30.09.24

Large standard library

Arka planda çok fazla kütüphanesi mevcuttur.

Pycharm güzel bir ide’dir.

Bir stringe hangi işaretle başlarsak o işaretle bitirmemiz gerekir.

- tek tirnak -> single quotes

- cift tirnak -> double quotes

- / -> slash

- - tire - dash

- : colon iki nokta ust uste

- ; semi colon , noktali virgul

01.10.24

Derste yazılanlar mutlaka ertesi gün tekrardan yazılacak.

""Zorluklar, başarıya giden yolun gizli rehberleridir. Her engel, seni bir adım daha ileri taşır."" Çift double code olmaz hata verir. Ancak üç tane olabilir. Ya bir adet olacak ya da üç adet olacaktır.

'"Disiplin, arzularınızı hedeflerinizin önüne koymamaktır." — Abraham Lincoln' bu çalışıyor. Çünkü string olarak kabul ediyor.

Concatenation: farklı stringlerin birleştirilmesidir.

Markdown

# başlık1 ile hash işareti 6 ya kadar artırılarak başlık yazna boyut küçültülebilir.

Pep8 kuralları

3+9 # pep8 kurallarına uygun değildir. estetik yazma kuralları olarak düşünülebilir. zorunlu olmayan ancak daha okunaklı olması için gerekli olan şyedir.

3 + 9 # pep8 kurallarına uygundur. Okunurluk çok önemlidir.

Variables’ta age olması lazım age her zaman yaştır. X yaş olarak atanırsa sorun olur doğal olarak bu pep8 kuralına uygun değildir denilebilir.

Variables

isim = aman dikkat büyük küçük harf önemlidir.

### string -> 'Jane Blue', 'USA', 'Kuru Fasulye'

### integer -> 78, 9000, -386539 (Tam Sayı)

### float -> 6.9, 35.99, -990.791 (küsuratlı sayı, nokta olacak asla vürgül değil )

### boolean -> isRetired -> True, False

# print kullanarak

print(type(age))

print(name, age)

# tek print kullanarak variables arasına virgül koyarak yazdırılabilir.

plakaNo = camelcase

plaka\_no = snakecase

userFirstname # ilk kelime küçük harflerle yazılı daha sonra gelen kelimeler ise ilk harfleri büyük olur.

Ctrl + python da yazıları büyütür ve küçültür.

Boolean, bool

Normalde adı boolean’dır ancak python da adı bool diye geçer.

# True, False kelimeleri ilk harfi buyuk yazilmalidir pythonda.

isScheduled = false #sistem false kelimesini variable zannediyor ilk harfi kucuk oldugundan

- ctrl +c -> kopyala

- ctrl +v -> yapistir

- ctrl +y -> yeniden yap

- shıft + alt +asagi oku tusu -> uzerinde bulundugunuz satirin kopyasini cikarir

- shift + enter -> hucrelerdeki kodlari calistirmak icin kullandigimiz bir kisayol

- ctrl + a -> hepsini secmek

Öncelikle city\_2’yi variable olarak bolu olarak atıyoruz. Sonrasında yazdırıyoruz sonra variable ı sildikten sonra bile yazdırmaya devam ediyor. Çünkü ram de bir yer kapladığından dolayı devam eder. Bunun için del fonksiyonunu kullanıyoruz.

del city\_2

print(city\_2)

variable oluştururken nelere dikkat etmeliyiz

1. Sayı ile variable isimleri başlayamaz
2. \_under\_score ile ile başlayabilir.
3. @ ile başlayamaz. $ ile & başlayamaz hocam.
4. \_ underscore haricinde herhangi bir işaret kullanamıyoruz. Bu iş bu kadar hocam…..

Collection = çoklu data tutabildiğimiz bir mevzudur.

country\_1, country\_2 = "USA" , "tURKEY"

print(country\_1, country\_2)

USA tURKEY

istanbul = izmir = adana = 30

ankara = trabzon = bursa = 25

van= erzurum = 20

print(van)

tuple ve collection data tiplerine bakmak lazım. İleride bakacağız inş.

02.10.24

jupyter kullandigimizdan dolayi print fonksiyonunu kullanmasak bile type kullandigimiz zaman cikti alabiliriz

fakat alt alta bir kac tane bu sekilde cikti almaya calistigimiz zaman, sistem sadece en sondakini print eder.

Digerlerini print etmez

Type conversion olarak bakılacak

Dersi derste anlamaya bak. Sabah bakmaya fırsatın olmayabilir.

Kalıcı ve geçici değişiklik

height = str(height) olacaktır.

print(type(height)) dediğimizde sonuç str olacaktır.

Bu kalıcıdır.

String integer e çevrilebilri mi?

Eğer bir string integer a benziyorsa ancak çevrilebilir. Aksi takdirde olmaz.

Float bir sayı integer a çevrilirken noktadan sonraki kısım tamamen atılır.

Olabildiğince değişken tanımlarsak daha iyi olacaktır.

Hocanın kuralı chat cpt ye yazdırılmayacak hocam

steps\_today = "5000"

steps\_today\_int = int(steps\_today)

steps\_yesterday = steps\_today\_int - 200

toplam\_adim = steps\_today\_int + steps\_yesterday

print('Toplam adım sayisi= ', toplam\_adim)

Kodlamada tekrar sistemi yorduğundan iyi değildir. Olabildiğiince kodu tekrarlamadan yazmak gerekir.

# Boolean to Int

int(True) varlık tır ve birdir.

İnt(False) yokluk tur ve sıfırdır.

- Carpma \*

- Bolme / = işleminin sonucu float tır. Tam sayı olarak gerekli ise tam bölme yani // kullanılabilir.

- Toplama +

- Cikarma -

- Modülüs %

- Tam Bolme // = sonucu integer olur. Farkı budur. Adı floor division dur.

- Us alma \*\*

pow(2,2) ikinin karesi hocam.

# kökünü almak için herhangi bir fonksioyn yoktur. O yüzdenkaresinin ters mantığını almak lazım.

İşlem önceliği bu tam anlaşılmadı bununla ilgili videolara bak hocam….

1. Parantez
2. Uslu ifadeler
3. Carpma ve bölme
4. Toplama ve cıkarma

String her zaman “” arasında bulunur.

İnteger bir adet sayıdır. Ancak kenarında “” dalgası bulunmaz.

Float ise mutlaka noktalı bir sayıdır.

String bir ifadeyi integer’a çevirmek istersek ifadenin integer a benzemesi gerekir.

Aynı şekilde string ifadeyi float a çevrimek istersek ya float a ya da integer a benzemesi gerekir.

Python case sensitive dir dolayısıyla false diye bir şey olmayacaktır.

Tuple çoklu data barındırabilecek bir değerdir. Arasında virgül ile yazılır.

# bölme işlemi sonucu her daim flaot değer vereri tam bölünmesinin önemi yoktur.

03.10.24

Python kodları soldan sağa doğru okuyacaktır.

Parametre= ürettiğimiz fonksiyonları dinamik hale getirmek için kullanılır.

help(print) yazdığımız zaman size fonksiyonun ayrıntılarını döndürecektir.

print(\*args, sep=' ', end='\n', file=None, flush=False)

Prints the values to a stream, or to sys.stdout by default.

sep

string inserted between values, default a space.

end

string appended after the last value, default a newline.

Buna bakıldı. Bununla ilgili örnek çözülecektir.

file

a file-like object (stream); defaults to the current sys.stdout.

flush

whether to forcibly flush the stream.

Unicode’lar: bu şekilde bir şey vardır.

<https://www.geeksforgeeks.org/python-program-to-print-emojis/>

burada emojiler vardır hocam. Güzel bir uygulamadır.

Dikkat= parametre nedir onu tamamen öğrenmemiz gerekmektedir. En az 3-4 video izlemek gerekir.

Fonksiyon nedir ve parametre nedir. Bu konuya bakmak lazım. Nasıl yazılır öğrenmek gerekmektedir.

'\n' escape sequence demektir.

Ctrl + shift + L aynı olan yazıları seçip bu kısayolu kullandığınızda dataları aynı anda değiştirebiliriz.

F string

Format() son derece karışıktır.

age = int(input(“Lütfen isminizi giriniz: ”))

input fonksiyonu değeri string şekilde alacaktır.

Kürenin yarı çapını kullanıcıdan alacağız. Sonrasında ise

yaricap = int(input(“lütfen hacmini bulmak istediğiniz kürenin yaricapini giriveriniz: “))

kurenin\_hacmi = (4/3)\*(3.14)\*(yaricap \*\* 3)

print(f”Yarıçapı {yaricap} olan kürenin hacmi {kurenin\_hacmi} cm3’tür efenim.”)

print('Laptop,' 'Computer,' 'Phone', sep='--')

# stringlerin arasinda virgul olmadigindan seperator parametresine ihtiyac duyulmaz,

# bu sebepten -- eklenmez ,

# python bu yan yana yazilmis stringleri birlestirir cikti verir

print("a" , "b" , "c" , sep="--")

seperator parametresi tetiklenmesi için araya virgül konması gerekliliğidir. Virgülsüz seperator çalışmıyor.

04.10.24

import math dediğimizde

matematik kütüphenesini açıyoruz demektir.

Print(f”Yarıçapı {yaricap} olan bir dairenin alanı {int(alan)} cm2’dir.”)

Print(f”Yarıçapı {yaricap} olan bir dairenin alanı {round(alan, 2)} cm2’dir.”)

\ bastığımızda bu kaçış manevrasıdır. Bunu yani string ten say diyoruz.

print('tom\'s shoes') burada tek tırnağı string olarak kabul eyle hocam diyoruz.

print('tomun \n ayakkabıları \n cok güzeldir.')

çıktısı:

tomun

ayakkabıları

cok güzeldir.

Şeklindedir.

Ödev1 \t

Sep = ‘b’

Comparison operators

<

>

>=

<=

== karşılaştırma operatörüdür.

= atama operatörüdür.

!= eşit değil midir diye sormaktır.

# database = veritabanı

sifre = 'hope7.\*'

user\_password = input('şifrenizi giriniz lütfen:')

print(sifre == user\_password) # eşit midir diye soruyor.

İnput dataları string olarak alır.

password = 123

web\_pass = int(input('şifreyi firin'))

print(password == web\_pass)

True > 0 ve False < 1 bu tarz örneklere bakılacaktır.

Ascii values

String kıyaslaması yapılabilir mi?

Ascii nerde işimize yarar? Programın çalışma mantığı bu şekildedir.

Ord diye bir fonksiyon mevcuttur.

ord(‘m’) ascii değerini gösterir yani bir çeşit ebcet hesabı sonuç 109’dur.

Bunun tersi ise chr(65) tir o nun da sonucu A’dır.

#### Falsy and Truthy Values

'ahmet'

'mehmet'

45

0

56

# Bunların hepsinin birer boolean dereği var arkadaşlar.

bool('murat') sonucu True’dur.

print(f' 0 {bool(0)}')

print(f' 0.0 {bool(0.0)}')

print(f' False {bool(False)}')

print(f"' {bool('')}")

bunların neticesi False tur.

And or Logical Expressions

6 and 9 and “melisa” sonucu melisa dır.

# and

# hepsi true ise sondaki elemanı yazdırır.

# bir tane bile false varsa ilk gördügü false i yazdırır.

# bütün şartların sağlandığı durumlarda and kullanacağız.

# mükemmelliyetçidir.

Or operatörü çok iyimser dir tek bir tane True varsa onu döndürür. Ama eğer truthy değer yoksa o zaman en sondaki değeri döndürür.

8 or 0 or ‘ankara’

Or true değeri arar and ise false değeri arar

True and 0 or False or 45 and 3

45 and 3

3

Bunun sonucunu bulmaya çabala hocam

Matematikteki mantık konusu ve veya değil dir.

7 and 0 or False neticesi ne olur?

False or (False and ‘Melisa’ or not ‘ ’ and (0 or 0.1))

Bunun neticesini bulana kadar çalışmak gerekmektedir.

# kutlesi(kg) ve hizini(m/s) kullanicidan aldiginiz bir cismin kinetik enerjisini hesaplayiniz

# Kullanicinin boy ve kilo degerlerini alip kullanicinin bmi degerini hesaplayin

('Ankara' or 'Istanbul' and 0 and not '') or not (True and 679 or '')

Cismin\_kutlesi = m = int(input(“Lütfen cismin kutlesini kg cinsinden giriniz: “)

Cismin\_hizi = v = int(input(“Lütfen cismin hizini metre/saniye cinsinden giriniz: “)

Kinetik\_enerji = ½ \* cismin\_kutlesi \* cismin\_hizi \*\*2

# and = iki veya daha fazla koşulun True olup olmadığını kontrol eder

# or = en az bir koşulun True olup olmadığını kontrol eder

# not = Koşul False ise True, ve tam tersi

('Ankara' or 'Istanbul' and 0 and not '') or not (True and 679 or '')

'Ankara' or 'Istanbul' = ‘Ankara’

05.10.24

sayi = 1

print(f"{sayi} karesi= {sayi \*\* 2}")

print(f"{sayi+1} karesi= {(sayi+1) \*\* 2}")

print(f"{sayi+2} karesi= {(sayi+2) \*\* 2}")

print(f"{sayi+3} karesi= {(sayi+3) \*\* 2}")

print(f"{sayi+4} karesi= {(sayi+4) \*\* 2}")

print(f"{sayi+5} karesi= {(sayi+5) \*\* 2}")

print(f"{sayi+6} karesi= {(sayi+6) \*\* 2}")

print(f"{sayi+7} karesi= {(sayi+7) \*\* 2}")

print(f"{sayi+8} karesi= {(sayi+8) \*\* 2}")

print(f"{sayi+9} karesi= {(sayi+9) \*\* 2}")

hackerrank sitesi gayet iyidir.

07.10.24

# or bir tane bile doğru görse direk dondürür.

# eğer hic true yolsa sondaki false elemanın kendisini döndürür.

# ilk true’yu son false’u döndürür hocam.

Sıra şu şekildedir = not and or

Güzel bir gün ile başlayan çalışma baştan yapılacak.

# Strings

1. immutable yani yeniden atama yapılmadığı müddetçe degistirilmeez.

2. iterable : elemanları tek tek kullanılabilendir.

3. pozitif indeksleme baştan ve sıfırdna başlar.

4. negatif indeksleme sondan ve -1'den başlar.

5. indeksleme yaparken [] kullanırız. Yani köşeli parantez kullanılır.

6. her zaman sonuncu karakter -1’dir.

course = "Techpro education"

course[7] burada boşluk da bir karakterdir.

7. string’ler inmutable dir yani değiştirilemezler.

8. == karşılaştırma operatörüdür.

9. stringlerde çarpma işlemi de çalışır. Çarpmadan ziyade duplication olarak söylesek daha iyi olur.

10. slicing

# [başlangıç noktası : bitiş noktası]

# başlangıç noktası dahildir.

# bitiş noktası dahil değildir.

11. "computer"[0,3] neticesi com’dur. Bitiş 900 desek yine de hata vermez.

“computer[:5] başlangıç noktasını tanımlamadığımız zaman default olarak yani varsayılan olarak en baştakini kabul eder. computer[5:] beşinciden başla demek. computer[:5] beşinciye kadar say demek. Eğer başlangıç indeksini belirleyip bitiş indeksini belirlemezsek sonuna kadar gider. Eğer başlangıcı boş bırakırsak “computer”[-3,2] netice olarak “ ” veriyor. Sebebi ise bitişi önce gördüğü için başlamadan bitirdi.

12. text[başlangıç: bitiş noktasi: adım]

13. "Dilara"[0:6:-1] başlamadan biteceği için netice yoktur.

14. "Abdullah kose"[-8:-2:-1]

15. “Furkan enes ersoy”[-9:-4:-1) bu boş stringe verir zira daha başlamadan bitiş noktasını görür hocam bu.

16. Birinci her zaman sıraya göre ikinciden daha küçük olmalıdır. Ters yönde gidenlerde başlangıç noktası bitiş noktasından hep yüksekse hep boş string veriyor denilebilir.

String Methods

1. Programlama dilinde bazı fonksiyonlar vardır ve metodlar vardır.

2. fonksiyonlar parantez ile çağrılır ve çalıştırılır.

3. country = "Turkey"

country.upper()

08.10.24

Color[2] dediğimizde bu indexing olur. Slicing olmaz.

And or not logical expressions

Sırası ise not and or şeklindedir. ,

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| and | T1 | T2 | T1 | F1 | F1 | F2 |
| T2 | | F1-2 | | F1 | |
| or | T1 | T2 | T1 | F1 | F1 | F2 |
| T1 | | t | | F2 | |
| not | T1 | |  |  | F1 | |
| f | |  | | t | |

# birinci parametrenin (başlangıç) varsayılan değeri : 0 dır. En baştan başlar

# ikinci parametrenin (bitiş) varsayılan değeri : 0 dır. En sona kadar.

# üçüncü parametrenin varsayılan değeri 1 dir. Soldan sağa doğru okur.

# -1 yaparak yönü değiştirebiliriz. Sağdan sola doğru okumasını sağlamış olduk.

BRO CODE GÜNÜN DERSİ İÇİN SÖYLEDİKLERİ

name = input("Enter your name: ")

phone\_number = input("Enter your phone #: ")

length = len(name) # yazılan bir string ifadenin uzunluğunu gösterir.

index = name.find(" ") # The Python string find() yöntemi, verilen dizede alt dizenin ilk geçtiği yerin dizinini bulmaya yardımcı olur.

name = name.capitalize() # bir kelime yada cümlenin ilk harfini büyütür.

name = name.upper() # tüm string objeleri büyütür.

name = name.lower() # tüm string objeleri küçültür.

result = name.isdigit() # Karakter dizisinin sadece sayılarda oluşup oluşmadığını sorgular

result = name.isalpha() # Bu metot yardımıyla bir karakter dizisinin 'alfabetik' olup olmadığını denetleriz.

result = phone\_number.count(" ") # bu metot yardımı ile arattığımız karakterin kaç tane olduğunu gösterecektir. Sıra sıfırdan başlar,

phone\_number = phone\_number.replace("-", "")

capitalize ilk harfin büyük diğerlerinin küçük harflerle yazılmasıdır. Sadece cümlenin ilk harfini büyük haline getirir diğerlerinin tamamını küçük harfe çevirir.

Title ise cümledeki her kelimenin ilk harfini büyük hale getirecektir.

Örneğin bir cümle giriniz ve press e basınız. ın title hali “Örneğin Bir Cümle Giriniz Ve Press E Basınız.” Şeklindedir.

Replace bu biraz karışık: # dikkatli olunması gereken konu burada büyük küçük sensitive olduğudur. Doğrudan örneğin A yı aratmamı gerekmektedir.

"my dear darling".replace("d", "@", 2) ilk iki d yi @ ile değiştir diyoruz.

"te.....chpro..... edu.....cati....on".replace(".", "") çıktısı techpro education’dur.

"!!!!cptunaldİ@gmaİl.cöm".replace("!","").replace("ö", "o").replace("İ", "i") çıktısı [cptunaldi@gmail.com’dur](mailto:cptunaldi@gmail.com'dur).

print("te.....chpro..... edu.....cati....on".replace(".", "\n")) bunun çıktısı her bir nokta yerine enter değeridir.

Dikakt : Yalnız bu escape sequence ile olması için print fonksiyonu ile yazdırılması gerekmektedir.

Python dizelerinde ters eğik çizgi "\", "kaçış" karakteri olarak da adlandırılan özel bir karakterdir. Belirli boşluk karakterlerini temsil etmede kullanılır: "\t" bir sekme, "\n" bir yeni satır ve "\r" bir satır başıdır.

help(str.replace) bunu ezberle hocam. Replace sadece string te kullanılır hocam.

Örnek: "Kendin mi senin yoksa, ümidin mi yüreksiz?".title().upper().capitalize().lower().replace("m", "@") sonucu: 'kendin @i senin yoksa, ü@idin @i yüreksiz?' # buna zincirleme metod denilir.

"deneme olsa da nerede olsa?".isalnum() netice false dır.

Ödev: tüm metodları nasıl kullanılacak unun pratiğini yap hocam.

Önemli : is ile başlayan string metodlar devamlı olarak boolean döndürür. Yani doğru yanlış veya evet hayır veya True False sevgili dostum.

# %%

s = "123"

print(s.isdigit()) # True

s = "123²"

print(s.isdigit()) # True (² karakteri kabul edilir)

s = "Ⅳ" # Roma rakamı

print(s.isdigit()) # True (Roma rakamı da kabul edilir)

s = "123.45"

print(s.isdigit())

# %%

s = "123.45"

print(s.isdecimal()) # False (Nokta dahil olduğu için)

s = "Ⅳ" # Roma rakamı

print(s.isdecimal()) # False

"Jennifer".isupper() # False çünkü hepsi upper değil lower da değildir. Arkadaşlar.

İsnumeric var ve roma rakamı ve rakam kabul ediyor.

Swapcase

Büyükleri küçük harfe ve tesine çevirir hocam.

"Ahmet ünaldı Edremit'te yaşamaktadır.".swapcase()

Count metodu

"Ahmet nerede oturmaktadır. Edremitte olabilir mi? Bilemiyorum".count("k") + 10 # denilebilir.

Netice olarak 1 verecektir.

Count fonksiyonu karakterleri değil karakterlerin arasını sayarmış.

Split fonksiyonu:

"mustafa".split("a") # neticesi: a lar yerine boşluk koyacaktır.

Strip()

"asassafasfasfasfasfasfasfasfasfasfasfasfasfasasmrkezeasasas".strip("as")

# 'fasfasfasfasfasfasfasfasfasfasfasfasfasasmrkeze'

strip(self, chars=None, /) unbound builtins.str method # sağındaki ve solundaki boşlukları alır diğerlerini yani ortadakileri ellemez.

Return a copy of the string with leading and trailing whitespace removed.

Bu fonksiyonun right strip ve left strip modelleri de vardır.

Not: öncelikle,

1. işi algılamak lazımdır.
2. Sonrasında onun fonksiyonunu bilmek gerekmektedir.
3. Sonrasında ise adımları bilmek gerekmektedir.
4. En sonunda ise sentaksına hakim olmak gerekmektedir.

Not: (bunlar tecrübedir) {bolca tanınabilir variable tanımlamak kolaylık sağlayacaktır.}

Strip metoduna bir kez daha bakmak lazım.

len fonksiyonu sadece stringler ile çalışır. İterable olmalı.

09.10.2024

İf’ten sonra gelen kısım True ise alttaki ibareyi çalıştırır.

bool(‘Yasin’) dersek netice True olacaktır.

# if in yanında ki sart kısmında olan değerin boolean değerine bakıyoruz.

# eger falsy bir değerse statement çalışmaz.

# truthy bir değerse statement calısır.

İf 7==8: # deniyorsa yedi eşit midir sekize deniyor.

True değilse

İndentation olmadan başlayan satırlar if ile alakası yoktur.

if True:

x = 0

x += 5

print(x) # neticesi 5 tir.

if condition1:

statement1:

else condition2:

statement2.:

eğer condition 1 True ise statemen1 olur. Eğer condition2 ise statemen2 olur.

x = 90

if x < 40:

print('40 tan küçük')

elif x < 60:

print("60 tan küçük")

elif x <80:

print('80 den küçük')

elif x <= 90:

print('90 dan küçük veya eşit')

else:

print('something went wrong')

# elif istenilen kadar kullanılabilir else ise en sonda ve diğer tüm opsiyonları kapsamalıdır. Else şart ile yazılmaz. Else son olarak yazılır ve tek kez yazılır.

Yarın listleri izleyeceğiz.

10.10.2024

1. Listler çoklu data bulundurabilirlre. Birden fazla aynı datayı atabilirsiniz.

2. Mutable dir yani atama yapmadan değiştirilebilir demektir. Buna mutlaka bir bak. Yani assignment olmadan elemanlarını değiştirebiliriz. Hem de kalıcı değiştirebiliriz. Büyük olaydır bu bre.

3. Stringler inmutable dir.

4. İterable da: elemanlarına teker teker ulaşabiliriz demektir.

5. İndekslenebilir demektir aynı zamanda.

# list metodunun içine koyduğumuz data iterable olmalıdır.

# daha dinamik kod yazacağız inş.

# çok uzun olduğu zaman alt alta yazar hocam.

# clear metodu listenin içini temizler. Listenin kendisine dokunmaz. Del metodu ise listenin kendisini siler. Ancak bu tehlikelidir. Çok dikkatli olunması gereken bir fonksiyondur. Geri alınmaz.

NOT: gün içinde önceki günün konularını tekrar etmekle etmemekle arasında dağlar kadar fark vardır.

# kalıcı değiştiren methodlar

# remove

# pop

# append

# insert

# sort

# clean

#del

y = [3,8,0,'a', False, 'B','C','True','c',0.0]

# listenin içinde integer ve string ifadeler olduğu için nasıl sıralama yapılacağı anlaşılmadı

# listenin içindeki dataların data tiplerinin birbirine uygun olması gerekir.

# yani str ve int birlikte olan bir list sıralanamaz. sort methodu uygulanamaz.

Practicepython.org sitesine bakmakta fayda var. Allah büyükür.

Hackinscience basic python alistirmalari icin iyi olabilir.

Bide DataLemur'un SQL sorulari cok iyi. beni haftaya bu gruptan atacaklari icin (:slightly\_smiling\_face:) simdiden paylasmak istedim. o siteyide kaydedip SQL bittikten sonra gunde birer tane falan cozebilirsiniz. Orda python'un da bi kac tane sorusu var.

11.10 2024

# tuple = collection which is ordered and unchangeable: tuple’lar düzenli ve değiştirilmezdir.

# used to group together related data: birbiri ile alakalı datayı bir arada tutmaya yarayacaktır.

# List = [] ordered and changeable. Duplicates OK

# Set = {} unordered and immutable, but Add/Remove OK. NO duplicates

# Tuple = () ordered and unchangeable. Duplicates OK. FASTER

Listelere göre daha güvenli ve daha hızlıdır. Bunun sebebi tuple’lar degistirilemez. İnmutable dır. Elemanlarını değiştiremeyiz demektir. Listelerde gördüğümüz pek çok fonksiyon tuple için kullanılamaz manasınagelmektedir.

Faklı data tiplerinde eleman bulundurulabilir.

Degistirmek istediğimizde önce listeye dönüştürüp degisiklik yaptıktan sonra tekrar tuple a dönüştürebiliriz.

Mutlaka her gün hocanın yazdıklarını mutlaka birkaç kez kendin de yazarak pratik yap. Bakmak ve bakmamak bir de yazmak ve yazmamak arasında dağlar kadar fark var.

7.9 bunun tipi float iken 7,90 bunun tipi ise tuple dır. İlginç mevzu. Etrafında parantez olmadığı halde bunu iki ayrı eleman olarak algılamaktadır.

Tuple ın içinde bir string varsa değiştirilemez ama bir liste varsa o değiştirilebilir.

Parantez olmasa da yine de tuple dır.

14.10.24

X = 4,6

Bu bir tuple dır.

X = 4,

Bu da tuple’dır.

(1,2)\* 3 şeklinde işlemler mevcuttur. Çıkarma ve bölmeye bakmanı tavsiye ederim.

Tuple ve tuple toplanabilir.

Not: dersten önce bir baktın ve çok iyi bir iş yaptın sevgili dostum.

karisik\_tuple[0] = 9999 # burada error verecektir çünkü işte bu işlem inmutable olduğunu göstermektedir.

Append fonksiyonu tuple’da bulunmamaktadır.

Tuple pek işleme müsait olmadığı için pek sık kullanılmazlar. Örneğin karisik\_tuple[2].upper dediğimizde bu error verecektir.

Tuple’ın içindeki bir list değiştirilebilir. Örneğin oraya bir eleman eklenebilir. Listeye bir eleman eklenebilir.

Not. Buna mutlaka desten önce bir bak ve her bir satır ile ilgili en az 5 satır kendin yaz.

Tuple\_2.index(“Alpaslan”, 2) dediğimizde ikinci apoyu arayacaktır.