underscoreها در پایتون میتوانند معانی متفاوتی داشته باشند.

اولین استفاده از underscoreها در پایتون زمانی هستش که شما میخواین از یک کلمه کلیدی به عنوان اسم یک آرگومان استفاده کنید. به کد زیر دقت کنید:

def show(name, class):

pass

همونطور که میدونید کلمه  class یک کلمه کلیدی رزرو شده پایتون هستش و شما نمیتونید ازش به عنوان اسم آرگومان استفاده کنید. به همین دلیل میتونید با اضافه کردن یک underscore به آخر کلمه اون رو از حالت رزرو خارج کنید.

def show(name, class\_):

pass

مصرف بعدی underscoreها در پایتون برای زمانی هستش که شما میخواید یه عضو از کلاس رو به شکل private یا protected در بیارید:

class Person:

name = 'amir' #public

\_age = 10 #protected

\_\_height = 170 #private

زمانی که قبل از اسم عضوی از کلاس یک underscore قرار بدین اون عضو به شکل protected خواهد بود. اگر دوتا underscore قرار بدین به شکل private خواهد بود.

نکته : در حالت protected باز هم میتوانم بصورت عادی از متغیر یا تابع استفاده کرد زیرا پایتون از حالت اجبار استفاده نمیکند و این فقط به عنوان یک نشان از طرف برنامه نویس استفاده میشود

Print (Person.\_age)

حالت دیگری هم برای \_(name) و

اما در حالت private دیگر نمیتوان به صورت عادی به متغیر و تابع دسترسی داشت و اگر استفاده شود با ارور روبرو میشوید اما روشی وجود دارد که میتوان بازهم از مقادیر private استفاده کرد مثلا:

Print ( Person.\_Person\_\_height)

که به این روش name mangling گفته میشود

مصرف بعدی زمانی هستش که شما به قبل و بعد از اسم یک متد در کلاس دوتا underscore اضافه کنید:

class Person:

def \_\_init\_\_(self):

pass

متدهایی که به این شکل اسم گذاری شوند به عنوان special method در پایتون شناخته میشوند. تمامی متدهای builtin پایتون به این شکل نامگذاری میشوند. شما به هیچ وجه نباید اسم متدهاتون رو به این شکل قرار بدین.

آخرین مصرف underscoreها در پایتون زمانی هست که شما مقدار متغیری که دارید براتون مهم نیست:

for i in range(10):

print('Hello')

مثال بالا رو اگه اجرا کنید مقدار Hello ده بار براتون اجرا میشه. اما اگه دقت کنید متغیر i که در حلقه for قرار دادید اصلا کاربرد نداره. پس به جاش میتونید به شکل زیر کار کنید:

س

for \_ in range(10):

print('Hello')

مثالی دیگر برای کاربرد بالا :

X = [1 ,2 ,3 ,4 ,5 ]

First , \_ , \_ , \_ , fifth = X

Print (first) #1

Print (fifth) #5

Print (\_) #4

معنی دیگر در فقط در مفسر پایتون وجود دارد به این صورت که آخرین مقداری که به دست آمده است در \_ دخیره میشود و با چاپ \_ به آن مقدار میتوانید دسترسی پیدا کنید