جزوه شماره ٦

PYTHON

ArcGIS Pro Notebook

محمدطاهر طاهرپور

دانشجوی ارشد مدیریت شهری

دانشگاه تهران

mttaherpoor
@mtaherpoor
۰۹۳۳٦۱٤٤۹٤۷

ناهید نعمتی کوتنائی (تیسا)

دکتری جغرافیا و برنامه‌ریزی شهری

برنامه‌ریز شهری و تحلیلگر GIS

Dr.nemati.K
@Nemati_k
۰۹۱۱۲۲۳۰۷۹۸

تابستان ۱٤۰۲

فهرست مطالب:

Fundamentals of Python

تو این جزوه اصول پایه پایتون رو با هم مرور می‌کنیم و از نوت بوک ArcGIS Pro واسه یادگیری بهتر کدها استفاده میکنیم. انتهای درسها چند تا تمرین به همراه جوابهاشون هم واسه یادگیری بهتر آوردیم. در انتهای جزوه هم چند تا از کتابخونه‌های مهم پایتون به همراه کاربردهاشون رو لیست کردیم.

مطالب این جزوه خلاصه مطالب Introduction to Python از سایت سولورن و Data Visualization with Python از سایت Udemy هست.

پایتون چیه؟

پایتون یه زبان برنامه‌نویسی محبوب، کاربردی و خیلی قدرتمند هست و خیلی آسون میشه یادش گرفت، چون سینتکس ساده داره یعنی هم آسون نوشته میشه و هم آسون خونده میشه. ساختار داده یا Data structure پویا یا Dynamic و کتابخونه یا Libraryهای بزرگ داره. این زبان رو تو کل دنیا به بقیه زبانها ترجیح میدن.

ما از این زبان استفاده میکنیم که بتونیم به ماشین دستورالعمل بدیم. هر دستورالعمل شامل یه سری عبارت یا Statement میشه که کامپیوتر باید پیگیریشون کنه. از پایتون میشه در زمینه نرم‌افزار و توسعه وب، علم داده یا Data science، یادگیری ماشین یا Machine Learning و زمینه‌های زیاد دیگه‌ای استفاده کرد.

معرفی چند منبع یادگیری پایتون

پیش فرضمون اینه که با زبان برنامه‌نویسی پایتون آشنایی داری. اگه اصلا با این زبان آشنایی نداری میتونی از سایت سولورن، W۳Schools یا Udemy استفاده کنی و آموزش حرفه‌ای ببینی.

سایت سولورن، هم آموزش‌هاش و هم گواهینامه پایان دوره‌هاش رایگان هست. سایت W۳Schools آموزش‌های رایگان داره ولی واسه گرفتن گواهی پایان دوره باید هزینه کنی.

https://www.sololearn.com/
https://www.w۳schools.com/
https://www.udemy.com/

از این لینک هم میتونی آموزش‌های Udemy که فوق‌العاده هستن رو رایگان دانلود کنی:

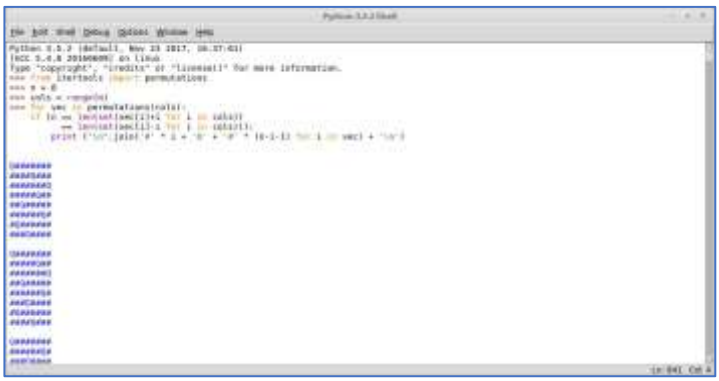
https://downloadly.ir

Python Interfaces in ArcGIS Pro

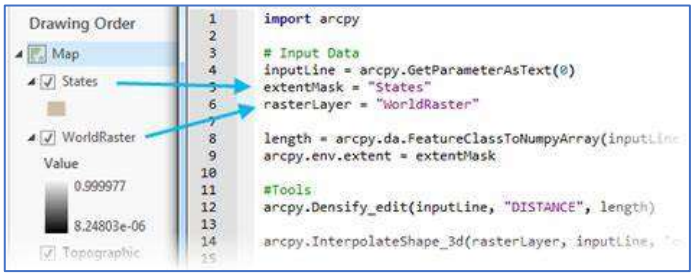
به پنج شیوه زیر می‌تونی تو ArcGIS Pro کد پایتون بنویسی و ازش استفاده کنی:

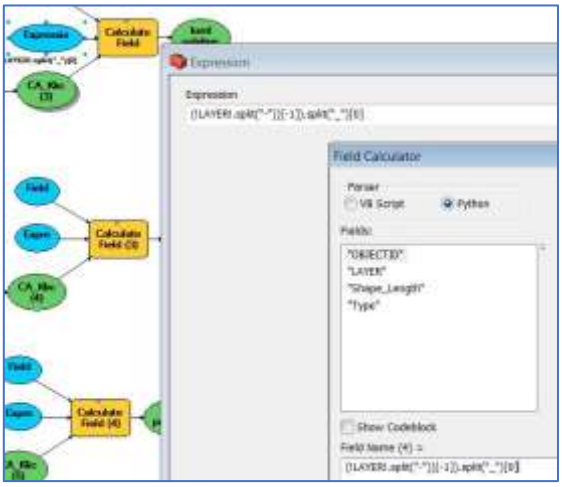
۱- استفاده از **Python window**

۲-استفاده از **Python IDE** (برای مثال IDLE) برای ساخت Scriptها خارج از ArcGIS Pro



۳- استفاده از arcpy.GetParameterAsText() در **Python Script Tool**





٤- استفاده از **Python Parser** برای برچسب زدن یا labeling، محاسبات روی فیلدها تو جدول اطلاعاتی یا Field Calculator و محاسبات مقادیر در ساخت مدل یا ModelBuilder

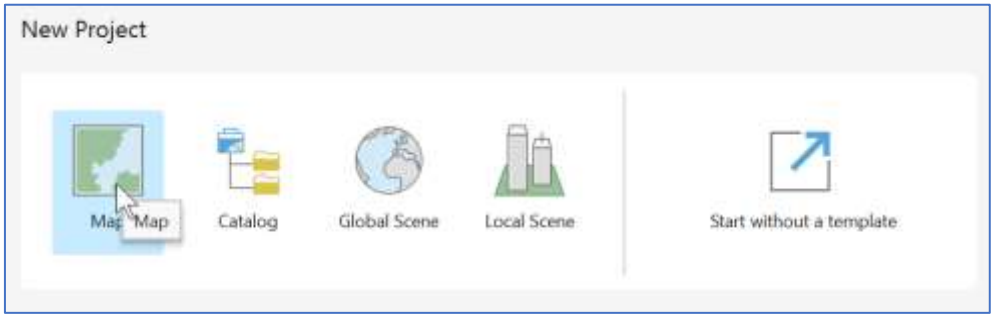
۵- استفاده از **ArcGIS Nootbooks** که بر اساس نوت بوک ژوپیتر هست.

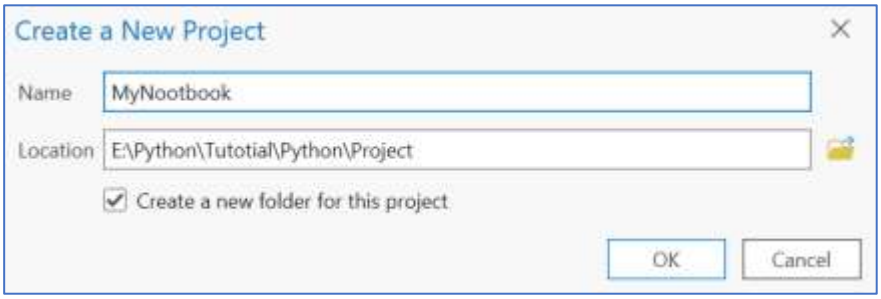
در مورد کار با همه این بخشها مطالب آموزشی آماده می‌کنیم، ولی تو این جزوه صرفاً می‌خوایم کدهای پایه پایتون رو تو نوت بوک ArcGIS Pro بنویسیم و تمرین کنیم.

اگه به زبان پایتون آشنایی داری، بریم که نکات پایه‌اش رو با هم دوره کنیم و ببینیم که چطوری میشه از NootBook تو ArcGIS Pro واسه تمرین کدهایی که یاد می‌گیریم استفاده کنیم.

مرحله اول: ساخت پروژه برای کار با Notebook در ArcGIS Pro

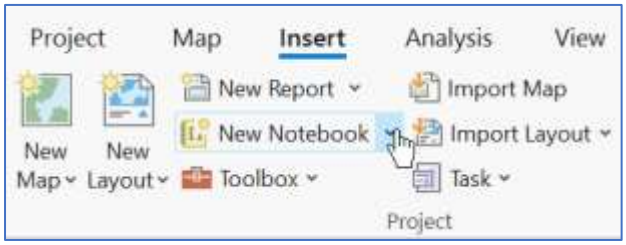
نرم افزار ArcGIS Pro باز کن. تو صفحه خوش‌آمد گویی روی گزینه Map کلیک کن که یه پروژه جدید بسازی.

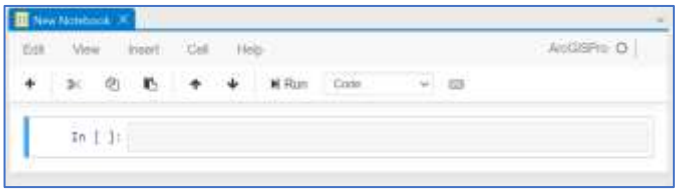




بهش اســـم و آدرس خروجی بده و OK کن که صــفحه نرم‌افزار باز شه.

از سربرگ Insert وارد بخش project شو و روی گزینه New Notebook کلیک کن که پنجره‌اش برات باز شه.





این نت‌بوک، cell یا سلولهای مختلف داره. میتونی تو هر سلول یه چیزی تعریف کنی که میتونن به صورت جداگونه اجرا یا Run بشــن. این ســلولها امکان جابجایی دارن و کل کدهات رو به همراه اجراشــون میتونی تو یه فایل ذخیره داشته باشی. در مورد این محیط جذاب تو ArcGIS Pro تو جزوه‌های بعدی بیشتر توضیح میدیم.

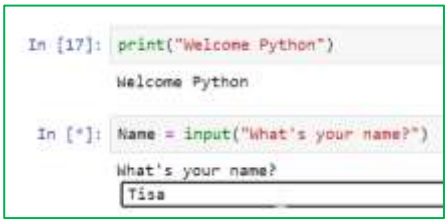
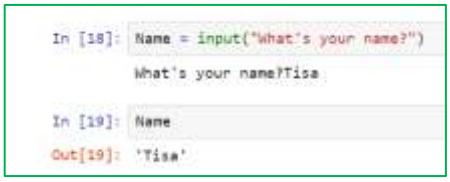
مرحله دوم: مرور اصول پایه زبان برنامه‌نویسی Python

تو جدولهای زیر توضـیحات کلی در مورد اصـول پایه پایتون آوردیم. چیزهایی که یاد می‌گیری رو حتما توی سلولهای نوت‌بوک بنویس که برات تمرین شه.

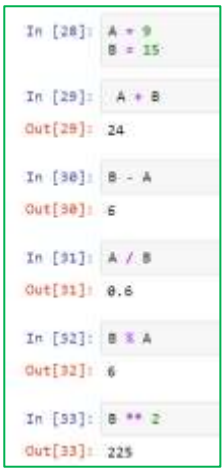
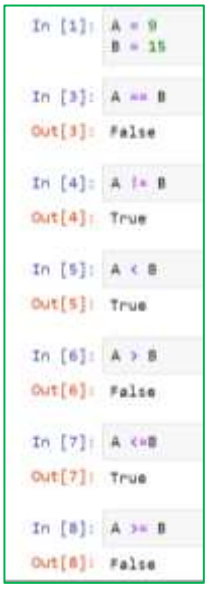
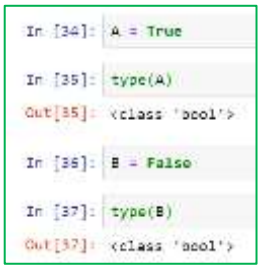
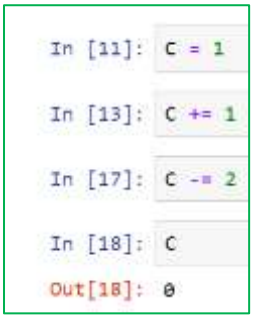
واسه رفتن به سلول بعدی، شیفت و اینتر رو بزن که کد سلولی که توش هستی رو اجرا کنه و بره تو یه سلول جدید.

عنوان	توضیحات	چطوری تعریف میشه؟
متغیرها یا Variables	✓ برای ذخیره کردن مقادیر (مثل اعداد، متنها و ...) استفاده میشه که بشه بعدا از این مقادیر در طول برنامه استفاده کرد. چون چیزی رو ذخیره میکنه نیاز به حافظه یا فضا داره. ✓ نیازی به دستور اعلان نداره و فقط شامل اسم و مقدارش هست که با علامت = به هم وصل میشن. ✓ واسـه **نامگذاری** متغیرها هم محدودیت داری. با عدد شـروع نکن. فقط با حروف یا آندرلاین شـروع کن. از فاصـله یا – بین کلمات استفاده نکن فقط _ بذار. پایتون به حروف کوچیک و بزرگ حساس هست مثلا Age, age, AGE براش سه تا اسم متفاوت هستن. از اسامی که تو پایتون تعریف شده هم استفاده نکن مثل for, while, elif, if و ... واسه نامگذاری اسامی ترکیبی هم به یکی از دو شیوه زیر عمل کن: ➢ Snake case یه روش محبوب واسه ایجاد نام متغیر در پایتون هست. تو این روش از زیرخط یا آندرلاین _ واسه جدا کردن اسامی چند کلمه‌ای در نام متغیر استفاده میشه. first_name ➢ Camel case هم روش محبوب دیگه واسه نامگذاری اسامی ترکیبی هست که اسم اول باحروف کوچیک نوشته میشه و اسم بعدی به صورت چسبیده بهش می‌آد ولی اولش با حرف بزرگ شروع میشه مثل firstName	A = ۱۰ Name = 'John' Mile = ۱۳۰,۷ City_Name = "London" 

عنوان	توضیحات	چطوری تعریف میشه؟
انواع داده یا data types	پایتون ۵ نوع داده داره: ۱- Numbers ۲- Strings ۳- Lists ٤- Tuples ۵- Dictionaries یه تابع داریم به اسم type() که وقتی داخش اسم متغیر رو بنویسیم، نوع داده‌اش رو بهمون میده. یادت باشه که توابع با حروف کوچیک نوشته میشن. ❖ سه نوع داده عددی یا **number** داریم: ۱- **عدد صحیح یا integers** که هم مثبت هست و هم منفی بدون اعشار ۲- **عدد اعشاری یا float** که مثبت و منفی هست و با اعشار. ۳- **عدد پیچیده یا complex** مثل A+Bj که A بخش اصلی و واقعی هست و BJ بخش تخیلی یا تصوری. Complex خیلی تو پایتون کاربرد نداره. ❖ داده رشته یا متنی یا **Strings** ✓ توالی حروف، متن و ... رو مشخص میکنه. ✓ متن (میتونه عدد هم باشه) رو باید بذاری تو کوتیشن " "، ' ' (تکی یا دوتایی فرق نداره فقط این ور و اون ور باید یکسان باشه) که اسمش بشه string یا رشته.	In [6]: type(A) Out[6]: <class 'int'> In [7]: type(Name) Out[7]: <class 'str'> In [8]: type(City_Name) Out[8]: <class 'str'> In [9]: type(Miles) Out[9]: <class 'float'> In [10]: a = 10 In [11]: b = 99.9 In [12]: c = 45.j In [13]: type(a) Out[13]: <class 'int'> In [14]: type(b) Out[14]: <class 'float'> In [15]: type(c) Out[15]: <class 'complex'>

عنوان	توضیحات	چطوری تعریف میشه؟
توابع داخلی یا Built-in در پایتون	✓ ازتابع print() واسه خروجی گرفتن یا ارسال پیام به اسکرین یا سایر وسایل نمایش استفاده میشه. اگه بخوای توی پرانتزش متن بنویسی باید بذاریش تو "" که تبدیل به استرینگ یا رشته بشه. اعداد به علامت نقل قول نیازی ندارن. میتونی اسم یه متغیر که از قبل تعریف شده رو هم بذاری تو پرانتز print() که مقدارش رو بهت خروجی بده. ✓ از تابع input() واسه گرفتن ورودی از کاربر استفاده میشه. **باید براش متغیر تعریف کنی مثل شکل مقابل.** وقتی کد رو با Shift + enter اجرا کنی یه کادر بهت میده که ورودی بهش بدی و بعد از اینکه چیزی نوشتی و اینتر زدی خارج از سلول بهت نشونش میده. اگه بخوای مجدد ببینی که چی وارد شده صرفا اسم متغیر رو تو یه سلول جدا بنویس و اجراش کن. ✓ **نکته:** پیش فرض اطلاعاتی که کاربر وارد میکنه همیشه String هست. یعنی اگه عدد ۳ رو هم وارد کنه باز هم به صورت '۳' که رشته هست شناخته میشه.	 

عنوان	توضیحات	چطوری تعریف میشه؟
تبدیل داده یا data Conversion	گاهی نیاز داری ورودی که از کاربر میگیری رو به نوع دیگه‌ای تبدیل کنی چون با تابع input() کاربر هر چیزی وارد کنه به صـــورت رشـــته یا string برمیگیرده. واسه تبدیل داده‌ها به هم: ✓ دستور int(): تبدیل نوع داده به عدد صحیح ✓ دستور float(): تبدیل نوع داده به عدد اعشاری ✓ دستور str(): تبدیل نوع داده به رشته	

عنوان	توضیحات	چطوری تعریف میشه؟
عملگرها یا Operators در پایتون	عملگرها یا Operators عملوندها رو دستکاری میکنن. ۵+٤ رو در نظر بگیر. ۵ و ٤ عملوند هستن و + عملگر هست. **انواع عملگرها در پایتون:** ۱- عملگرهای ریاضی یا Arithmetic operators: ✓ + (به علاوه)، - (منها)، * (ضـــرب)، / (تقســـیم)، % (باقیمانده)، ** (توان)، // (خارج قسـمت یا تقسـیم کف: یعنی وقتی دو تا عدد تقسـیم شـد عدد کف رو بده مثلا نتیجه تقسـیم بشـه ۸ ممیز ۳، بهت عدد ۸ رو برمیگردونه) ۲- عملگرهای مقایسه‌ای Comparison Operators: ✓ نتیجـه این عملگر بولین (یـه نوع داده تو پـایتون) هسـت یعنی یا True برمیگرده یا False. یادت باشـه که T و F رو با حرف بزرگ بنویسی. ✓ == (مســـاوی)، != (نامســـاوی)، < (کوچکتر از)، > (بزرگتر از)، <= (کوچکتر مســـاوی)، >= (بزرگتر مساوی) ۳- عملگرهای انتصاب یا Assignment operators ✓ = (مساوی)، =+ (یعنی عدد سمت راست رو با عدد ســـمت چپ جمع کن)، =- (یعنی عددی که ســـمت راست می‌آد رو از عدد سمت چپ کم کن) ٤- عملگرهای منطقی یا logical operators ✓ and (اگه همه شـــرایط برقرار باشـــه نتیجه True هست در غیر اینصورت false رو خروجی میده) ✓ or (اگه فقط یکی از شـــرایط برقرار باشـــه True برمیگیرده و در غیر اینصـــورت False رو خروجی میده) ✓ not (مخالف مقایســـه معنیش کن. اگه نتیجه True باشه False رو خروجی میده و برعکس) ❖ یادت باشه که این سه تا رو با حروف کوچیک بنویسی. به جای علامت = هم از علامت == باید اسـتفاده شـه یعنی معادل.	    

عنوان	توضیحات	چطوری تعریف میشه؟
کامنتها و داک استرینگها در پایتون	❖ کامنتها یه جور حاشیه‌نویسی واسه کد هستن که درکش رو آسونتر میکنن ولی روی نحوه اجرای کد تاثیری ندارن. در پایتون اگه اول عبارت یه # بذاری کامنت ساخته میشه و کامپیوتر تمام متن بعدش رو نادیده میگیره. به این علامت octothorpe یا علامت یه عدد یا هشتگ # هم میگن. میتونی تو کدت چندین کامنت داشته باشی. ❖ داک استرینگها شبیه به کامنتها هستن و برای توضیح کد طراحی میشن ولی خیلی خاص‌تر هستن و سینتکس متفاوتی دارن. با گذاشتن چندتا رشته خط حاوی توضیح در مورد تابع، در زیر خط اول تابع نوشته میشن. """ """ Docstringها یه جور سند هستن واسه اینکه بقیه دولوپرها از توابعی که مینویسی استفاده کنن. برخلاف کامنتهای مرسوم، داک استرینگها در طول زمان اجرای برنامه حفظ میشن. این کار به برنامه‌نویس اجازه میده که نظرات رو در زمان اجرا هم بررسی کنه.	

عنوان	توضیحات	چطوری تعریف میشه؟

شرط یا Conditionals در پایتون	قبل از اینکه وارد Control statements یا عبارات کنترلی بشیم باید در مورد تصمیم‌سازی یا decision making بدونیم.
	➢ **تصمیم سازی** یعنی پیش‌بینی شرایطی که در طول اجرای برنامه انجام میشه و تشخیص اقدامات بر اساس این شرایط. در واقع شرایط درست و غلط رو بررسی میکنه که True هست یا False و بر اساس اون تصمیم میگیره.
	➢ **عبارت کنترلی** واسه کنترل شرایط استفاده میشن و شامل:
	✓ **if** یه عبارت کنترلی هست. if از یه عبارت منطقی استفاده میکنه، داده‌ها رو با هم مقایسه میکنه و نتیجه مقایسه رو خروجی میده. اگه شرایط برقرار نباشه یا False باشه، عبارات بعد از شرطی if اجرا میشن.
	✓ **else** تو ترکیب با if می‌اد و به صورت جداگونه استفاده نمیشه. اگه شرایط if برقرار نباشه، شرط else اجرا میشه. میتونی else رو هم نیاری چون آوردنش اجباری نیست.
	✓ **elif** چندین شرط رو بررسی میکنه و هر کدوم True بود همون رو برمیگردونه. در واقع اگه if برقرار نباشه میره سراغ elifها.
		`age = int(input("Please enter yout age: "))`
		`if age < 21:`
		`  print("You can't enter this place!")`
		`Please enter yout age: 22`
		
		`number = int(input("Please enter number: "))`
		`if number < 0:`
		`  print("You number is negative")`
		`Please enter number: 1`
		
		`age = int(input("Please enter yout age: "))`
		`if age < 21:`
		`  print("You can't enter this place!")`
		`else:`
		`  print("Welcome")`
		`Please enter yout age: 22`
		`Welcome`
		
		`number = int(input("Please enter number: "))`
		`if number < 0:`
		`  print("You number is negative")`
		`else:`
		`  print("Positive")`
		`Please enter number: -1`
		`You number is negative`
		
		`Model = int(input("Please choose 1-3: "))`
		`if Model == 1:`
		`  print("You choose Rolls-Royce.")`
		`elif Model == 2:`
		`  print("You choose Aston_Martin.")`
		`elif Model == 3:`
		`  print("You choose Bentley.")`
		`else:`
		`  print("Invalid Number")`
		`Please choose 1-3: 3`
		`You choose Bentley.`

مشق شب ۱:

یه کد بنویس که سن کاربر رو بگیره و به شکل زیر بهش خروجی بده:

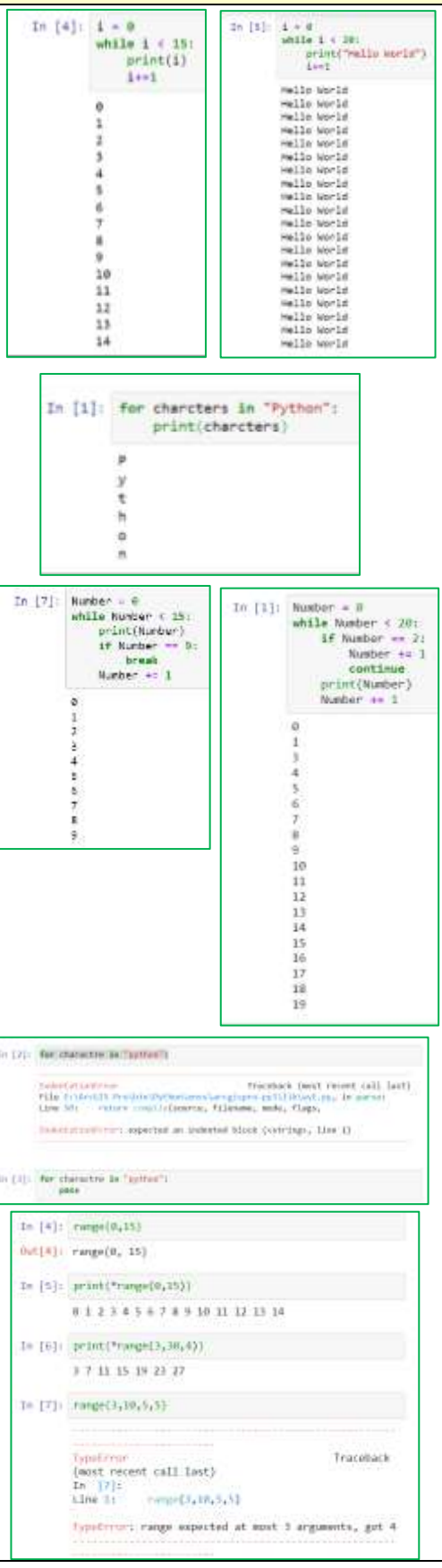
۱- اگه زیر ۱۸ سال بود بهش بگه "تو کودک هستی".

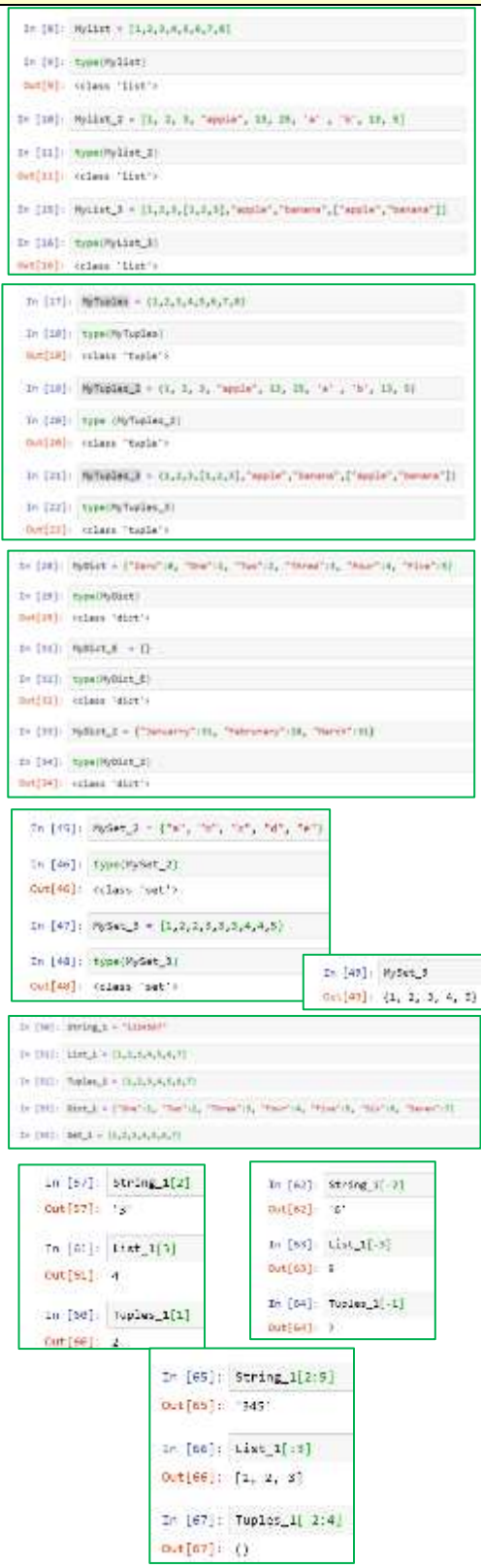
۲- اگه بین ۱۸ تا ۳۵ بود بهش بگه "تو نوجوان هستی".

۳- اگه بین ۳۵ تا ٦٥ بود بهش بگه "تو جوان هستی".

٤- اگه بین ٦٥ تا ۹۰ بود بهش بگه "تو میانسال هستی".

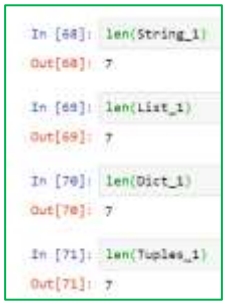
٥- اگه ۹۰ به بالا بود (درغیر اینصورت یعنی اگه جز هیچ کدوم از گزینه‌های بالا نبود) بهش بگه "تو پیر هستی".

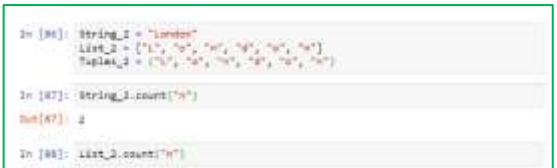
عنوان	توضیحات	چطوری تعریف میشه؟
حلقه‌ها یا loops در پایتون	حلقه‌ها عباراتی هستن که میشه باهاشون کد رو به صورت تکراری اجرا کرد. دو نوع حلقه داریم: ➢ **حلقه while:** تا وقتی که شرط برقرار باشه اجرا میشه. کد مقابل میگه که تا زمانیکه i <۱۵ یا i <۲۰ هست یه فعالیتی رو انجام بده. حلقه while نیاز به شمارنده داره یعنی باید i = ۰ رو که تعریف کردیم بعد انتهای حلقه i += ۱ رو بنویسیم. اگه به i یکی یکی اضافه نکنیم، حلقه تا ابد ادامه پیدا میکنه. ➢ حلقه for: بهت اجازه میده که یک یا چندین فعالیت رو برای هر شرایط اجرا کنی. کد مقابل روی کلمه Python تکرار انجام میده و هر حرفش رو تو هر تکرار خروجی میده و وقتی به اخرین حرف رسید حلقه متوقف میشه. کلا از حلقه‌های for واسه وقتی که از قبل میدونی تعداد تکرار چند تا هست استفاده کن و از حلقه while واسه وقتی که یه شرط هست که باید بهش برسی. حواست به تورفتگیهای کد بلوک حلقه و : باشه. از ساختارهای کنترلی هم میشه واسه حلقه‌ها استفاده کرد که بشه جریان نرمال حلقه‌ها و پیامدشون رو تغییر داد. این ساختارهای کنترلی شامل موارد زیر هستن و با شرط if داخل حلقه می‌آن: ➢ **break:** بیشتر برنامه‌نویسها از break تو حلقه‌ها استفاده میکنن که تو فرایند در حال انجام کد، وقفه ایجاد کنن. هر وقت کد به break رسید اجرای برنامه متوقف میشه. ➢ **continue:** ساختار کنترلی continue واسه شرایطی هست که میخوایم از روی یه چیزی بپریم. تو کد مقابل عدد ۲ خروجی گرفته نمیشه یا از روش میپره. ➢ **pass:** این ساختار نمیخواد هیچ دستور یا کدی رو اجرا کنه. در واقع هیچ اتفاقی نمی‌افته. ازش به عنوان placeholder یا نگهدارنده مکان استفاده میشه. تابع range(): یه تابع از پیش تعریف شده تو پایتون هست و بر اساس مقداری که بهش میدیم اجرا میشه. رنج **دو تا** آرگومان یا مقدار قبول میکنه که ابتدا و انتهای رنج رو بهمون میده. مثلا range(۰،۱۵) از عدد ۰ تا یکی مونده به ۱۵ یعنی ۱۴ رو خروجی میده. آرگومان سوم گام هست یعنی range(۳،۳۰،۴) یه رنج از ۳ تا ۲۹ میده ولی هر عدد با ۴ جمع میشه. اگه ۴ تا آرگومان بهش بدیم خطا میده. علامت * قبل از range هم کمک میکنه که خروجی به صورت افقی باشه نه عمودی. با حقله for هم میتونی بنویسیش: for I in range().اینکار به شکل مرسوم واسه تکرار یه کد به تعداد دفعات مشخص استفاده میشه. با لیست هم میشه نوشتش: x= list(range())	

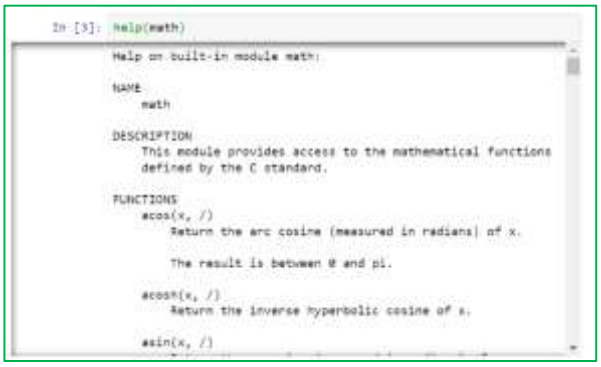
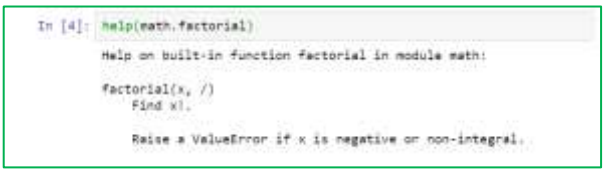
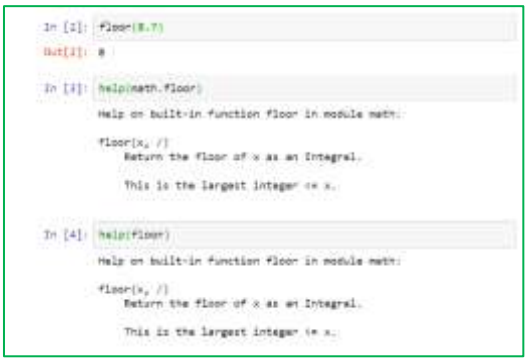
عنوان	توضیحات	چطوری تعریف میشه؟
توالی یا Sequences یا انواع collects در پایتون	پایتون ٦ تا توالی داخلی داره. در واقع انواع داده در پایتون یه توالی هستن. هر مقدار در این نوع داده‌ها ایندکس داره که از ۰ شروع میشه. یعنی ایندکس عدد سوم میشه ۲، ایندکس عدد پنجم میشه ٤ و ... در مورد دو نوع داده یا توالی یعنی Strings و numbers با هم یاد گرفتیم ٤ تا توالی دیگه شامل موارد زیر هستن: ➢ **List:** همه کاره هست. داده‌هاش با کاما از هم جدا مشن و دورشون براکت مربعی میاد [,]. ➢ **Tuples**: تاپلها شبیه لیست هستن ولی immutable یا تغییرناپذیر هستن. ما به این داده‌های تغییر ناپذیر واسه فایلهای پیکربندی یا config نیاز داریم. با کاما جدا میشن و میرن تو پرانتز (,). اگه بین اعدادش کاما نذاریم نوع داده رو integer یا عدد صحیح برمیگردونه. ➢ **Dictionary**: شامل key-value هست. بین key و value دونقطه یا : کولن میاد و دورشون آکولاد {Key: Value}. مقادیر کلیدها میتونن مشابه باشن. کلیدهای مشابه هم میشه داشت. ولی آخرین کلید و مقدار رو در نظر میگیره و خروجی میده. ➢ **Sets**: واسه این خوبه که اعداد تکرای رو از مجموعه برداره و فقط مقادیر اصلی رو خروجی بده. ستها هم آکولاد و ویگول قبول میکنن ولی key-value ندارن. {,} واسه دسترسی به هر کدوم از داده‌های انواع داده باید ایندکسش رو صدا بزنیم. یعنی موقعیتش رو تو نوع داده مشخص کنیم. با [] و گذاشتن شماره ایندکس داخلش اینکار انجام میشه. ✓ اگه از آخر به اول بخوایم بشمریم باید عدد منفی بدیم. ۱- همون عدد آخر هست. ✓ اگه تو [] دو تا عدد بیاریم که با دونقطه : از هم جدا شدن، در واقع میخوایم که مقدار اولین ایندکس تعیین شده رو تا یکی مونده به آخرین ایندکس بهمون بده. ❖ اگه عدد اول حذف شده باشه منظور از ایندکس ۰ هست یا اولین عدد.	

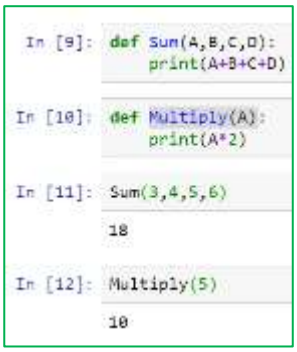
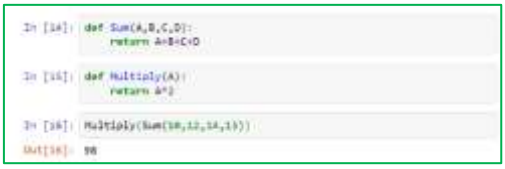
از تابع len() واسه بدست آوردن تعداد اعضای انواع داده استفاده میشه.

- واسه اضافه کردن یه مقدار به مجموعه داده‌ها میشه به شیوه‌های مختلفی اقدام کرد مثل += یا استفاده از متدهای append()، add() و ... یادت باشه که به Tuples نمیتونی مقدار اضافه کنی چون غیرقابل تغییر هست.
- با متد pop() میشه یه عنصر رو از یه نوع داده حذف کرد. یادت باشه که از Tuples نمیتونی چیزی رو حذف کنی.
- با تابع count() میشه تعداد تکرار یه عضو تو مجموعه رو شمرد.
- برای اضافه کردن دو تا مجموعه Set و یا دو تا مجموعه dictionary به هم از update() استفاده کن.
- تابع insert() یه آیتم جدید رو تو موقعیت داده شده به لیست اضافه میکنه. اولین آرگومانش موقعیت ایندکس رو مشخص میکنه. در حالیکه دومین پارامتر آیتمی هست که باید در اون موقعیت جا داده بشه.
- تابع index() اولین آیتمی که با اون گزینه تطبیق داره رو پیدا میکنه و ایندکسش رو برمیگردونه. اگه گزینه مورد نظر تو لیست نباشه خطا میده.
- max(list) بزرگترین عدد لیست رو برمیگردونه.
- min(list) کوچیکترین عدد لیست رو برمیگردونه.
- list.count(item) تعداد اون آیتمی که تو پرانتزش می‌آد رو میشمره که چندبار تو لیست اومده.
- list.remove(item) آیتمی که تو پرانتزش اومده رو از لیست پاک میکنه.
- list.reverse() آیتمهای یه لیست رو معکوس میکنه.





عنوان	توضیحات	چطوری تعریف میشه؟
ماژول یا Modules در پایتون	ماژولها فایلهایی هستن که شامل کدهای پایتون میشن و میتونن توابع، کلاسها و متغیرها رو تعریف کنن. علاوه بر اون میتونن شامل کدهای قابل اجرا باشن. واسه اینکه از کدهای آماده استفاده کنیم ومجدد ننویسیمشون از ماژولها استفاده میکنیم. با عبارت import میشه یه ماژول رو صدا زد. ماژولها فقط یکبار لود میشن. داخل کتابخونه پایتون ready-made modules وجود داره که توسط دولوپرها نوشته شده و میشه از اینترنت و گیت هاب پیداشون کرد. یکی از این ماژولها، ماژول ریاضی یا **math** هست که محاسبات ریاضی رو برامون انجام میده. وقتی واردش کنیم به همه محاسبات ریاضی میتونیم دسترسی داشته باشیم. با تابع dir() میشه فهمید که چه چیزهایی در ماژول math وجود داره. اگه بخوایم ببینم که این ماژول چیکار میکنه از تابع help() استفاده میکنیم. تمام عملگرهای ریاضی و وظایفشون رو برات می اره. اگه فقط یکی از عملگرها رو بخوایم ببینیم باید با نقطه به کلمه math بچسبونیمش. اگه بخوایم یکی از توابع ریاضی رو بیاریم از from math import ... به همراه نام تابع استفاده میکنیم.	   

عنوان	توضیحات	چطوری تعریف میشه؟
توابع یا function	توابع یه ساختار هستن که میتونی به صورت تکراری ازشون استفاده کنیم مثل print(). میشه باهاش کارها رو تو یه مرحله انجام داد. توابع فقط وقتی اجرا میشن که صداشون بزنیم. پایتون تعداد خیلی خیلی زیادی توابع داخلی و توابعی که توسط کاربر تعریف میشه داره. یعنی میتونیم خودمون هم تابع تعریف کنیم. واسه تعریفش از **def** استفاده میکنیم. عبارت return: وقتی بخوایم یه مقدار رو که تو تابع دریافت کردیم، تو یه متغیر نگه داریم یا ذخیره کنیم از return استفاده میکنیم و بعدا از این متغیر میتونیم تو برنامه استفاده کنیم. تو کد مقابل اگه بخوایم با تابع پرینت یه تابع تو در تو بنویسیم خطا دریافت میکنیم. به جای print() باید از return استفاده کنیم. توابع lambda expression: توابع ناشناس یا Anonymous رو میشه با lambda تعریف کرد. باهاش میشه توابع رو در یه خط تعریف کرد. با یه تابع کوتاه تعریف میشه و چندین آرگومان و یه عبارت یا expression داره. معمولا به صورت توابع ناشناس داخل یه تابع دیگه تعریف میشن.	    

عنوان	توضیحات	چطوری تعریف میشه؟
توابع رشته‌ای String Functions	**انواع توابع رشته‌ای:** ➢ رشته‌ها یه تابع format() دارن که این امکان رو فراهم میکنه که با کمک نگهدارنده مکان یـا placeholder مقـادیری در داخلش قرار بگیرن. هر آرگومان تابع format در رشته در موقعیـت مربوط بـه خودش قرار میگیره کـه این کار با کمک آکولادها {} تعیین میشـــه. میتونی نگهدارنده‌های مکان رو به جای اعداد ایندکس نامگذاری کنی. ➢ join() با کمک یه جداکننده، یه لیســـت از رشته رو به یه رشته دیگه وصل میکنه. ➢ split() متضـاد join() هسـت. یه رشـته رو با یه جداکننده مشـــخص داخل یه لیســـت برمیگردونه. ➢ replace() یه زیررشته رو در یه رشته با یکی دیگه جایگزین میکنه. ➢ lower() و upper() رشـــتـه‌هـای بـا حروف کوچیک و بزرگ رو به شـــرایط درخواســـتی تغییر میده.	In [12]: nums = [4, 5, 6] msg = "Numbers: {0} {1} {2}".format(nums[0], nums[1], nums[2]) print(msg) Numbers: 4 5 6 In [13]: a = "{x}, {y}" .format(x=5, y=12) print(a) 5, 12 In [14]: x = ", ".join(["spam", "eggs", "ham"]) print(x) spam, eggs, ham In [15]: str = "some text goes here" x = str.split(' ') print(x) ['some', 'text', 'goes', 'here'] In [16]: x = "Hello ME" print(x.replace("ME", "world")) Hello world In [17]: print("This is a sentence.".upper()) print("AN ALL CAPS SENTENCE".lower()) THIS IS A SENTENCE. an all caps sentence

حل چند تا تمرین

تمرین ۱: دو تا متغیر به اسم a و b بساز و به هر کدوم یه مقدار عددی یا رشته‌ای اختصاص بده و ازشون کنار هم خروجی بگیر.

تمرین ۲: یه لیست بساز که عضوهاش شامل red, blue, green, yellow باشه. بعد یه عضو دیگه به اسم black بهش اضافه کن و از سومین عضو لیست و تعداد اعضای لیست خروجی بگیر

تمرین ۳: از تک تک اعضای لیست خروجی بگیر.

تمرین ٤: یه کد بنویس که موقع خروجی گرفتن از اعضای لیست وقتی به رنگ سبز رسید اجرا کد متوقف شه.

تمرین ٥: برای لیست یه شرط بنویس که هر وقت کد به رنگ زرد رسید پرینت شه 'yellow' و در غیر اینصورت 'not yellow' خروجی گرفته شه.

تمرین ٦: به تابع تعریف کن که تمام اعضای لیست بالا رو چک کنه و اگه قرمز رو پیدا کرد 'red' رو خروجی بده و در غیر اینصورت بنویسه 'not red'.

تمرین ۷: یه کد بنویس که از کاربر رنگ مورد علاقه‌اش رو بگیره و بعد چیزی که کاربر وارد کرده رو خروجی بده.

تمرین ۸: یه کد بنویس که تو اعضای لیست چک کنه و ببینه آیا رنگی که کاربر وارد کرده قرمز هست یا نه اگه نبود 'not red' رو خروجی بده.

جواب تمرین‌ها

جواب تمرین ۱:

```
In [19]: a = "Hello"
         b = "world"
         print (a, b)

         Hello world
```

جواب تمرین ۲: لیست رو با [,] درست میکنیم. برای اضافه کردن عضو بهش از متد append() استفاده میکنیم و برای تعداد اعضا از تابع len().

```
In [24]: list = ['red', 'blue', 'green', 'yellow']
         list.append('black')
         print(list[2])
         print(len(list))

         green
         5
```

جواب تمرین ۳: باید یه حلقه for بنویسیم.

```
In [31]: for colors in list:
             print(colors)

         red
         blue
         green
         yellow
         black
```

جواب تمرین ٤: باید یه حلقه for به همراه یه if و همینطور break بنویســـیم. اگه بخوایم از روی چیزی بپریم باید به جای break از continue اســـتفاده کنیم. تو کد زیر اگه print(colors) رو ببریم زیر break کلمه 'green' تو خروجی نمی آد.

```
In [32]: for colors in list:
             print(colors)
             if colors == "green":
                 break

         red
         blue
         green
```

جواب تمرین ۵: باید یه if و else داخل حلقه for بنویسیم.

```
In [37]: for colors in list:

             if colors == "yellow":
                 print(colors)
             else:
                 print("not yellow")

         not yellow
         not yellow
         not yellow
         yellow
         not yellow
```

جواب تمرین ٦: باید یه تابع با def با یه آرگومان مثلا c تعریف کنیم و تو بلوکش یه شـــرط با if و else بنویســـیم. بعد باید تابع رو صدا بزنیم. واسه اینکه این بررسی روی همه اعضای لیست انجام بشه باید یه حلقه for قبل از صدا زدن تابع بنویسیم. یادت باشه که وقتی میخوای تابع رو صدا بزنی تو دل حلقه for باشه یعنی تورفتگی داشته باشـــه. یادت باشـــه که از عبارت return واســـه نگهداری یا ذخیره یه مقدار تو یه متغیر که از تابع دریافت کردیم اسـتفاده میکنیم که بعدا بتونیم ازش تو برنامه اسـتفاده کنیم. یعنی به جای print() از return اسـتفاده کن و وقتی تابع رو صـــدا زدی بذارش تو تابع print().

```
In [51]: def checkRed(c):
             if c == 'red':
                 return 'red'
             else:
                 return 'not red'
         for colors in list:
             print(checkRed(colors))

         red
         not red
         not red
         not red
         not red
```

جواب تمرین ٧: واســـه گرفتن ورودی از کاربر باید از تابع input() اســـتفاده کنیم و واســـه خروجی گرفتن از تابع print(). یادت باشه که تابع input() رو با متغیر تعریف کنی.

```
In [64]: x = input('What is your favourite color: ')
         print(x)

         What is your favourite color: orange
         orange

In [65]: x

Out[65]: 'orange'
```

جواب تمرین ٨: کافیه تو کد بالا به جای print(x) عبارت print(checkRed(x)) رو وارد کنی.

```
In [66]: x = input('What is your favourite color: ')
         print(checkRed(x))

         What is your favourite color: orange
         not red
```

کتابخونه‌های پایتون

پایتون کتابخونه‌های بی‌نظیری برای data Science علم داده، Machine learning یادگیری ماشین، Artificial Intelligence هوش مصنوعی، Deep leaning یادگیری عمیق و ... داره. چند تا از مهمترین کتابخونه‌های پایتون که در سال ۲۰۲۳ معرفی شده که یادگیریشون ضروری هست رو اینجا آوردیم.

- Numpy
- Pandas
- Matplotlip
- Scikit_Learn
- PyTorch
- Natural Language Toolkit (NLTK)
- Seaborn
- SciPy
- Plotly
- OpenCv
- LightGBM
- Eli۵
- PyBrain
- Squrify
- Permetrics
- …

دسته‌بندی پایین هم به تجربه به دست اومده:

- برای کار با داده‌ها یا data:
  - numpy as np ✓
  - pandas as pd ✓
  - matplotlib.pyplot as plt ✓
- برای زیبایی بیشتر نمودارها
  - seaborn as sns ✓
  - squarify ✓
- برای چک کردن داده‌های موجود در یک جدول
  - missingno as msno ✓
- برای یادگیری ماشین
  - scikit_learn ✓
  - permetrics ✓
- برای کار با داده‌های جغرافیایی
  - geopandas as gpd ✓
  - geoplotlib ✓
  - arcpy ✓
- برای کار با داده‌های رستری
  - rasterio ✓

تو جزوه‌های بعدی چند تا از این کتابخونه‌های مهم رو به همراه کاربردهاشون آموزش میدیم. البته پیش‌نیازش، تسلط به این جزوه هست، پس حسابی این جزوه رو تمرین کن که درک جزوه‌های بعدی برات راحت‌تر شه.