**تعریف برنامه نویسی**:

زبان رابط بین کاربر و سیستم عامل ماشین را زبان برنامه نویسی مینامند.

**----------------------------------------------------------------------------**

**انواع تقسیم بندی زبان های برنامه نویسی**:

-1زبان های سطح بالا،سطح میانه و سطح پایین

-2زبان های مفسری(interpreter) و کامپایلری

زبان های شی‌گرایی و غیرشی‌گرایی-3

**#**پایتون یک زبان سطح بالا ‌، مفسری و شی گرایی می‌باشد

**----------------------------------------------------------------------------**

**زبان های سطح بالا،سطح میانه و سطح پایین:**

این دسته بندی از نظر مدریت حافظه ، خوانایی و سرعت قابل بررسی می‌باشد.

زبان های سطح بالا: سرعت پایین | خوانایی بالا | مدریت حافظه دست زبان

زبان های سطح میانه: سرعت متوسط | خوانایی متوسط | مدریت حافظه میتواند هم در دست کاربر باشد هم در دست زبان

زبان های سطح پایین: سرعت بالا | خوانایی پایین | مدریت حافظه دست کاربر

**----------------------------------------------------------------------------**

زبان های مفسری(interpreter) و کامپایلری:

زبان مفسری (interpreter): کد های نوشته شده توسط مفسر ترجمه میشود به این صورت که خط به خط کد ها خوانده میشوند و در خروجی نمایش داده میشنود تا به ارور برسد یا برنامه تمام شود.

**………. \_\_\_\_\_\_\_\_ ……..\_\_\_\_\_\_\_\_**

**\_\_\_\_\_\_\_\_ ………. \_\_\_\_\_\_\_\_ ……..**

**\_\_\_\_\_\_\_\_ ……….** **\_\_\_\_\_\_\_\_ ……..**

**\_\_\_\_\_\_\_\_** error  **\_\_\_\_\_\_\_\_ ……..**

**\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_ ……..**

زبان کامپایلری: کد های نوشته شده توسط کامپایلر ترجمه میشود(به اصطلاح کد ها کامپایل میشوند) به این صورت که تمام خطوط خوانده میشوند ‌، در صورت وجود خطا ، برنامه اجرا نمیشود در غیر این صورت یک فایل نصبی به کاربر تحویل میدهد .

**\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_**

**\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_**

**\_\_\_\_\_\_\_\_** error **\_\_\_\_\_\_\_\_** exe

**\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_**

**\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_**

**----------------------------------------------------------------------------**

زبان های شی‌گرایی و غیرشی‌گرایی:

در برنامه نوسی شی گرا به جای تعریف توابع و منطق از اشیا و داده ها استفاده میکنیم در این مدل از برنامه نویسی هر شی به عنوان یک مدل داده ای در نظر گرفته میشود که دارای ویژگی های منحصر به فرد می‌باشد و زبان های غیر شی گرایی این ویژگی را ندارند

**----------------------------------------------------------------------------**

الگوریتم: حل کردن گام به گام مسئله تا رسیدن به خروجی به مطلوب

نکته: نمیتوان در هر مرحله از الگوریتم بیش از یک کار را انجام دهیم و باید مراحل دور از ابهام باشد.

**-**شروع

**-**باقی مراحل

**-**پایان

**----------------------------------------------------------------------------**

الگوریتم بررسی سن قانونی:

1-شروع

2-ذخیره سن کاربر در x

3-اگر x>18:

چاپ کن به سن قانونی رسیده ای

4-در غیر این صورت:

چاپ کن به سن قانونی نرسیده ای

**5-**پایان

الگوریتم دم کردن چای:

1-شروع

2-آب در سماور یا کتری یا چای ساز یا ... بریزید

3-سماور یا اجاق گاز یا ... روشن کنید

4-منتظر بمانید تا آب به جوش بیاید

5-اگر آب به جوش آمده به مرحله 6 بروید در غیر این صورت به مرحله 4 بروید

6-چای خشک در قوری یا فلاکس یا ... بریزید

7-آب جوشیده را داخل قوری یا ... بریزید

8-چند دقیقه صبر کنید تا چای دم بکشد

7-پایان

چاپ کن به سن قانونی نرسیده ای

**-**پایان

**----------------------------------------------------------------------------**

**ابزار های برنامه نویسی با پایتون:**

نصب نسخه مناسب پایتون از سایت [python.org](https://www.python.org/)

#فعال کردن تیک add python to path در صفحه اول فایل نصبی پایتون واجب است

نصب وی‌اس کد از سایت [code.visualstudio.com](https://code.visualstudio.com/)

نصب افزونه [python](https://marketplace.visualstudio.com/items?itemName=ms-python.python) از بخش extensions وی اس کد

**----------------------------------------------------------------------------**

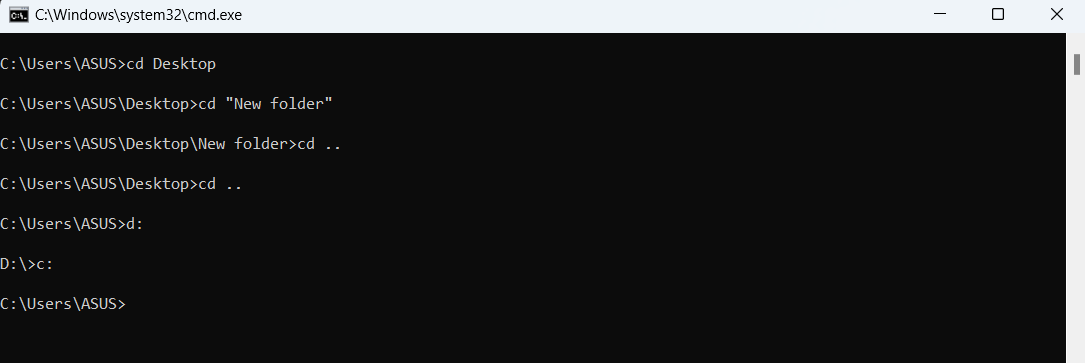
ترمینال: ترمینال یا محیط تعاملی،محیطی است که به وسیله آن میتوان به طور مستقیم به سیستم عامل ماشین دستور داد

win+r ----> write(cmd)

با نوشتن دستور python --version نسخه پایتون نصب شده روی سیستم عامل شما در خروجی نمایش داده میشود

درصورت نصب نداشتن پایتون در خروجی برای شما ارور چاپ می‌شود

دستور cd: دستور cd مخفف کلمه change directoryبه معنای تغیر مسیر می باشد با این دستور میتوانیم در مسیر های کامپیوتر جابه‌جا شویم



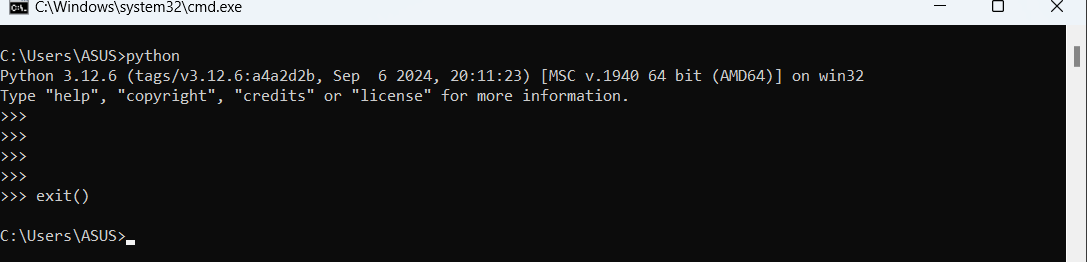
تغیر مسیر cd name\_directory

رفتن به مسیر قبلی cd ..

رفتن به درایو دیگر drive:

#با نوشتن دستور python در ترمینال میتوان مستقیم کد پایتون نوشت و برای خروج از آن حال میتوانیم از دستور exit() استفاده کنیم

#دو مشکل ذخیره نشدن کد ها در ترمینال و عدم توانایی در تغیر کد ها باعث شده که ما به سمت کد ادیتور هایی مثل vscode بریم



**----------------------------------------------------------------------------**

**اجرا فایل پایتون در ترمینال:**

python name\_file

در صورت نبود فایل شما در مسیر ترمینال با ارور مواجه میشوید

باید اسم فایل را با پسوند آن نوشت

#پسوند فایل های پایتون py می‌باشد

# پسوند py پسوند معمولی تر و رایج تر است اما با این حال برنامه نویس ها میتونن پسوند فایل پایتونی خودشون رو .pyw هم بزارن این پسوند مال زمانی هست که یک برنامه گرافیکی درست کرده اید و میخواهید که محیط ترمینال همزمان با محیط گرافیکی شما باز نشود

یک مثال برای درک موضوع بالا:

اگه ما تلگرام رو با پایتون بسازیم میتونیم به دو صورت .py و .pyw ذخیره کنیم که اتفاقات زیر رخ میده:

.py ---> اگه تلگراممون با این پسوند باشه هر کسی که اون رو بازش کنه هم تلگرام براش باز میشه هم ترمینال یعنی کاربر میتونه ببینه که پشت صحنه داره چه اتفاقی می افته

.pyw ---> اگه تلگراممون با این پسوند باشه هر کسی که اون رو بازش کنه فقط محیط گرافیکی تلگرام رو میبیه

در صورت اینکه پسوند فایل ها برای شما نشان داده نمیشود میتوانید با طی کردن مراحل زیر و زدن تیک file name extention این مشکل را رفع کنید

یا به مسیر زیر بروید و تیک گزینه

Hide extensions for known file types

را بردارید

control panel

File Explorer Options or Folder Options

View Tab

Hide extensions for known file types

Apply

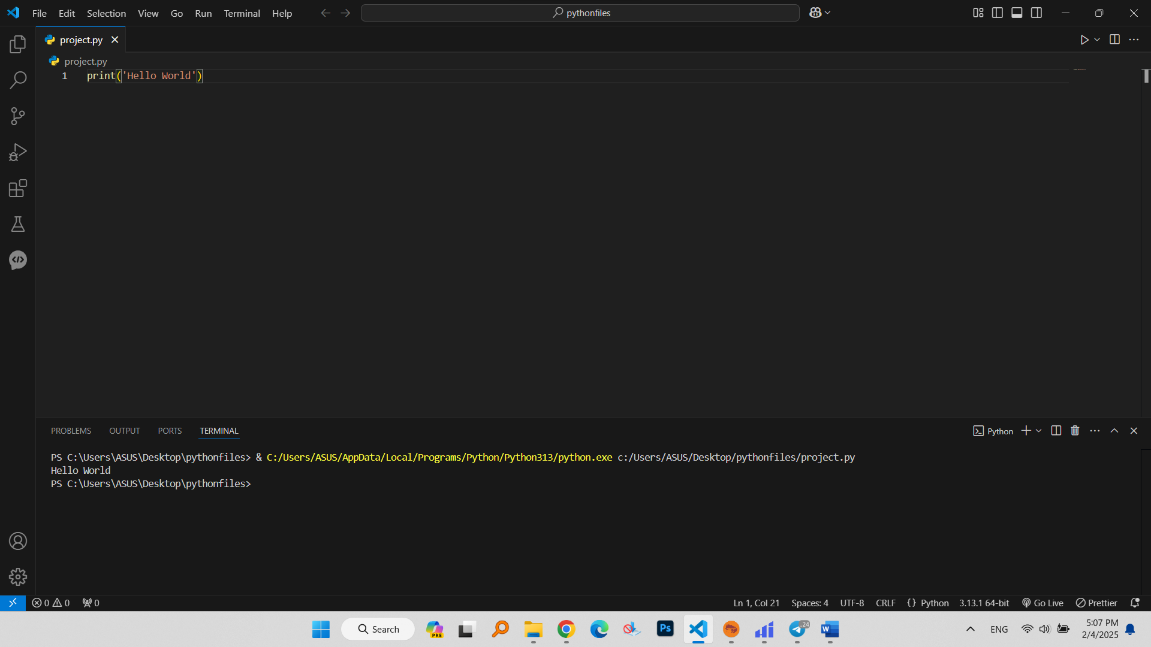
Ok

**----------------------------------------------------------------------------**

**انواع محیط های کدزنی:**

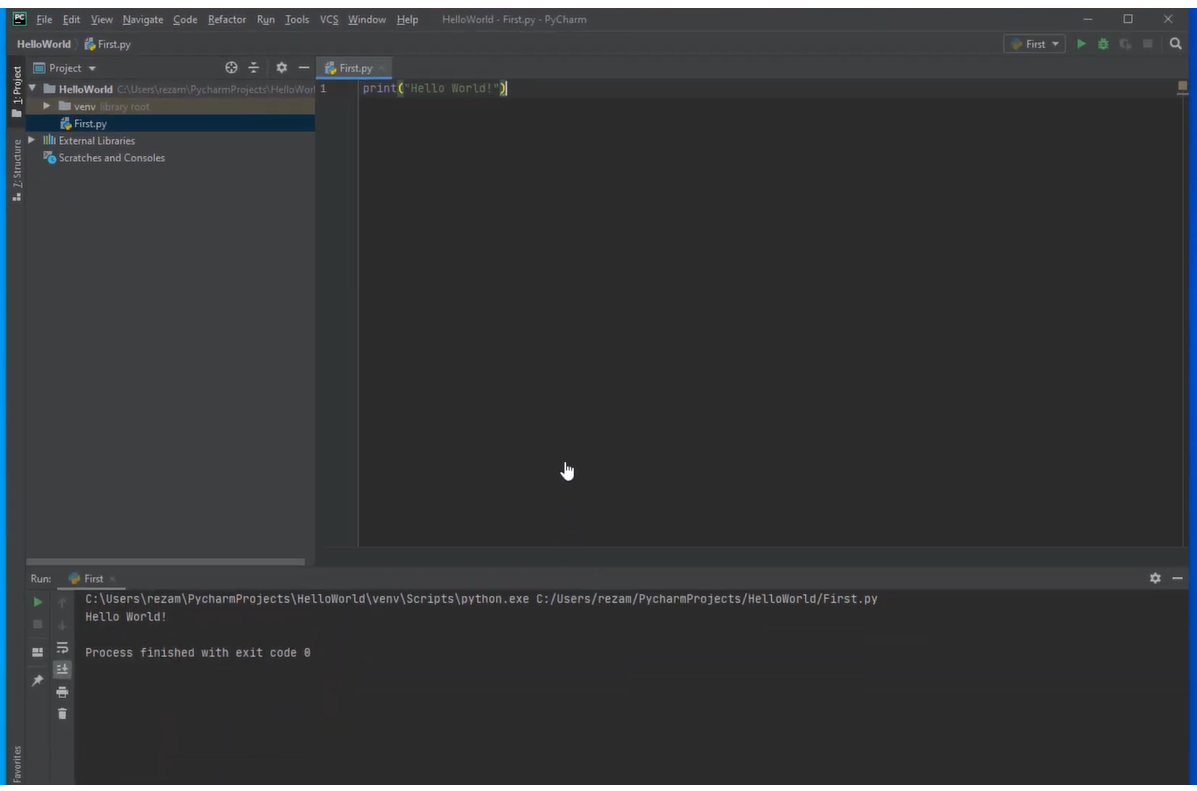
کد ادیتور ها مثل وی اس کد محیط هایی هستند که ما :CodeEditor

میتونیم داخلشون کد های زبان های مختلف رو بنویسیم مثلا تو وی اس کد هم میتونیم کد پایتونی بزنیم هم کد جاوایی هم کد جاوااسکریپتی و ... بزنیم



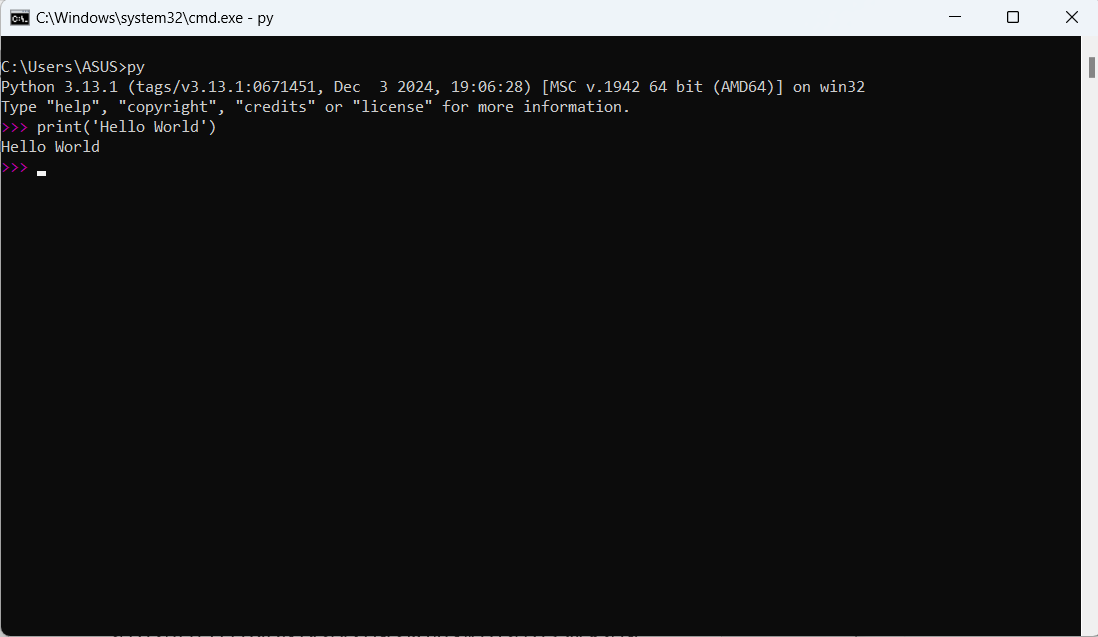
آی دی ای ها محیط هایی هستند که اختصاصی برای یک زبان :IDE

درست شده اند مثل آی دی ای پایچارم برای زبان پایتون



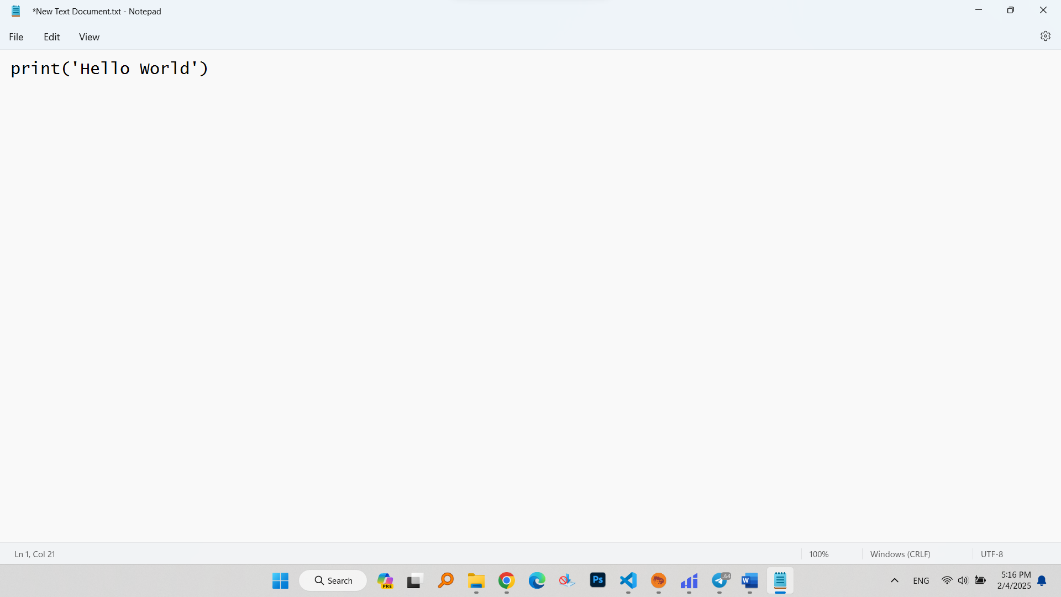
ترمینال ها محیط هایی مثل سی ام دی ویندوز هستند :Terminal

که ما میتونیم به صورت مستقیم کد بزنیم و خروجی رو در لحظه ببینیم ولی قابلیت ذخیره سازی و ادیت کد هارو ندارن



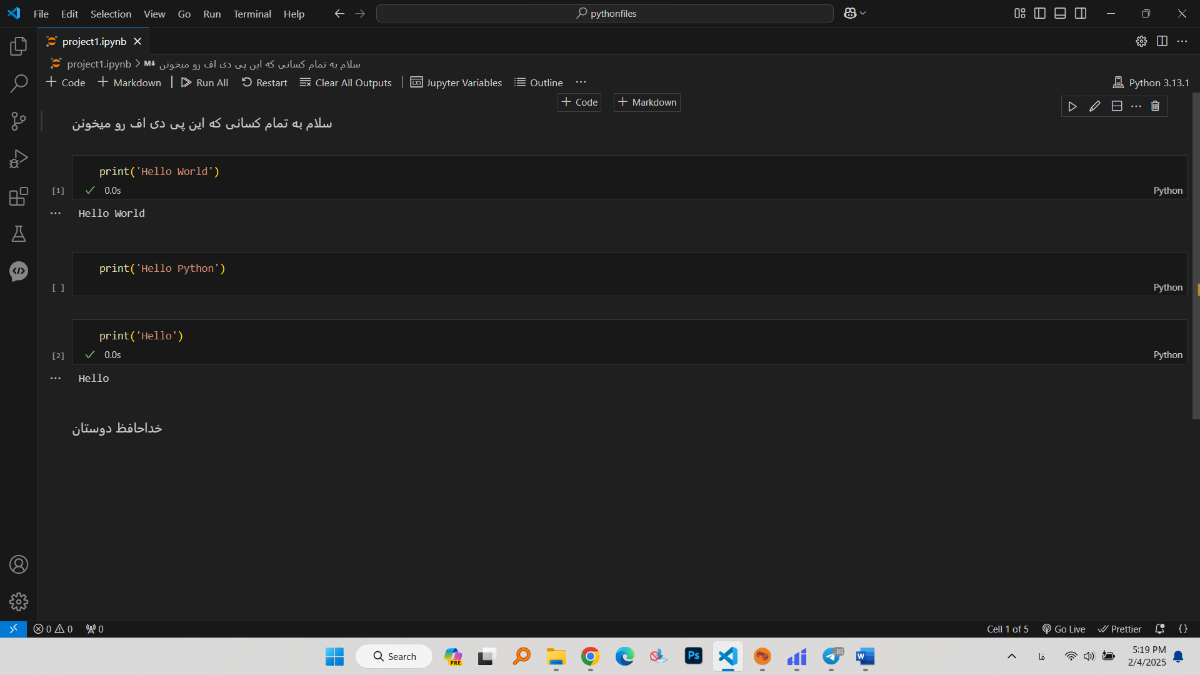
تکس ادیتور ها محیط هایی مثل نوت پد هستند که مزایا:TxtEditor

های خاصی برای برنامه نویس ها ندارن و فقط صرفا یک فایل برای ذخیره و ویرایش کد ها هستند



در نوت بوک ها میتونیم تو سلول های مشخص شده :NoteBook

کد بزنیم و در صورت نیاز میتونیم به جای اجرا کردن کل برنامه فقط یک سلول رو اجرا کنیم و خروجی رو در همون لحظه ببینیم هم چنین قابلیت این رو دارند که ما داخلشون از متن ها و عکس ها هم استفاده کنیم



**----------------------------------------------------------------------------**

متغیر variable: متغیر مانند ظرفی است که در آن میتوانیم داده ذخیره کنیم با استفاده از عملگر انتساب( = ) میتوانیم داده های سمت راست عملگر را در متغیر سمت چپ عملگر ذخیره کنیم

Variable1 = dita type1

|  |  |  |  |
| --- | --- | --- | --- |
| RAM memory | | | |
| Variable1 | Variable2 | Variable3 | Name variable |
| data type1 | data type2 | data type3 | Variable value |

اسم متغیر در دست برنامه نویس است و میتواند هر چیزی باشد(کار پسندیده آن است که مناسب با داده نام متغیر را انتخاب کنیم)

age = 25

abcde = 25

#هر دو تعریف متغیر بالا درست و بدون خطا می باشد اما تعریف متغیر به روش اول پسندیده تر است

**نباید ها:**

نام یک متغیر نباید با عدد شروع شود 5age = 17

در نام یک متغیر نباید اسپیس وجود داشته باشد age ali = 44

در نام یک متغیر نباید کارکتر خاص مثل  !@#.$%^&\*وجود داشته باشد age$ = 33

نام متغیر نباید کلمات خاص پایتونی باشد print = 29

**باید ها:**

نام یک متغیر میتواند دارای آندرلاین باشد age\_ali = 30

در وسط یا آخر نام متغیر میتواند عدد وجود داشته باشد age1 = 19

# نام متغیر ها به حروف کوچک و بزرگ حساس هستند برای مثال دو نام Age و age با هم متفاوتند.

رفتار مفسر با دو تا متغیر هم نام:

مفسر همیشه آخرین مقدار یک متغیر رو در نظر میگیره برای مثال تو کد زیر تا خط 3 آخرین مقدار متغیر age عدد 20 هستش پس مفسر تو خط 1 و 2 و 3 این متغیر رو با عدد 20 در نظر میگیره اما تو خط 4 مقدار درون متغیر تغیر کرده و از خط 4 به بعد آخرین مقدار این متغیر عدد 5 خواهد بود

1 age = 20

2

3

4 age = 5

**----------------------------------------------------------------------------**

**دیتا تایپ ها:**

-1عدد صحیح integer به اختصار int:

number = 18

-2عدد اعشاری float به اختصار float:

number1 = 18.5

-3متن string به اختصار str:

متن ها در پایتون بین دو کوتیشن"text" یا دو آپاستروف'text' قرار میگیرند

name = 'alireza'

name = "alireza"

**----------------------------------------------------------------------------**

**کامنت ها و داک استریتگ ها:**

کامنت:قطعه ای از کد است که نه خوانده میشود و نه اجرا میشود برنامه نویس میتواند با کامنت گذاری توضیحاتی در باره کد خود بدهد.با علامت # میتوان در پایتون کامنت گذاری کرد

در خط زیر سن کاربر را ذخیره میکنیم#

age **=** 25

age **=** 25 در این خط سن کاربر را ذخیره میکنیم#

**داک استرینگ:** قطعه ای از کد است که هم خوانده میشود و هم اجرا میشود برنامه نویس میتواند با داک استرینگ گذاشتن توضیحاتی در باره نحوه اجرای کد خود بدهد.

داک استرینگ بین 3 آپاستروف ''' docstring ''' یا 3 کوتیشن

""" docstring """قرار میگیرد.

""" در خط زیر سن کاربر را بعلاوه 2 میکنیم """

age **=** 25

**تفاوت استرینگ و داک استرینگ**:

داک استرینگ به صورت مستقل تعریف میشود و استرینگ در متغیر ذخیره میشود.

""" #DokString **در خط زیر سن کاربر را ذخیره میکنیم**"""

""" #String**در خط زیر سن کاربر را ذخیره میکنیم**""" a **=**

**----------------------------------------------------------------------------**

**خط فیزیکی و منطقی:**

خط فیزیکی:بر اساس خطوط نوشته شده و از دید بصری است.

خط منطقی**:** بر اساس دستور های نوشته شده و از دید مفسر است.

خطوط منطقی را میتوان با \ در چند خط فیزیکی نوشت

خطوط منطقی را میتوان با ; در یک خط فیزیکی نوشت

#یک خط فیزیکی و یک خط منطقی

a = 20+5-17-2\*9+3

------------------------------------

#سه خط فیزیکی و یک خط منطقی

a = 20+5-\

    17-2\*\

    9+3

------------------------------------

# دو خط فیزیکی و دو خط منطقی

print(16)

print("alireza")

------------------------------------

#یک خط فیزیکی و دو خط منطقی

print(16); print("alireza")

----------------------------------------------------------------------------

نکته: ما نمیتوانیم که یک خط منطقی را به چند خط فیزیکی تقسیم و بعد در همان خط کامنت بگذاریم مانند مثال زیر:

a = 20+5-\ #سه خط فیزیکی و یک خط منطقی

    17-2\*\

    9+3

دلیل: در اصل مقدار درون متغیر a به شکل زیر می باشد:

a = 20+5-17-2\*9+3

و ما بعد از 20+5- کامنت گذاشته ایم که عبارت به شکل زیر میشود:

a = 20+5-#سه خط فیزیکی و یک خط منطقی 17-2\*9+3

و چون کامنت ما انتهای خط نیست باعث ارور میشویم

در کل زمانی که میخواهید با خط های فیزیکی و منطقی و قابلیت هایشان کارکنید کامنت هایتان را باید خط قبل دستورات خود بنویسید تا به مشکل نخورید به این شکل:

#سه خط فیزیکی و یک خط منطقی

a = 20+5-\

    17-2\*\

    9+3