.

[ST\_ClusterDBSCAN](https://postgis.net/docs/ST_ClusterDBSCAN.html) — DBSCAN algoritmasını kullanarak her giriş geometrisi için bir küme kimliği döndüren pencere işlevi.

[ST\_ClusterIntersecting](https://postgis.net/docs/ST_ClusterIntersecting.html) — Giriş geometrilerini bağlı kümeler halinde kümeleyen toplama işlevi.

[ST\_ClusterKMeans](https://postgis.net/docs/ST_ClusterKMeans.html) — K-ortalama algoritmasını kullanarak her giriş geometrisi için bir küme kimliği döndüren pencere işlevi.

[ST\_ClusterWithin](https://postgis.net/docs/ST_ClusterWithin.html) — Giriş geometrilerini ayırma mesafesine göre kümeleyen toplama işlevi.

## Sınırlayıcı Kutu İşlevleri

Bu işlevler sınırlayıcı kutular üretir veya bunlar üzerinde çalışır. Ayrıca otomatik veya açık dökümler kullanarak geometri değerleri sağlayabilir ve kabul edebilirler.

Ayrıca bkz. [Bölüm 15.7, “PostGIS Box Fonksiyonları”](https://postgis.net/docs/PostGIS_Special_Functions_Index.html#PostGIS_BoxFunctions) .

[Box2D](https://postgis.net/docs/Box2D.html) — Bir geometrinin 2B kapsamını temsil eden bir BOX2D döndürür.

[Box3D](https://postgis.net/docs/Box3D.html) — Bir geometrinin 3B kapsamını temsil eden bir BOX3D döndürür.

[ST\_EstimatedExtent](https://postgis.net/docs/ST_EstimatedExtent.html) — Bir uzamsal tablonun tahmini kapsamını döndürür.

[ST\_Expand](https://postgis.net/docs/ST_Expand.html) — Başka bir sınırlayıcı kutudan veya bir geometriden genişletilmiş bir sınırlayıcı kutu döndürür.

[ST\_Extent](https://postgis.net/docs/ST_Extent.html) — Sınırlayıcı geometri kutusunu döndüren toplama işlevi.

[ST\_3DExtent](https://postgis.net/docs/ST_3DExtent.html) — Geometrilerin 3B sınırlayıcı kutusunu döndüren toplama işlevi.

[ST\_MakeBox2D](https://postgis.net/docs/ST_MakeBox2D.html) — İki 2B nokta geometrisi tarafından tanımlanan bir BOX2D oluşturur.

[ST\_3DMakeBox](https://postgis.net/docs/ST_3DMakeBox.html) — İki 3B nokta geometrisi tarafından tanımlanan bir BOX3D oluşturur.

[ST\_XMax](https://postgis.net/docs/ST_XMax.html) — 2B veya 3B sınırlama kutusunun veya geometrinin X maksimumunu döndürür.

[ST\_XMin](https://postgis.net/docs/ST_XMin.html) — 2B veya 3B sınırlayıcı kutu veya geometrinin X minimumunu döndürür.

[ST\_YMax](https://postgis.net/docs/ST_YMax.html) — 2B veya 3B sınırlama kutusunun veya geometrinin Y maksimumunu döndürür.

[ST\_YMin](https://postgis.net/docs/ST_YMin.html) — 2B veya 3B sınırlama kutusunun veya geometrinin Y minimumunu döndürür.

[ST\_ZMax](https://postgis.net/docs/ST_ZMax.html) — 2B veya 3B sınırlama kutusunun veya geometrinin Z maksimumunu döndürür.

[ST\_ZMin](https://postgis.net/docs/ST_ZMin.html) — 2B veya 3B sınırlama kutusunun veya geometrinin Z minimumunu döndürür.

## Doğrusal Referanslama

[ST\_LineInterpolatePoint](https://postgis.net/docs/ST_LineInterpolatePoint.html) — Kesirli bir konumda bir çizgi boyunca enterpolasyon yapılmış bir nokta döndürür.

[ST\_3DLineInterpolatePoint](https://postgis.net/docs/ST_3DLineInterpolatePoint.html) — Kesirli bir konumda 3B çizgi boyunca enterpolasyon yapılmış bir nokta döndürür.

[ST\_LineInterpolatePoints](https://postgis.net/docs/ST_LineInterpolatePoints.html) — Kesirli bir aralıkta bir çizgi boyunca enterpolasyon yapılan noktaları döndürür.

[ST\_LineLocatePoint](https://postgis.net/docs/ST_LineLocatePoint.html) — Bir çizgideki bir noktaya en yakın noktanın kesirli konumunu döndürür.

[ST\_LineSubstring](https://postgis.net/docs/ST_LineSubstring.html) — Doğrunun iki kesirli konum arasındaki kısmını döndürür.

[ST\_LocateAlong](https://postgis.net/docs/ST_LocateAlong.html) — Bir geometri üzerindeki bir hesaplama değeriyle eşleşen noktaları döndürür.

[ST\_LocateBetween](https://postgis.net/docs/ST_LocateBetween.html) — Bir geometrinin ölçü aralığıyla eşleşen kısımlarını döndürür.

[ST\_LocateBetweenElevations](https://postgis.net/docs/ST_LocateBetweenElevations.html) — Bir yükseklik (Z) aralığında yer alan bir geometrinin bölümlerini döndürür.

[ST\_InterpolatePoint](https://postgis.net/docs/ST_InterpolatePoint.html) — Bir noktaya en yakın geometrinin enterpolasyonlu ölçüsünü döndürür.

[ST\_AddMeasure](https://postgis.net/docs/ST_AddMeasure.html) — Doğrusal bir geometri boyunca ölçümleri enterpolasyonlar.

## Yörünge Fonksiyonları

Bu işlevler yörüngelerle çalışmayı destekler. Bir yörünge, her koordinatta artan ölçüler (M değeri) olan doğrusal bir geometridir. Uzamsal-zamansal veriler, ölçü değerleri olarak göreli zamanlar (dönem gibi) kullanılarak modellenebilir.

[ST\_IsValidTrajectory](https://postgis.net/docs/ST_IsValidTrajectory.html) — Geometrinin geçerli bir yörünge olup olmadığını test eder.

[ST\_ClosestPointOfApproach](https://postgis.net/docs/ST_ClosestPointOfApproach.html) — İki yörüngenin en yakın yaklaşma noktasında bir ölçü döndürür.

[ST\_DistanceCPA](https://postgis.net/docs/ST_DistanceCPA.html) — İki yörüngenin en yakın yaklaşma noktası arasındaki mesafeyi döndürür.

[ST\_CPWithin](https://postgis.net/docs/ST_CPAWithin.html) — İki yörüngenin en yakın yaklaşma noktasının belirtilen mesafe içinde olup olmadığını test eder.

## SFCGAL Fonksiyonları

SFCGAL, gelişmiş 2B ve 3B uzamsal işlevler sağlayan CGAL çevresinde bir C++ sarmalayıcı kitaplığıdır. Sağlamlık için, geometri koordinatları tam bir rasyonel sayı temsiline sahiptir.

Kütüphane için kurulum talimatları SFCGAL ana sayfasında bulunabilir ( [http://www.sfcgal.org](http://www.sfcgal.org/) ). İşlevleri etkinleştirmek için kullanın **create extension postgis\_sfcgal**.

[postgis\_sfcgal\_version](https://postgis.net/docs/postgis_sfcgal_version.html) — Kullanımdaki SFCGAL sürümünü döndürür

[postgis\_sfcgal\_full\_version](https://postgis.net/docs/postgis_sfcgal_full_version.html) — CGAL ve Boost sürümleri dahil olmak üzere kullanımda olan SFCGAL'in tam sürümünü döndürür

[ST\_3DArea](https://postgis.net/docs/ST_3DArea.html) — 3B yüzey geometrilerinin alanını hesaplar. Katılar için 0 döndürür.

[ST\_3DConvexHull](https://postgis.net/docs/ST_3DConvexHull.html) — Bir geometrinin 3B dışbükey gövdesini hesaplar.

[ST\_3DIntersection](https://postgis.net/docs/ST_3DIntersection.html) — 3B kesişimi gerçekleştir

[ST\_3DDifference](https://postgis.net/docs/ST_3DDifference.html) — 3D farkı gerçekleştir

[ST\_3DUnion](https://postgis.net/docs/ST_3DUnion.html) — 3B birleştirme gerçekleştirin.

[ST\_AlphaShape](https://postgis.net/docs/ST_AlphaShape.html) — CGAL Alpha Shapes algoritmasını kullanarak olası bir içbükey geometriyi hesaplar.

[ST\_ApproximateMedialAxis](https://postgis.net/docs/ST_ApproximateMedialAxis.html) — Bir alan geometrisinin yaklaşık orta eksenini hesaplayın.

[ST\_ConstrainedDelaunayTriangles](https://postgis.net/docs/ST_ConstrainedDelaunayTriangles.html) — Verilen giriş geometrisi etrafında kısıtlı bir Delaunay üçgenlemesi döndürür.

[ST\_Extrude](https://postgis.net/docs/ST_Extrude.html) — Bir yüzeyi ilgili bir birime ekstrüde edin

[ST\_ForceLHR](https://postgis.net/docs/ST_ForceLHR.html) — LHR yönlendirmesini zorla

[ST\_IsPlanar](https://postgis.net/docs/ST_IsPlanar.html) — Bir yüzeyin düzlemsel olup olmadığını kontrol edin

[ST\_IsSolid](https://postgis.net/docs/ST_IsSolid.html) — Geometrinin katı olup olmadığını test edin. Geçerlilik kontrolü yapılmaz.

[ST\_MakeSolid](https://postgis.net/docs/ST_MakeSolid.html) — Geometriyi bir katıya dönüştürün. Herhangi bir kontrol yapılmamaktadır. Geçerli bir katı elde etmek için giriş geometrisi kapalı bir Çokyüzlü Yüzey veya kapalı bir TIN olmalıdır.

[ST\_MinkowskiSum](https://postgis.net/docs/ST_MinkowskiSum.html) — Minkowski toplamını gerçekleştirir

[ST\_OptimalAlphaShape](https://postgis.net/docs/ST_OptimalAlphaShape.html) — "Optimal" alfa değerini hesapladıktan sonra CGAL Alpha Shapes algoritmasını kullanarak olası bir içbükey geometriyi hesaplar.

[ST\_Orientation](https://postgis.net/docs/ST_Orientation.html) — Yüzey yönünü belirleyin

[ST\_StraightSkeleton](https://postgis.net/docs/ST_StraightSkeleton.html) — Bir geometriden düz bir iskelet hesaplayın

[ST\_Tesselate](https://postgis.net/docs/ST_Tesselate.html) — Bir çokgen veya çokyüzlü yüzeyin yüzey Mozaikasyonunu gerçekleştirin ve TIN veya TINS ​​koleksiyonu olarak geri dönün

[ST\_Volume](https://postgis.net/docs/ST_Volume.html) — 3B katının hacmini hesaplar. Yüzeye uygulanırsa (kapalı bile olsa) geometriler 0 döndürür.

## Uzun İşlem Desteği

Bu işlevler, uzun işlemleri desteklemek için bir satır kilitleme mekanizması uygular. Öncelikle [Web Özelliği Hizmeti](http://www.opengeospatial.org/standards/wfs) belirtiminin uygulayıcıları için sağlanırlar.

[AddAuth](https://postgis.net/docs/AddAuth.html) — Geçerli işlemde kullanılacak bir yetkilendirme belirteci ekler.

[CheckAuth](https://postgis.net/docs/CheckAuth.html) — Yetkilendirme belirtecine dayalı olarak satırların güncellenmesini ve silinmesini önlemek/izin vermek için bir tabloda bir tetikleyici oluşturur.

[DisableLongTransactions](https://postgis.net/docs/DisableLongTransactions.html) — Uzun işlem desteğini devre dışı bırakır.

[EnableLongTransactions](https://postgis.net/docs/EnableLongTransactions.html) — Uzun işlem desteğini etkinleştirir.

[LockRow](https://postgis.net/docs/LockRow.html) — Tablodaki bir satır için kilitleme/yetkilendirme ayarlar.

[UnlockRows](https://postgis.net/docs/UnlockRows.html) — Yetkilendirme belirteci tarafından tutulan tüm kilitleri kaldırır.

## Sürüm İşlevleri

Bu işlevler PostGIS sürümlerini bildirir ve yükseltir.

[PostGIS\_Extensions\_Upgrade](https://postgis.net/docs/PostGIS_Extensions_Upgrade.html) — Paketler ve PostGIS uzantılarını (örn. postgis\_raster, postgis\_topology, postgis\_sfcgal) mevcut en son sürüme yükseltir.

[PostGIS\_Full\_Version](https://postgis.net/docs/PostGIS_Full_Version.html) — Tam PostGIS sürümünü bildirir ve yapılandırma bilgilerini oluşturur.

[PostGIS\_GEOS\_Version](https://postgis.net/docs/PostGIS_GEOS_Version.html) — GEOS kitaplığının sürüm numarasını döndürür.

[PostGIS\_Liblwgeom\_Version](https://postgis.net/docs/PostGIS_Liblwgeom_Version.html) — liblwgeom kitaplığının sürüm numarasını döndürür. Bu, PostGIS sürümüyle eşleşmelidir.

[PostGIS\_LibXML\_Version](https://postgis.net/docs/PostGIS_LibXML_Version.html) — libxml2 kitaplığının sürüm numarasını döndürür.

[PostGIS\_Lib\_Build\_Date](https://postgis.net/docs/PostGIS_Lib_Build_Date.html) — PostGIS kitaplığının oluşturma tarihini döndürür.

[PostGIS\_Lib\_Version](https://postgis.net/docs/PostGIS_Lib_Version.html) — PostGIS kitaplığının sürüm numarasını döndürür.

[PostGIS\_PROJ\_Version](https://postgis.net/docs/PostGIS_PROJ_Version.html) — PROJ4 kitaplığının sürüm numarasını döndürür.

[PostGIS\_Wagyu\_Version](https://postgis.net/docs/PostGIS_Wagyu_Version.html) — Dahili Wagyu kitaplığının sürüm numarasını döndürür.

[PostGIS\_Scripts\_Build\_Date](https://postgis.net/docs/PostGIS_Scripts_Build_Date.html) — PostGIS komut dosyalarının oluşturma tarihini döndürür.

[PostGIS\_Scripts\_Installed](https://postgis.net/docs/PostGIS_Scripts_Installed.html) — Bu veritabanında yüklü olan PostGIS komut dosyalarının sürümünü döndürür.

[PostGIS\_Scripts\_Released](https://postgis.net/docs/PostGIS_Scripts_Released.html) — Yüklü PostGIS lib ile yayınlanan postgis.sql betiğinin sürüm numarasını döndürür.

[PostGIS\_Version](https://postgis.net/docs/PostGIS_Version.html) — PostGIS sürüm numarasını ve derleme zamanı seçeneklerini döndürür.

## Büyük Birleşik Özel Değişkenler (GUC'ler)

Bu bölüm, özel PostGIS Büyük Birleşik Özel Değişkenlerini (GUC) listeler. Bunlar global olarak, veritabanına, oturuma veya işleme göre ayarlanabilir. En iyi küresel veya veritabanı düzeyinde ayarlanır.

[postgis.backend](https://postgis.net/docs/postgis_backend.html) — GEOS ve SFCGAL'in çakıştığı bir işleve hizmet verecek arka uç. Seçenekler: geos veya sfcgal. Varsayılan olarak coğrafi konumlar.

[postgis.gdal\_datapath](https://postgis.net/docs/postgis_gdal_datapath.html) — GDAL'in GDAL\_DATA seçeneğinin değerini atamak için bir yapılandırma seçeneği. Ayarlanmadıysa, çevresel olarak ayarlanmış GDAL\_DATA değişkeni kullanılır.

[postgis.gdal\_enabled\_drivers](https://postgis.net/docs/postgis_gdal_enabled_drivers.html) — PostGIS ortamında etkinleştirilmiş GDAL sürücülerini ayarlamak için bir yapılandırma seçeneği. GDAL yapılandırma değişkeni GDAL\_SKIP'i etkiler.

[postgis.enable\_outdb\_rasters](https://postgis.net/docs/postgis_enable_outdb_rasters.html) — db dışı tarama bantlarına erişimi etkinleştirmek için bir boole yapılandırma seçeneği.

[postgis.gdal\_config\_options](https://postgis.net/docs/postgis_gdal_vsi_options.html) — Db dışı taramayla çalışırken kullanılan seçenekleri ayarlamak için bir dize yapılandırması.

## Sorun Giderme İşlevleri

Bu işlevler, geometri verilerini sorun giderme ve onarma yardımcı programlarıdır. Yalnızca, normal koşullar altında asla olmaması gereken, geometri verileri bir şekilde bozulmuşsa gereklidirler.

[PostGIS\_AddBBox](https://postgis.net/docs/PostGIS_AddBBox.html) — Geometriye sınırlayıcı kutu ekleyin.

[PostGIS\_DropBBox](https://postgis.net/docs/PostGIS_DropBBox.html) — Sınırlayıcı kutu önbelleğini geometriden çıkarın.

[PostGIS\_HasBBox](https://postgis.net/docs/PostGIS_HasBBox.html) — Bu geometrinin bbox'ı önbelleğe alınmışsa TRUE, değilse FALSE döndürür.

# Pip install library

<https://python-istihza.yazbel.com/standart_moduller/>

1. **Fiona =** Shape dosyalarını okumaya yarar.
2. [**Sys**](https://python-istihza.yazbel.com/standart_moduller/sys.html) **=** Tıpkı os modülü gibi, sys de **Python programlama dilindeki önemli standart kütüphane modüllerinden biridir**. Bu modül, kullandığınız Python sürümü ile ilgili bilgi edinmenizi ve kullandığınız Python sürümü ile çeşitli işlemler yapabilmenizi sağlar.
3. **Shapely =** metinsel bir ifadeden point veya polygon pars etmek için
4. **Numpy =**
5. **Pandas =**
6. **Geopandas =** veriyi çekip görselleştirmek için
7. **Matplotlib =**
8. **Tkinter =**
9. **FastAPI =** web servis oluşturmak için
10. **Uvicorn =** web servis çalıştırabilmek için
11. **Re =**
12. **Psycopg2 =**
13. **Docxpy =**
14. [**Os**](https://python-istihza.yazbel.com/standart_moduller/os.html) **=**
15. **Webbrowser =**
16. **Random =**
17. **OSMPythonTools =**
18. **Collections =**
19. **Folium =**
20. [**Math**](https://python-istihza.yazbel.com/standart_moduller/math.html) **=** math modülü matematiksel işlemler yapmanızı kolaylaştırmak için yazılmış bir modüldür.

[1-NumPy](https://miracozturk.com/python-kutuphaneleri-ve-ozellikleri/#1-NumPy)

[2-SciPy](https://miracozturk.com/python-kutuphaneleri-ve-ozellikleri/#2-SciPy)

[3-Matplotlib](https://miracozturk.com/python-kutuphaneleri-ve-ozellikleri/#3-Matplotlib)

[4-SciKit-Learn](https://miracozturk.com/python-kutuphaneleri-ve-ozellikleri/#4-SciKit-Learn)

[5-SymPy](https://miracozturk.com/python-kutuphaneleri-ve-ozellikleri/#5-SymPy)

[6-Keras](https://miracozturk.com/python-kutuphaneleri-ve-ozellikleri/#6-Keras)

[7-PyTorch](https://miracozturk.com/python-kutuphaneleri-ve-ozellikleri/#7-PyTorch)

[8-GgPlot2](https://miracozturk.com/python-kutuphaneleri-ve-ozellikleri/#8-GgPlot2)

[9-TensorFlow](https://miracozturk.com/python-kutuphaneleri-ve-ozellikleri/#9-TensorFlow)

[10-Pandas](https://miracozturk.com/python-kutuphaneleri-ve-ozellikleri/#10-Pandas)

[11-GnuPlot](https://miracozturk.com/python-kutuphaneleri-ve-ozellikleri/#11-GnuPlot)

[12-Theano](https://miracozturk.com/python-kutuphaneleri-ve-ozellikleri/#12-Theano)

[13-Caffe](https://miracozturk.com/python-kutuphaneleri-ve-ozellikleri/#13-Caffe)

[14-PyMC3](https://miracozturk.com/python-kutuphaneleri-ve-ozellikleri/#14-PyMC3)

[15-Chainer](https://miracozturk.com/python-kutuphaneleri-ve-ozellikleri/#15-Chainer)

[16-Gensim](https://miracozturk.com/python-kutuphaneleri-ve-ozellikleri/#16-Gensim)

[17-Statsmodels](https://miracozturk.com/python-kutuphaneleri-ve-ozellikleri/#17-Statsmodels)

[18-SpaCy](https://miracozturk.com/python-kutuphaneleri-ve-ozellikleri/#18-SpaCy)

[19-HyperLearn](https://miracozturk.com/python-kutuphaneleri-ve-ozellikleri/#19-HyperLearn)

[20-NLTK](https://miracozturk.com/python-kutuphaneleri-ve-ozellikleri/#20-NLTK)

[21-Seaborn](https://miracozturk.com/python-kutuphaneleri-ve-ozellikleri/#21-Seaborn)

[22-Plotly](https://miracozturk.com/python-kutuphaneleri-ve-ozellikleri/#22-Plotly)

[23-MpMath](https://miracozturk.com/python-kutuphaneleri-ve-ozellikleri/#23-MpMath)

[24-Scrapy](https://miracozturk.com/python-kutuphaneleri-ve-ozellikleri/#24-Scrapy)

[25-Requests](https://miracozturk.com/python-kutuphaneleri-ve-ozellikleri/#25-Requests)

[26-Beautiful Soup](https://miracozturk.com/python-kutuphaneleri-ve-ozellikleri/#26-Beautiful_Soup)

[27-Flask](https://miracozturk.com/python-kutuphaneleri-ve-ozellikleri/#27-Flask)

[28-Bokeh](https://miracozturk.com/python-kutuphaneleri-ve-ozellikleri/#28-Bokeh)

[29-wxPython](https://miracozturk.com/python-kutuphaneleri-ve-ozellikleri/#29-wxPython)

[30-Pillow](https://miracozturk.com/python-kutuphaneleri-ve-ozellikleri/#30-Pillow)

[31-SQL Alchemy](https://miracozturk.com/python-kutuphaneleri-ve-ozellikleri/#31-SQL_Alchemy)

[32-Twisted Matrix Lab](https://miracozturk.com/python-kutuphaneleri-ve-ozellikleri/#32-Twisted_Matrix_Lab)

[33-PyGame](https://miracozturk.com/python-kutuphaneleri-ve-ozellikleri/#33-PyGame)

[34-Pyglet](https://miracozturk.com/python-kutuphaneleri-ve-ozellikleri/#34-Pyglet)

[35-PyQt](https://miracozturk.com/python-kutuphaneleri-ve-ozellikleri/#35-PyQt)

[36-DeepFace](https://miracozturk.com/python-kutuphaneleri-ve-ozellikleri/#36-DeepFace)

[37-NetworkX](https://miracozturk.com/python-kutuphaneleri-ve-ozellikleri/#37-NetworkX)

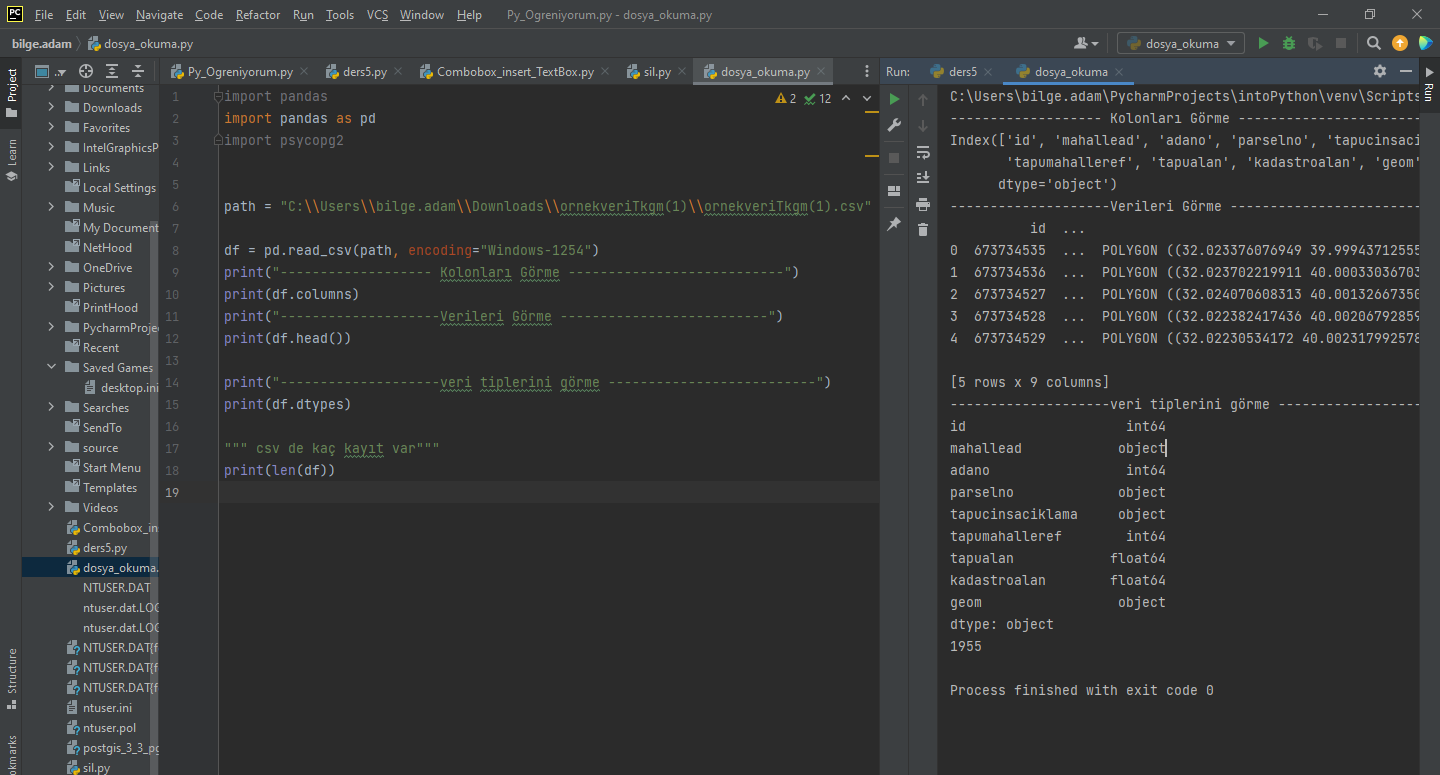
# Pandas Dosya Okuma – Veri Tabanı Oluşturma – Veri Tabanına Pythondan Veri Gönderme

## Csv okuma

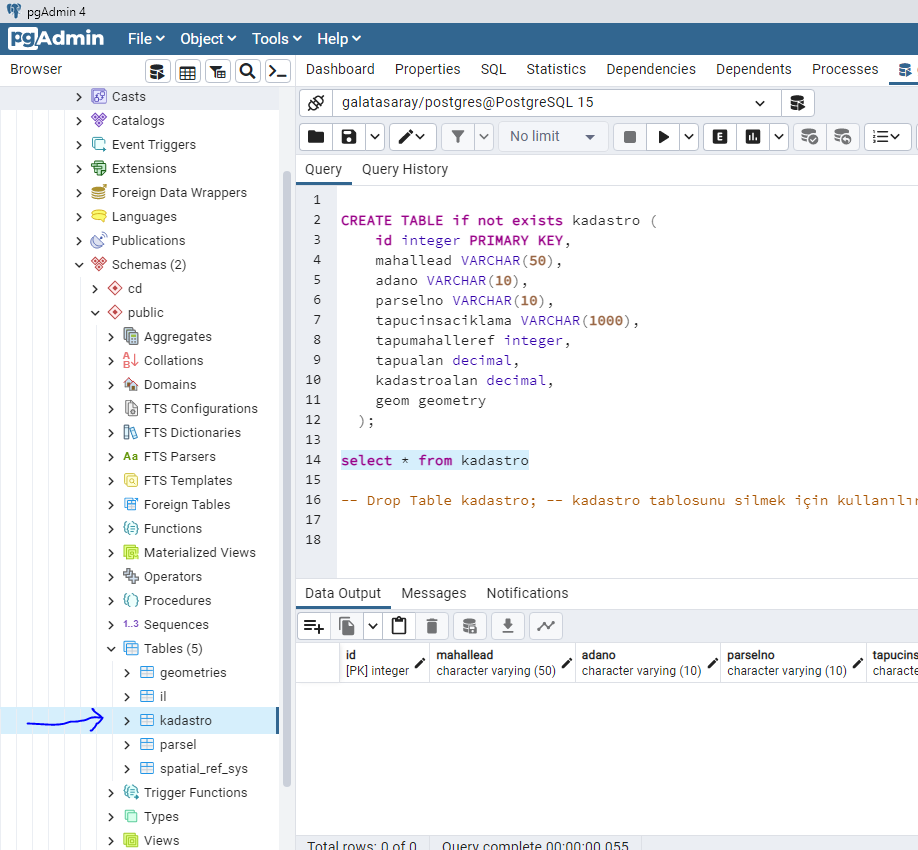
import pandas  
import pandas as pd  
  
  
path = "C:\\Users\\bilge.adam\\Downloads\\ornekveriTkgm(1)\\ornekveriTkgm(1).csv"  
  
df = pd.read\_csv(path, encoding="Windows-1254")

Csv içindeki kolonları görme – columns

import pandas  
import pandas as pd  
  
  
path = "C:\\Users\\bilge.adam\\Downloads\\ornekveriTkgm(1)\\ornekveriTkgm(1).csv"  
  
df = pd.read\_csv(path, encoding="Windows-1254", sep=";")  
print("------------------- Kolonları Görme ---------------------------")  
print(df.columns)

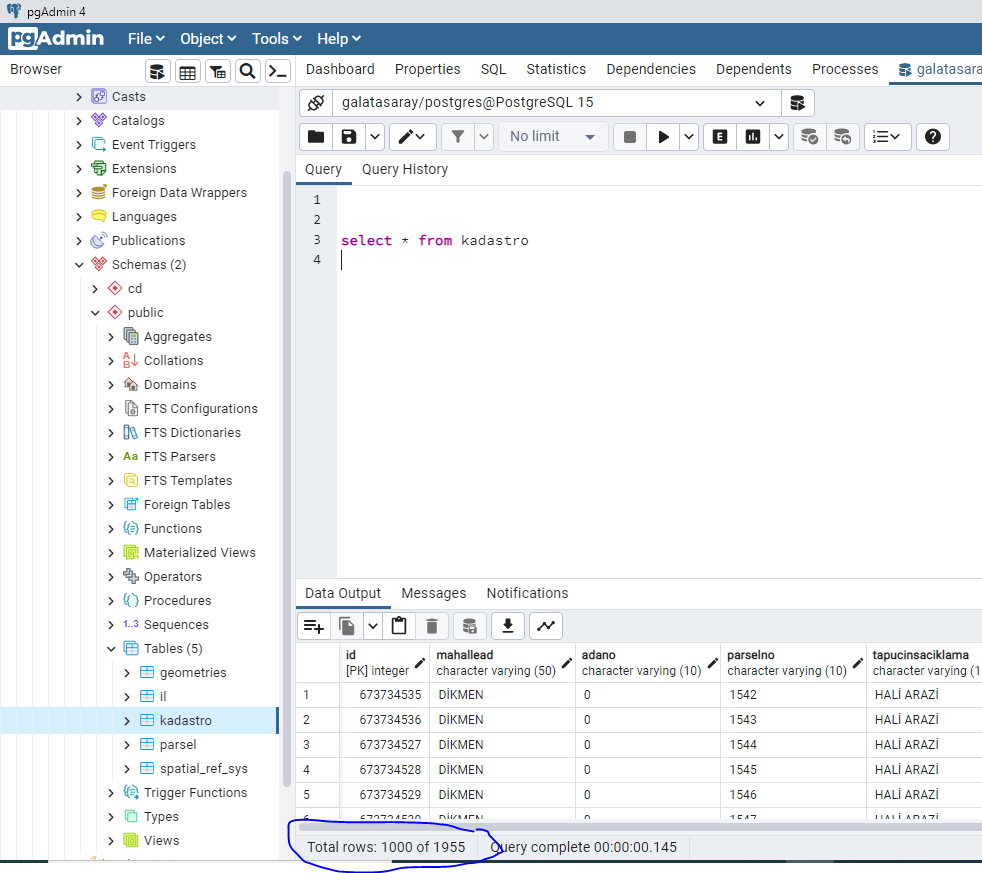


## Veri tabanı oluşturma



## Csv verisini veri tabanına gönderme

import pandas  
import pandas as pd  
import psycopg2  
  
  
path = "C:\\Users\\bilge.adam\\Downloads\\ornekveriTkgm(1)\\ornekveriTkgm(1).csv"  
  
df = pd.read\_csv(path, encoding="Windows-1254")  
print("------------------- Kolonları Görme ---------------------------")  
print(df.columns)  
print("--------------------Verileri Görme --------------------------")  
print(df.head())  
  
print("--------------------veri tiplerini görme --------------------------")  
print(df.dtypes)  
  
""" csv de kaç kayıt var"""  
print(len(df))  
  
print(" ---------------------veri tabanına bağlanma------------------------------- ")  
conn=psycopg2.connect("dbname=galatasaray user=postgres password=123")  
cur=conn.cursor()  
  
print(" ------------------- csv deki verileri veritabanındaki kadastro tablosuna basma ------------------------ ")  
for i in range(len(df)):  
 satır=df.iloc[i]  
 sql= f"""  
 INSERT INTO kadastro values (  
 {satır['id']},   
 '{satır['mahallead']}',   
 '{satır['adano']}',  
 '{satır['parselno']}',  
 '{satır['tapucinsaciklama']}',  
 {satır['tapumahalleref']},  
 {satır['tapualan']},  
 {satır['kadastroalan']},   
 '{satır['geom']}'  
 );  
 """  
 cur.execute(sql)  
conn.commit()



## Combobox’tan Texbox’a docx uzantılı dosyayı gönderme

import docxpy  
import os  
import webbrowser  
from tkinter import \*  
from tkinter import ttk  
import tkinter as tk  
  
anaSayfa = Tk()  
width\_value = anaSayfa.winfo\_screenwidth() #: anaSayfa'nın otomatik genişliğini ayarlar  
anaSayfa.geometry("750x750+0+0") #: anaSayfa'nın otomatik genişliğini ayarlar  
  
anaSayfa.title("Python Öğreniyorum")  
anaSayfa.resizable(width=False, height=False) #:anasayfayı sabitleme - genişletip daraltmayı kaldırır  
  
  
def retrieve():  
 print(Combobox1.get())  
  
  
frame = Frame(anaSayfa)  
frame.pack()  
frame.place(x=5, y=18)  
  
vlist = ["abc.docx", "Option2", "Option3", "Option4", "Option5"]  
  
Combobox1 = ttk.Combobox(frame, values=vlist, height=600, width=30)  
Combobox1.set("abc.docx")  
Combobox1.pack()  
  
  
def BrowseFile():  
 file1 = 'C:\\abc.docx'  
  
 message1 = docxpy.process(file1)  
 message1 = message1.encode('ascii', 'ignore').decode('ascii')  
  
 print("reach here", message1)  
 print(dir(textBox1))  
 textBox1.insert(tk.END, message1)  
  
  
btn1 = Button(frame, text="Getir", command=BrowseFile)  
btn1.pack(padx=5, pady=5)  
  
global textBox1  
textBox1 = Text(anaSayfa)  
textBox1.place(x=210, y=18, height=600, width=1695)  
  
anaSayfa.mainloop()

Fatih ERCAN – Bilge Adam \* Eğitmen

fatih.ercan@bilgeadam.com

Tel: 0532 475 0257

Deniz AKGÜNLÜ – Bilge adam Salon sorumlusu

0537 476 05 95

Emre YAZICI – Bilge Adam Eğitmen

<https://github.com/eyazicimina/tk/blob/main/sql-%C3%B6rnek.txt>