

تعریف برنامه نویسی:

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

انواع تقسیم بندی زبان های برنامه نویسی:

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

2- زبان های مفسری (interpreter) و کامپایلری

3- زبان های شی گرای و غیرشی گرای

پایتون یک زبان سطح بالا، مفسری و شی گرای می‌باشد

زبان های سطح بالا، سطح میانه و سطح پایین:

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

زبان های سطح بالا: سرعت پایین | خوانایی بالا | مدیریت حافظه

دست زبان

زبان های سطح میانه: سرعت متوسط | خوانایی متوسط | مدیریت

حافظه میتواند هم در دست کاربر باشد هم در دست زبان

زبان های سطح پایین: سرعت بالا | خوانایی پایین | مدیریت حافظه

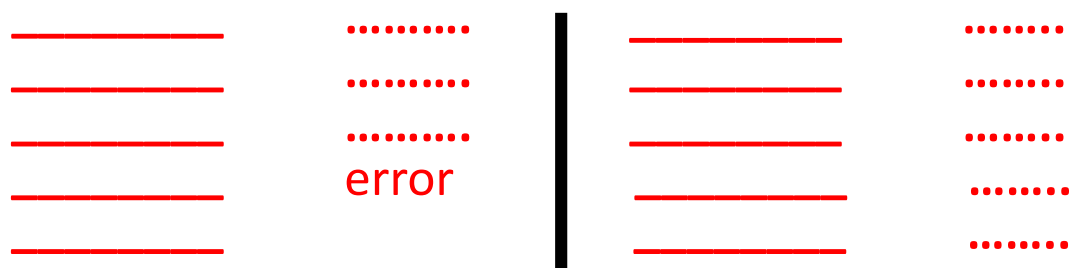
دست کاربر

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 میشوند تا به ارور برسد یا برنامه تمام شود.

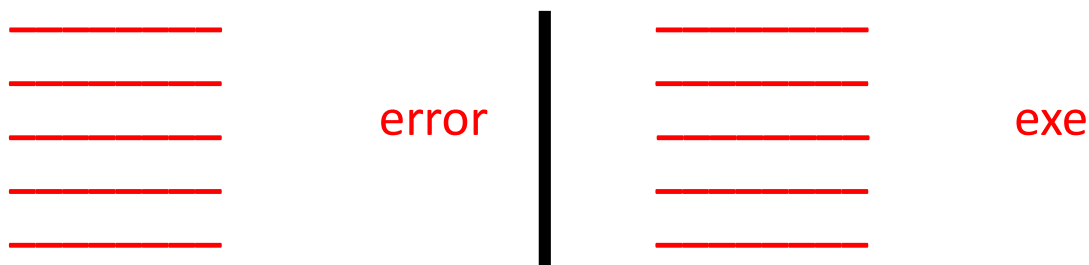


زبان کامپایلری: کد های نوشته شده توسط کامپایلر ترجمه

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 .



زبان های شی گزایی و غیرشی گزایی:

در برنامه نویسی شی گزایی 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 می باشد و زبان های غیرشی گزایی این ویژگی را ندارند

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

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

- شروع

- باقی مراحل

- پایان

الگوریتم بررسی سن قانونی:

۱- شروع

۲- ذخیره سن کاربر در x

۳- اگر $x > 18$:

چاپ کن به سن قانونی رسیده ای

۴- در غیر این صورت:

چاپ کن به سن قانونی نرسیده ای

۵- پایان

الگوریتم دم کردن چای:

۱- شروع

۲- آب در سماور یا کتری یا چای ساز یا ... پریزید

۳- سماور یا اجاق گاز یا ... روشن کنید

۴- منتظر بمانید تا آب به جوش بیاید

۵- اگر آب به جوش آمده به مرحله ۶ بروید در غیر این صورت به

مرحله ۴ بروید

۶- چای خشک در قوری یا فلاکس یا ... پریزید

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

۸- چند دقیقه صبر کنید تا چای دم بکشد

۷- پایان

چاپ کن به سن قانونی نرسیده ای

- پایان

ابزارهای برنامه نویسی با پایتون:

نصب نسخه مناسب پایتون از سایت python.org

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

پایتون واجب است

نصب وی اس کد از سایت code.visualstudio.com

نصب افزونه [python](https://python.org) 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)

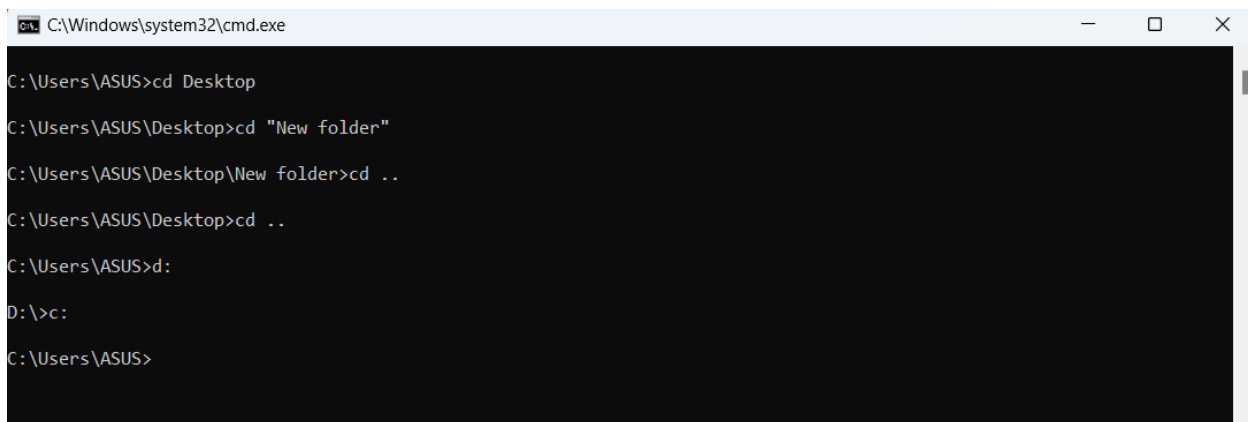
با نوشتن دستور python --version نسخه پایتون نصب شده روی
سیستم عامل شما در خروجی نمایش داده میشود

در صورت نصب نداشتن پایتون در خروجی برای شما ارور چاپ می شود

دستور cd: دستور cd مخفف کلمه change directory به معنای تغییر

مسیر می باشد با این دستور میتوانیم در مسیر های کامپیوتر

جابه جا شویم



```
C:\Windows\system32\cmd.exe
C:\Users\ASUS>cd Desktop
C:\Users\ASUS\Desktop>cd "New folder"
C:\Users\ASUS\Desktop\New folder>cd ..
C:\Users\ASUS\Desktop>cd ..
C:\Users\ASUS>d:
D:\>c:
C:\Users\ASUS>
```

cd name_directory

تغییر مسیر

cd ..

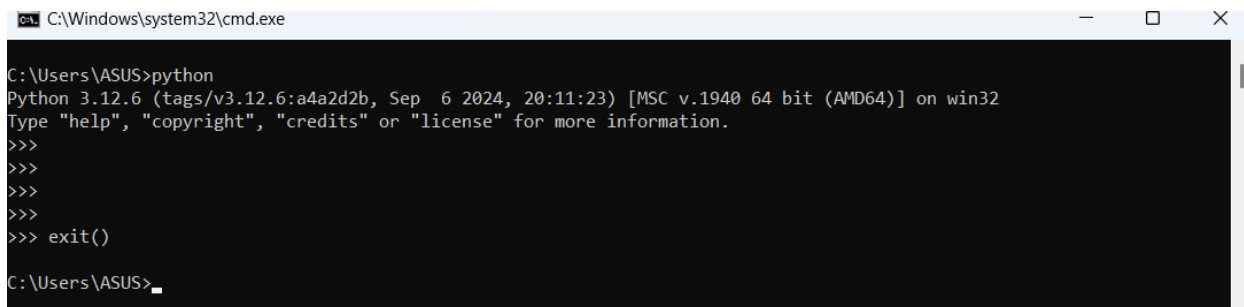
رفتن به مسیر قبلی

drive:

رفتن به درایو دیگر

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 تغییر کد ها باعث شده که ما به سمت کد ادیتور هایی مثل vscode پریم

A screenshot of a Windows command prompt window titled "C:\Windows\system32\cmd.exe". The prompt shows the user typing 'python' at the C:\Users\ASUS> prompt. The output displays the Python version (3.12.6) and build information. The user then enters three empty lines followed by 'exit()', and the prompt returns to C:\Users\ASUS>.

```
C:\Windows\system32\cmd.exe
C:\Users\ASUS>python
Python 3.12.6 (tags/v3.12.6:a4a2d2b, Sep  6 2024, 20:11:23) [MSC v.1940 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>>
>>>
>>>
>>> exit()
C:\Users\ASUS>
```

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 می باشد

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

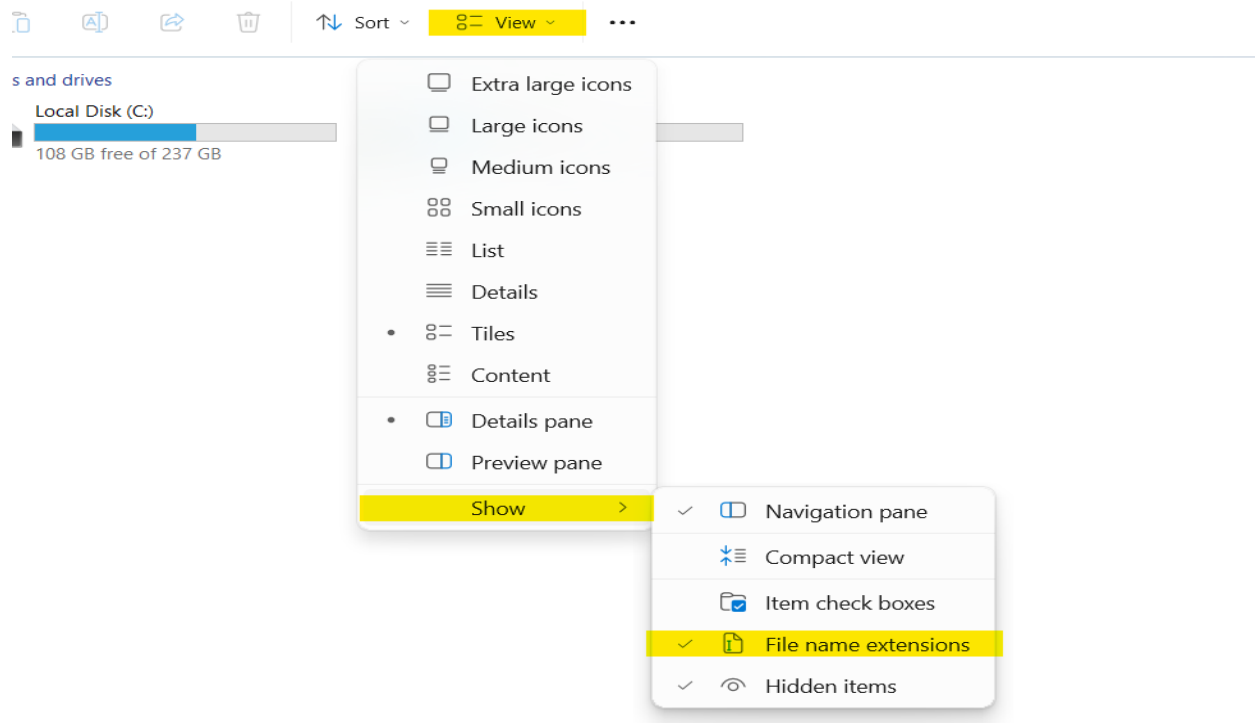
یک مثال برای درک موضوع بالا:

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 میبینه

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

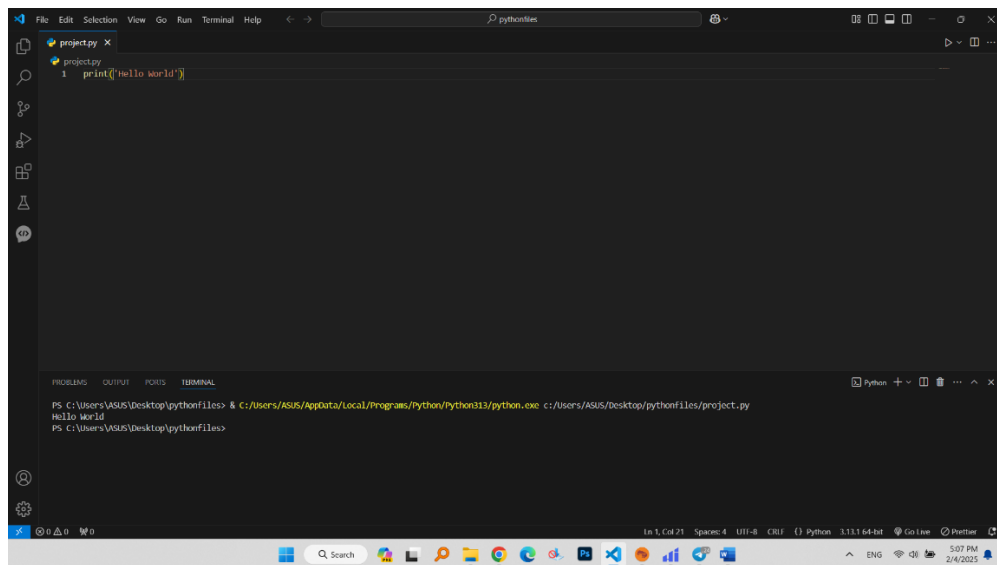
Hide extensions for known file types

Apply

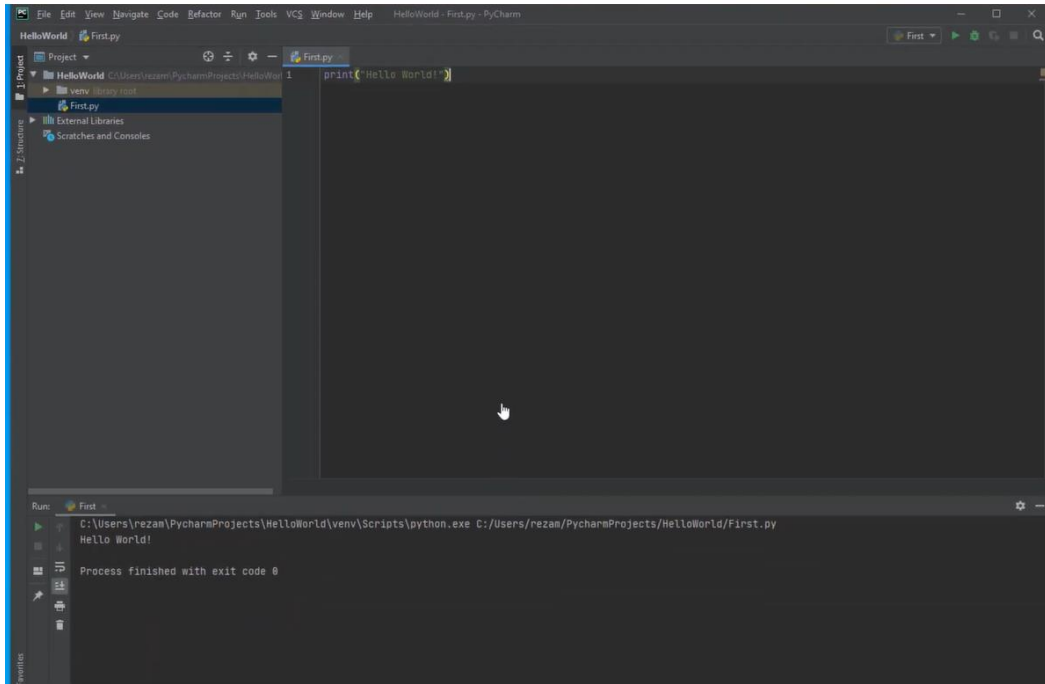
Ok

انواع محیط های کد زنی:

CodeEditor: کد ادیتورها مثل وی اس کد محیط هایی هستند که ما می‌تونیم داخلشون کد های زبان های مختلف رو بنویسیم مثلاً تو وی اس کد هم می‌تونیم کد پایتونی بنویسیم هم کد جاوایی هم کد جاوااسکریپتی و ... بنویسیم

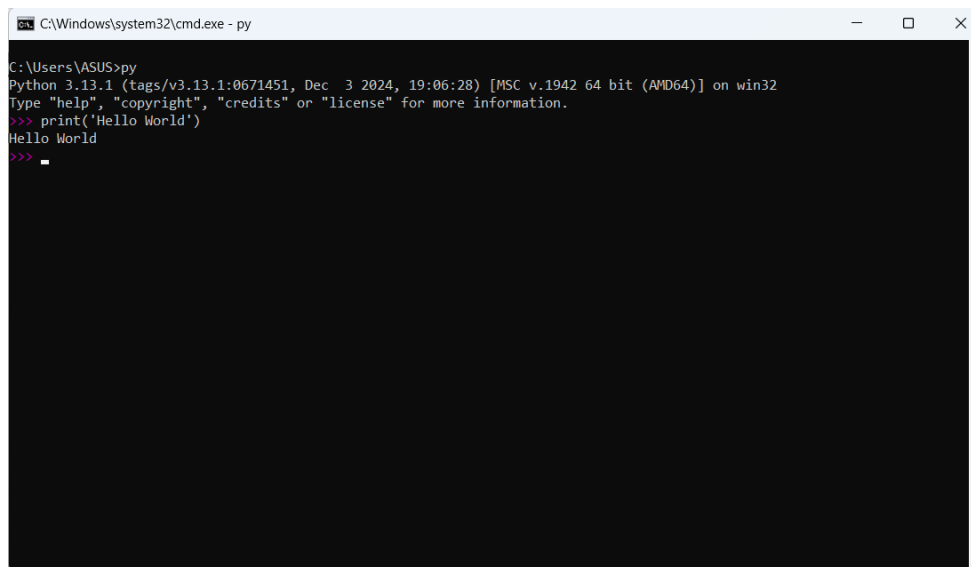


IDE: آی دی ای ها محیط هایی هستند که اختصاصی برای یک زبان درست شده اند مثل آی دی ای پایچارم برای زبان پایتون



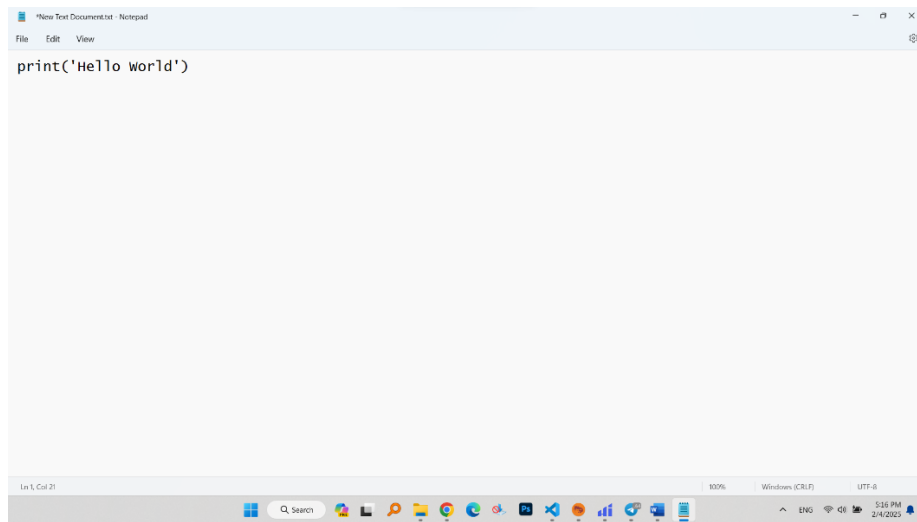
Terminal: ترمینال ها محیط هایی مثل سی ام دی ویندوز هستند

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



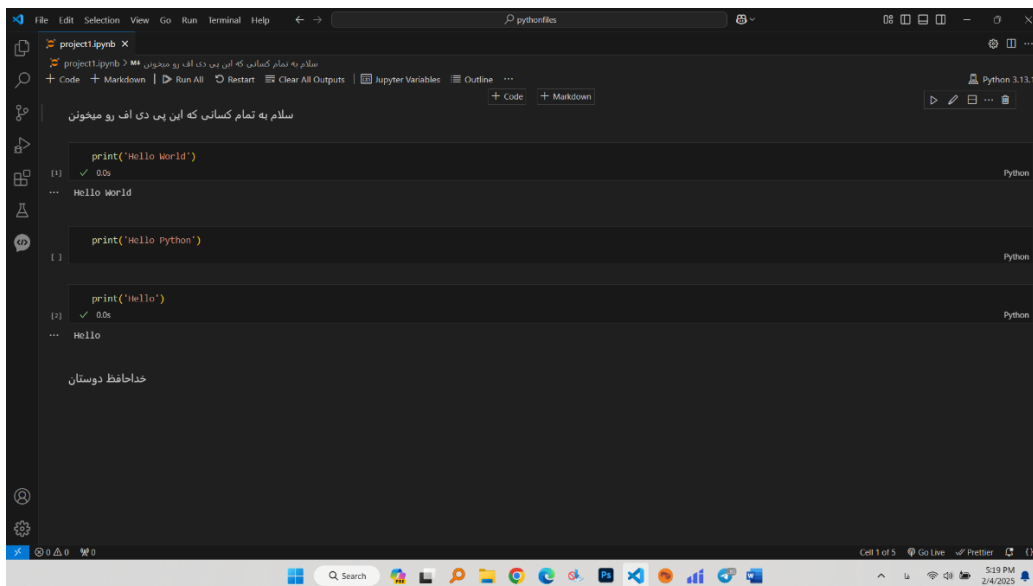
TextEditor: تکس ادیتور ها محیط هایی مثل نوت پد هستند که مزایا

های خاصی برای برنامه نویسی ها ندارند و فقط صرفا یک فایل برای ذخیره و ویرایش کد ها هستند



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

کد بنویسیم 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 یا استفاده از عملگر انتساب (=) میتوانیم داده های سمت راست عملگر را در متغیر سمت چپ عملگر ذخیره کنیم

Variable1 = data type\

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:

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

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

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

print = 29 نام متغیر نباید کلمات خاص پایتونی باشد

باید ها:

age_ali = 30 نام یک متغیر میتواند دارای آندرلاین باشد

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

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 ۳ آخرین مقدار متغیر age عدد ۲۰ هستش پس
مفسر تو خط ۱ و ۲ و ۳ این متغیر رو با عدد ۲۰ در نظر میگیره اما تو خط
۴ مقدار درون متغیر تغییر کرده و از خط ۴ به بعد آخرین مقدار این
متغیر عدد ۵ خواهد بود

1 age = 20

2

3

4 age = 5

دیتا تایپ ها:

1- عدد صحیح integer به اختصار int:

number = 18

2- عدد اعشاری float به اختصار float:

number1 = 18.5

3-متن 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 استرینگ ها:

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

در خط زیر سن کاربر را ذخیره میکنیم #

```
age = 25
```

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

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

داک استرینگ بین 3 آپاستروف " docstring " یا 3 کوتیشن "" docstring "" قرار میگیرد.

"" در خط زیر سن کاربر را علاوه 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: هر اساس خطوط نوشته شده و از دید بصری است.

خط منطقی: هر اساس دستورهای 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:

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

$17-2*\backslash$

$9+3$

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:

$$a = 20+5-\backslash \text{سه خط فیزیکی و یک خط منطقی} \#$$

و چون کامنت ما انتهای خط نیست باعث ارور میشویم

در کل زمانی که میخواهید با خط های فیزیکی و منطقی و قابلیت

هایشان کار کنید کامنت هایتان را باید خط قبل دستورات خود

بنویسید تا به مشکل نخورید به این شکل:

سه خط فیزیکی و یک خط منطقی #

$$a = 20+5-\backslash$$

$$17-2*\backslash$$

$$9+3$$