جزوه آموزش پایتون

Python

پودمان برنامه نویسی

کاروفناوری هفتم – هشتم و نهم

توجه : این جزوه به کوشش سایت کاروفناوری کلاله تهیه شده است. لطفا اگر هزینه استفاده از آن را پرداخت نکرده اید. برای پرداخت هزینه به لینک پایین مراجعه فرمایید.

پرداخت هزینه

همراهان محترم یکی از پودمان های کتب کاروفناوری متوسطه اول، پودمان برنامه نویسی می باشد که ممکن است همکاران و دانش آموزان گرامی در تدریس و یاد گیری این پودمان مشکلاتی داشته باشند. لذا ما در سایت کاروفناوری کلاله جزوه آموزشی کامل و جامعی برای این پودمان تهیه کرده ایم که امیدواریم مورد استفاده شما عزیزان قرار بگیرد.

در این جزوه مطالب آموزشی گام به گام پایتون از دانلود و نصب نرم افزار گرفته تا حل کارهای کلاسی و مثال و نکات آموزشی پایتون را گردآوری نموده ایم.

البته ما سعی کردیم مطالب را در حد کتاب و حوصله همکاران و دانش آموزان بنویسیم.

امیدواریم این جزوه برای شما مفید باشد.

ممنون از توجه شما

Suherfe.blog.ir

مقدمه :

مهارت برنامه نویسی، روش های خلاقانه ای را برای حل مسئله به صورت گام به گام به ما می آموزد و باعث پرورش تفکر الگوریتمی، ارتقای مهارت حل مسئله و خلاقیت می شود. تحقیقات نشان می دهد دانش آموزانی که مهارت برنامه نویسی را در سنین کمتر می آموزند، به طور قابل توجهی قدرت حل مسئله در آن ها پرورش می یابد. همچنین این دانش آموزان در حوزه های دیگر مانند؛ تغییر نگرش نسبت به خطا، اعتماد به نفس، احساس توانمندی در یادگیری و استدلال منطقی نیز نسبت به دیگران تفاوت چشم گیری خواهند داشت. تأثیر مثبت این آموزش ها بر قدرت درک مفاهیم ریاضی و ارتقای مهارت های اجتماعی نیز ثابت شده است.

امروزه رایانه ها همه جا حضور دارند و در همه امور مربوط به انسان ها می توان ردی از رایانه ها را دید. در صنعت، پزشکی، آموزش، بازی های رایانه ای حمل ونقل،خانه های هوشمند ، ادارات و غیره. سؤال مهم این است که چرا رایانه ها این گونه در زندگی انسان عصر مدرن نفوذ پیدا کرده اند؟ به عبارت دیگر، می خواهیم به این مسئله بپردازیم که اصلاً رایانه ها چه می کنند؟ رایانه ها را می توان دستیاری قدرتمند برای انسان ها دانست. رایانه ها آمده اند تا وظایف و کارهای تکراری ای از انسان ها را به عهده بگیرند که انجام آن ها دشوار و گاهی امکان پذیر نیست. البته رایانه ها با وجود پیشرفت های شگرف در حوزه هوش مصنوعی، در برخی زمینه ها مانند هوش و قدرت تشخیص، هنوز به پای انسان نرسیده اند. با این حال سرعت پردازش آن ها به ویژه در برخی موارد مانند محاسبات تکراری و دنباله دار، بسیار بیشتر از سرعت پردازش انسان است. به این موارد می توان قدرت ذخیره سازی بیشتر و دقت بالا در محاسبات را هم افزود. در نهایت همه این موارد باعث شده که رایانه ها نقشی حیاتی در زندگی انسان امروز ایفا کنند.

سؤال مهم این است که انسان ها چگونه می توانند با رایانه ها حرف بزنند یا زبان آن ها را بفهمند؟ زبان رایانه ها که به آن زبان ماشین هم گفته می شود، زبان صفر و یک ها است و تعامل با این زبان برای انسان کار ساده ای نیست.

همان طور که انسان ها برای انتقال پیام از ابزاری به نام زبان استفاده می کنند و پیام خود را در قالب کلماتی که با ساختاری قانونمند در کنار هم چیده شده اند، به دیگری منتقل می کنند، برای تعامل با رایانه ها نیز از ابزار زبان استفاده می کنند. این نوع خاص از زبان را زبان برنامه نویسی می گویند.

زبان برنامه نویسی پایتون یکی از زبان های برنامه نویسی است که فراگیری آن ساده و در عین سادگی، کارایی فوق العاده ای دارد. پایتون در مقایسه با دیگر زبان های برنامه نویسی، گرامر بسیار ساده ای برای حروف چینی و اجرای برنامه ها دارد.کدهای نوشته شده به این زبان خوانایی بالایی دارند و برنامه های آن را می توان با تعداد خط کد کمتری نسبت به زبان های دیگر نوشت

همچنین این زبان جزء زبان هایی با منبع کد آزاد است و استفاده از آن کاملاً رایگان می باشد. پایتون به همراه کتاب خانه های مختلف به صورت رایگان در اختیار کاربران قرار می گیرد و قدرت فوق العاده ای به کاربران آن می بخشد. به همین دلیل این زبان طرفداران زیادی دارد.

زبان برنامه نویسی پایتون یکی از محبوب ترین زبان های برنامه نویسی دنیاست که هم در بین مبتدیان و هم در میان حرفه ای ها طرف دار دارد. شرکت های بزرگ و بسیاری از دانشگاه ها و مراکز علمی برای پروژه های تحقیقاتی خود از این زبان استفاده می کنند و این نشان از موفقیت و قدرت آن است.

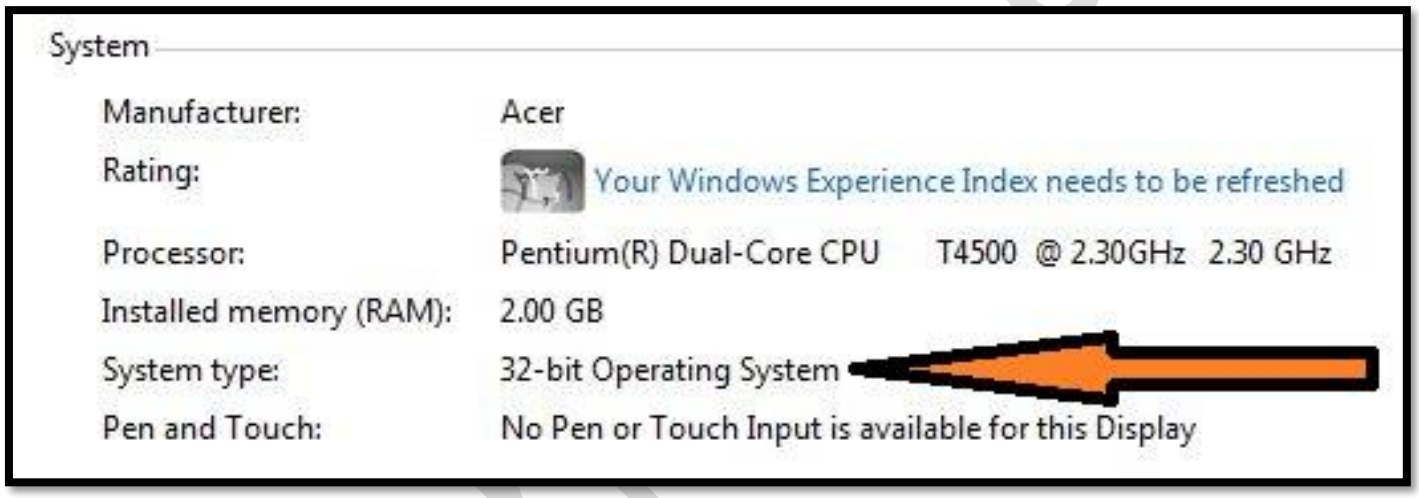
نصب و اجرای پایتون

ابتدا ببینید ویندوز رایانه شما چند بیتی است ۳۲ یا ۶۴

برای اینکار روی Computer در دسکتاپ ویندوز ۷ یا روی This Pc روی دسکتاپ ویندوز ۱۰ راست کلیک کرده و گزینه Properties را انتخاب کنید.

در پنجره باز شده می توانید نوع ویندوز را ببینید.

در تصویر زیر ویندوز ما ۳۲ بیتی است.



حال که معماری ویندوز خود را شناختید بر حسب این نوع ویندوز نسخه برنامه پایتون را روی رایانه خود نصب کنید. برای این کار می توانید از لینک زیر بر اساس نوع ویندوز، برنامه Python را دانلود و روی رایانه خود نصب کنید.

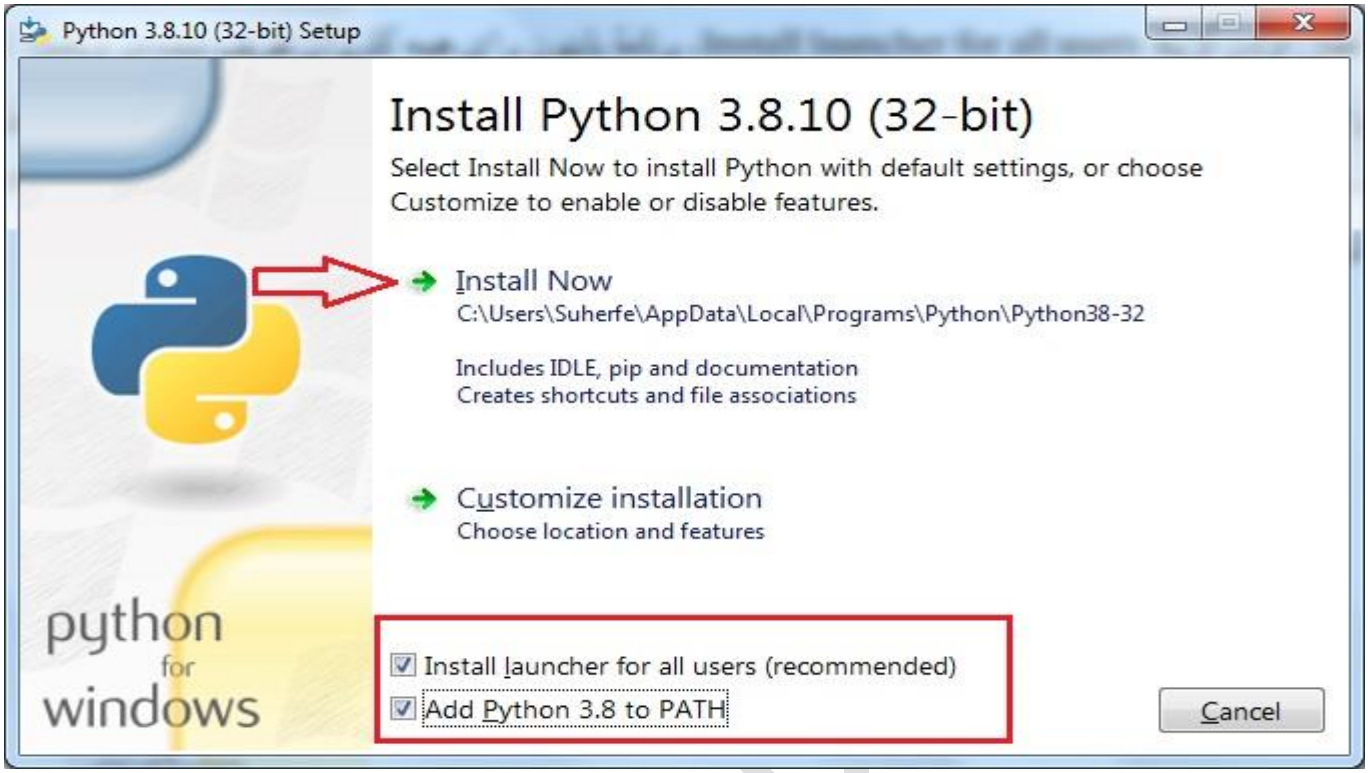
دانلود پایتون Python

نصب Python

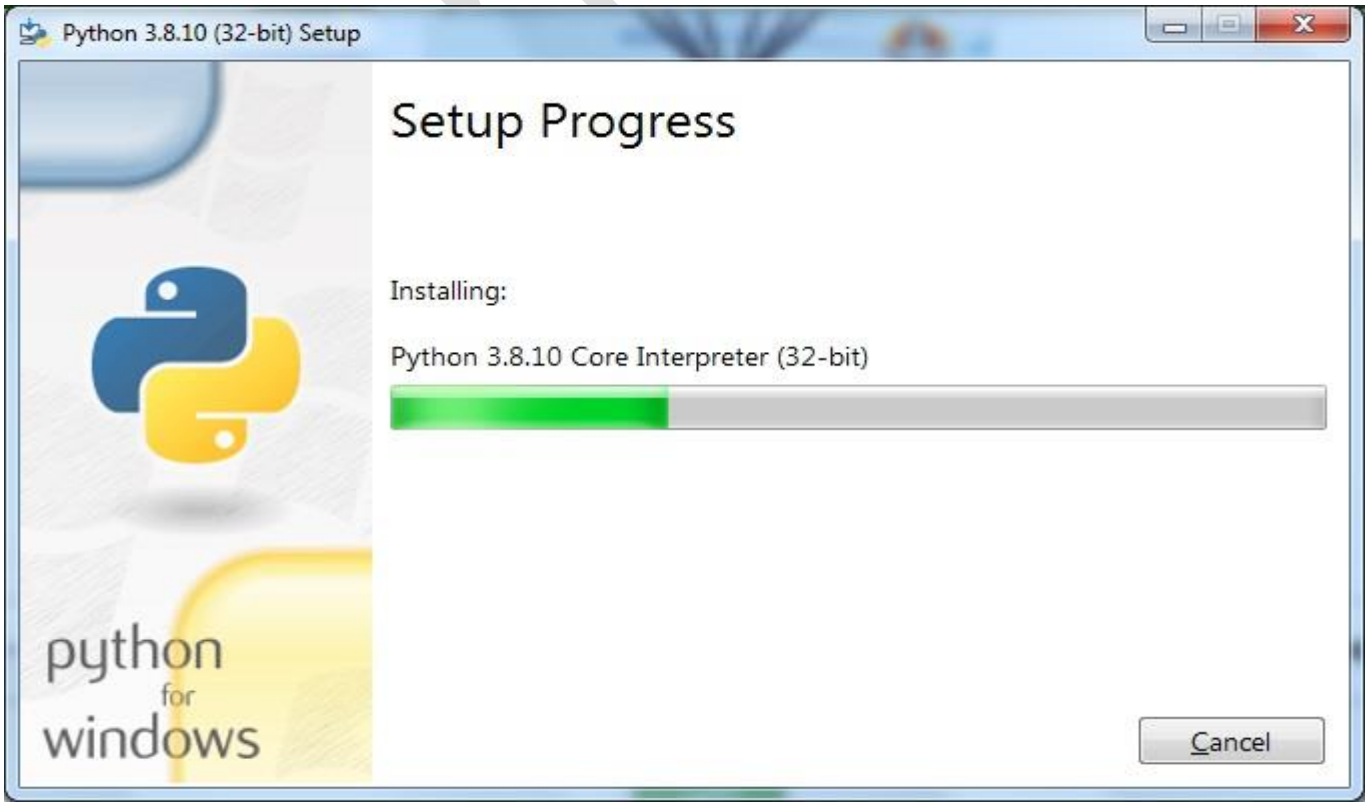
پس از دانلود، فایل را از حالت فشرده خارج کنید.

سپس روی فایل راست کلیک و گزینه Run as Administrator را انتخاب کنید. بعد پیغام ظاهر شده را با انتخاب Yes یا Run تأیید و آن را نصب کنید.

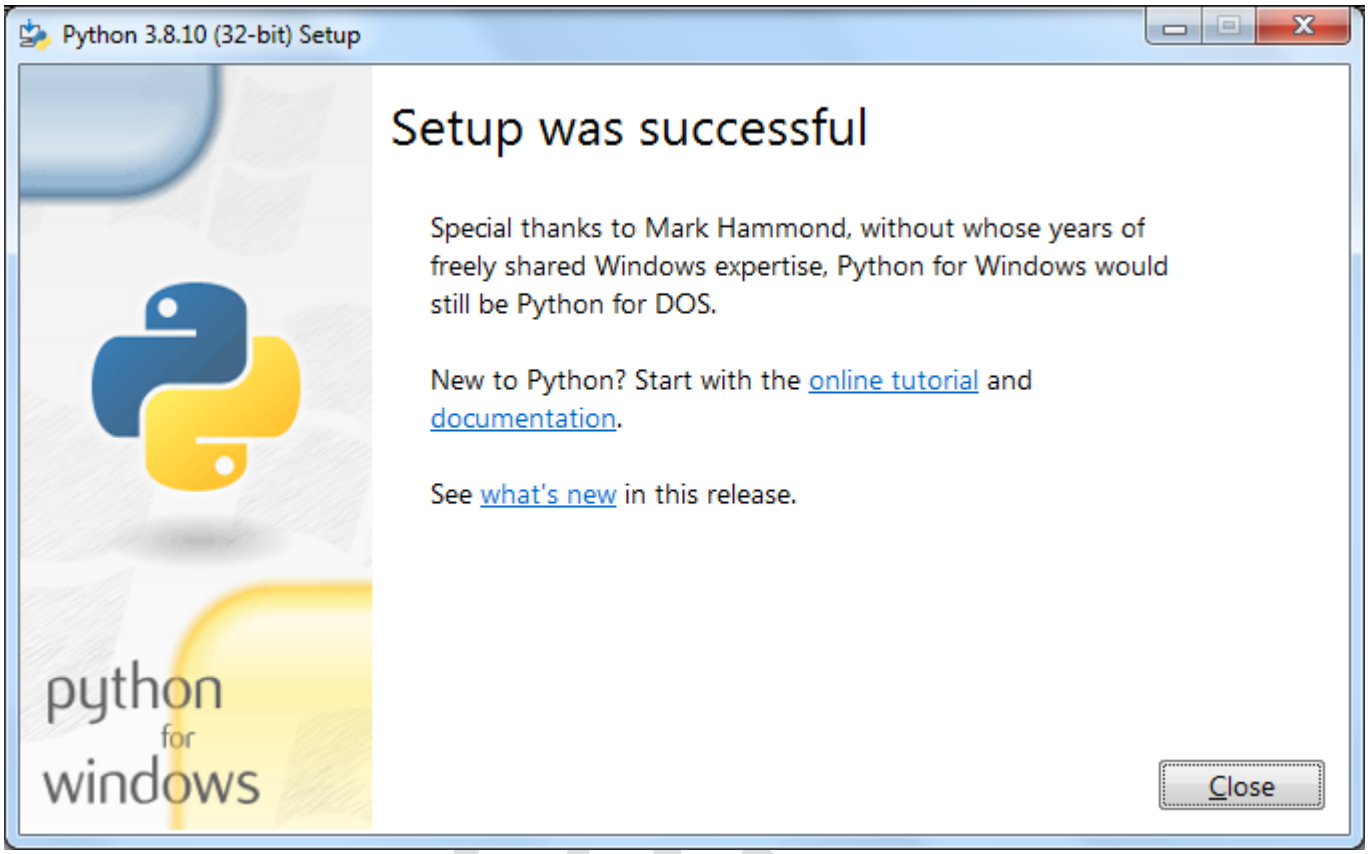
در ادامه نصب پنجره پایین باز می شود.



در کادر باز شده ابتدا گزینه های مشخص شده در مستطیل قرمز را تیک بزنید و سپس روی گزینه Install Now کلیک کنید تا برنامه در رایانه نصب شود.

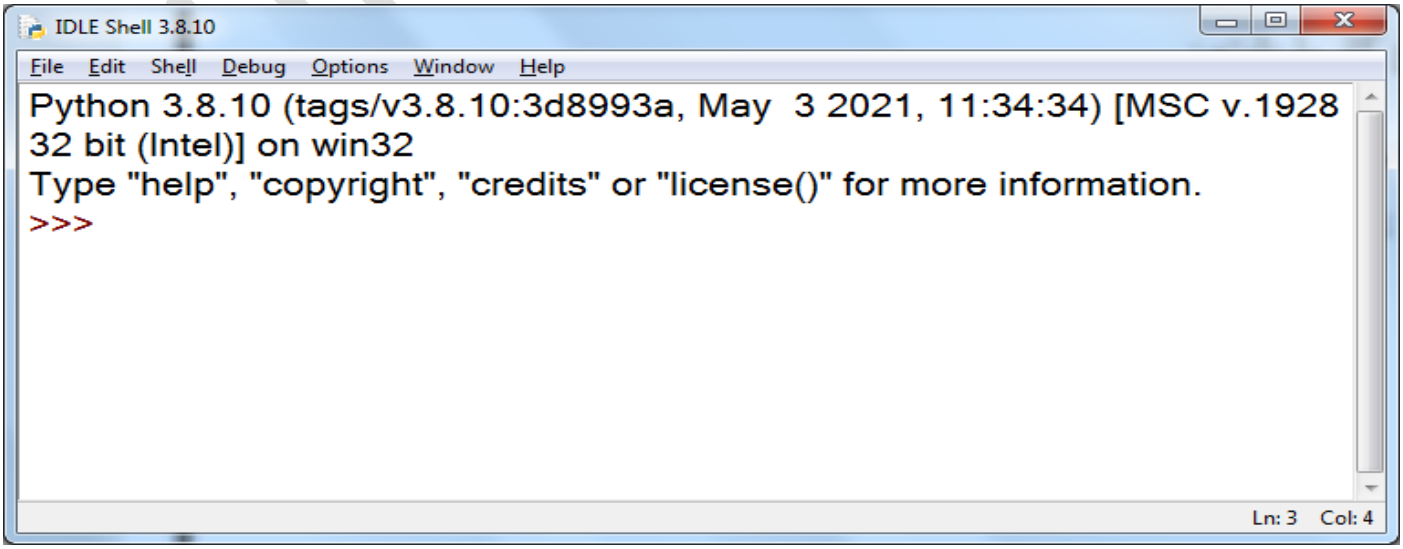


با ظاهر شدن پنجره پایین پایتون با موفقیت در رایانه شما نصب می شود.



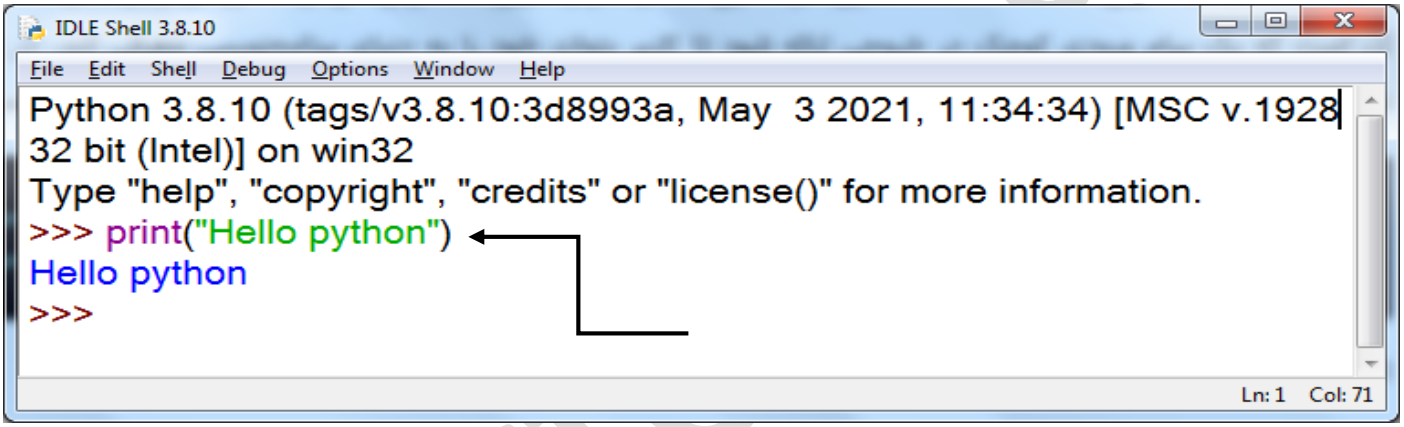
حال روی close کلیک کنید.

برای اجرای برنامه از منوی Start و پوشه Python گزینه IDLE را انتخاب کنید تا برنامه باز شود. این پنجره شل پایتون نام دارد که در آن می توانید کدنویسی کنید.

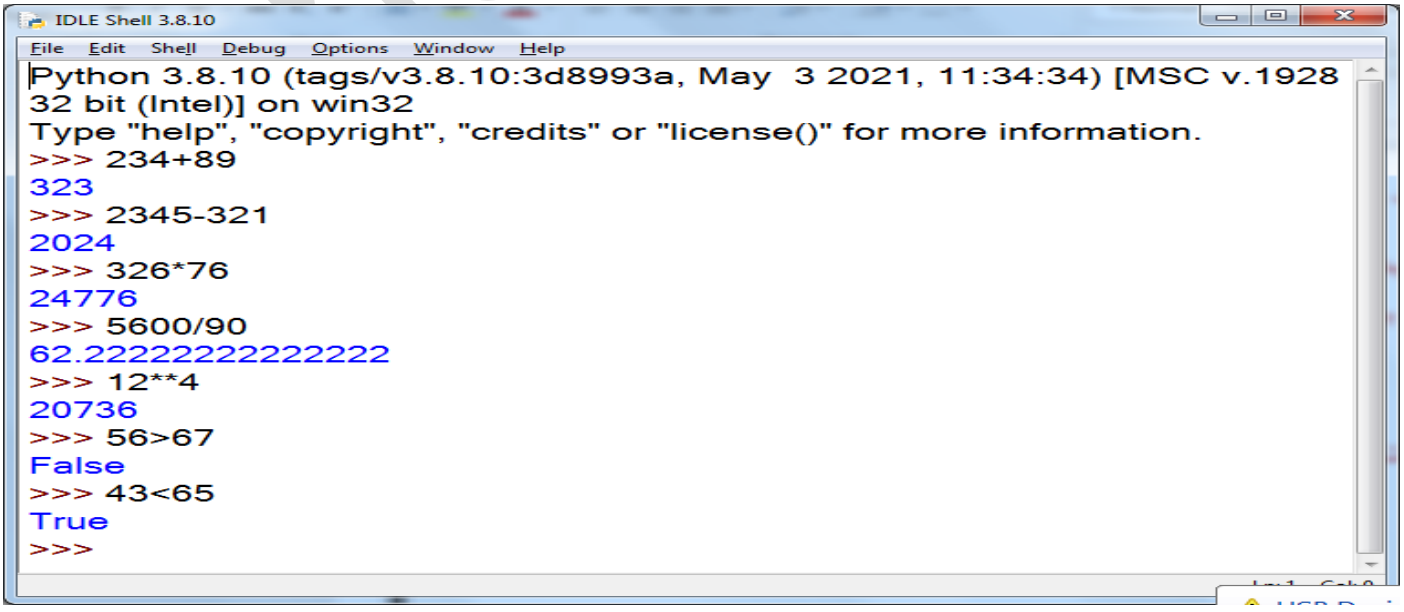


شروع نوشتن در پایتون

۱- دستور پرینت - یکی از ساده ترین فرمان ها در پایتون دستور Print() است. به عنوان مثال در کادر بنویسید print("Hello Python") و سپس اینتر بزنید خواهید دید که در خط بعدی کلمه Hello Python تایپ می شود. به یاد داشته باشید که این کلمه باید در پرانتز و بین دو کوتیشن ("") قرار داشته باشد. پس شما می توانید هر متن یا کلمه دیگری بین دو کوتیشن را قرار دهید.



۲- انجام محاسبات ریاضی - در پایتون می توانید محاسبات ریاضی و منطقی را نوشته و نتیجه را با فشردن کلید اینتر در خط بعدی پنجره شل پایتون ببینید.



گاهی پس از اجرای برنامه ممکن است خطاهایی ایجاد شود و برنامه اجرا نشود که باید بررسی کرده و متن و نوع فرمان را اصلاح کنید. و آنقدر اجرا و تغییر دهید تا برنامه اجرا شود.

یکی از متداول ترین خطا ها خطاهای نحوی Syntax Error مثل اشتباه نوشتن کدها یا حروف چینی است که برای رفع آن باید نحوه نوشتن کد ها را بررسی و اصلاح کنید.

متغیر:

متغیر Variable نامی است که به یک مکان خاص در حافظه کامپیوتر برای ذخیره مقادیر داده در یک برنامه کامپیوتری نسبت داده می‌شود. متغیر در برنامه نویسی یک قطعه نام‌گذاری شده در حافظه کامپیوتر است که داده‌ها در آن قابل ذخیره‌سازی هستند. متغیرها در برنامه نویسی را می‌توان مانند جعبه‌ای با یک نام در نظر گرفت که در داخل آن اطلاعاتی ذخیره می‌شوند. متغیر یک فضای رزرو شده در حافظه است که مقادیر انتساب داده شده به خود را درون آن بخش از حافظه نگهداری می‌کند. در زمان ایجاد متغیر، بخشی از حافظه به آن تخصیص داده شده و اشغال می‌شود. در کد زیر چند متغیر تعریف کرده و مقادیر دلخواهی به آن‌ها انتساب داده‌ایم.

```
name = "Ali"
age = 21
score = 236.7
```

در این کد، سه متغیر name و age و score داریم که به ترتیب دارای مقادیر Ali و ۲۱ و ۲۳۶,۷ هستند.

قواعد نام‌گذاری متغیرها در پایتون

نام متغیرها در پایتون از چند قاعده پیروی می‌کند که باید رعایت شوند. اگر این قوانین را رعایت نکنیم، با خطا مواجه خواهیم شد.

این قوانین عبارتند از :

۱- حروف مجاز برای نام متغیر در پایتون حروف الفبا (کوچک و بزرگ) انگلیسی و علامت زیر خط (_) است

۲- نام متغیر باید با یک حرف انگلیسی آغاز شود.

۳- ابتدای نام متغیر نمی‌تواند عدد باشد.

۴- نام متغیر در پایتون حساس به حروف کوچک و بزرگ است بنابراین متغیرهای name و Name و NAME سه متغیر متفاوت هستند.

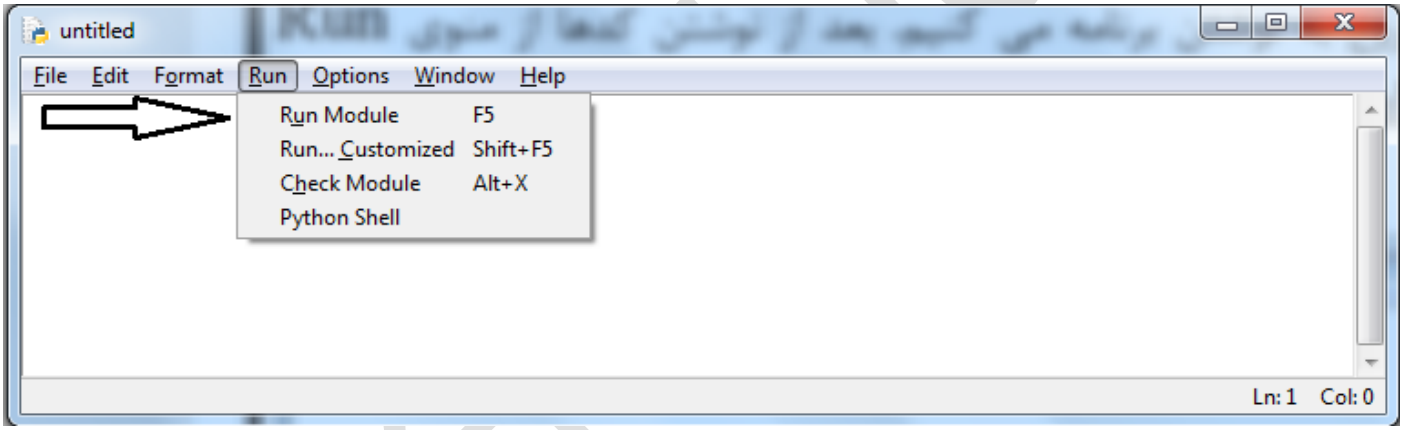
۵- استفاده از کاراکتر فاصله در نام متغیر غیرمجاز است.

۶- استفاده از کلمات کلیدی پایتون در نام متغیر غیرمجاز است. (تصویر پایین)

Keywords in Python				
False	class	finally	is	return
None	continue	for	lambda	try
True	def	from	nonlocal	while
and	del	global	not	with
as	elif	if	or	yield
assert	else	import	pass	
break	except	in	raise	

در ادامه بحث با مطرح کردن مثال های مرتبط و متعدد و درج پاسخ آن ها در خدمت شما هستیم.

برای کد نویسی و شروع کار از منوی File در شل پایتون گزینه New File را انتخاب کنید تا پنجره جدیدی برای نوشتن برنامه ها باز شود حال در این پنجره شروع به نوشتن برنامه می کنیم. بعد از نوشتن کدها از منوی Run گزینه Run Module را انتخاب کنید تا خروجی برنامه را ببینید.

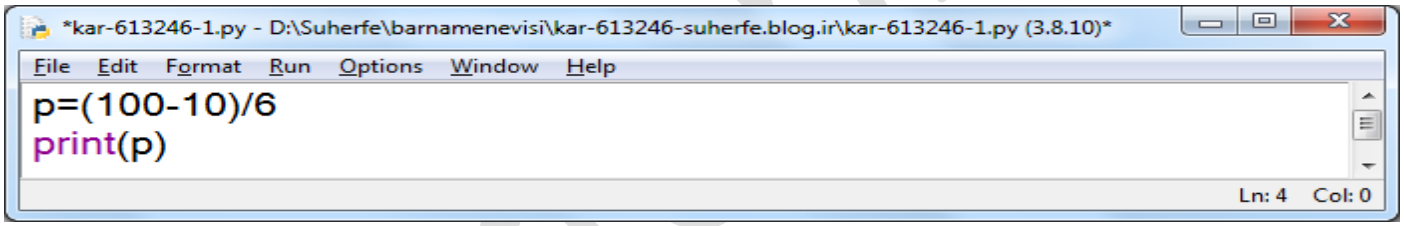


مثال ۱

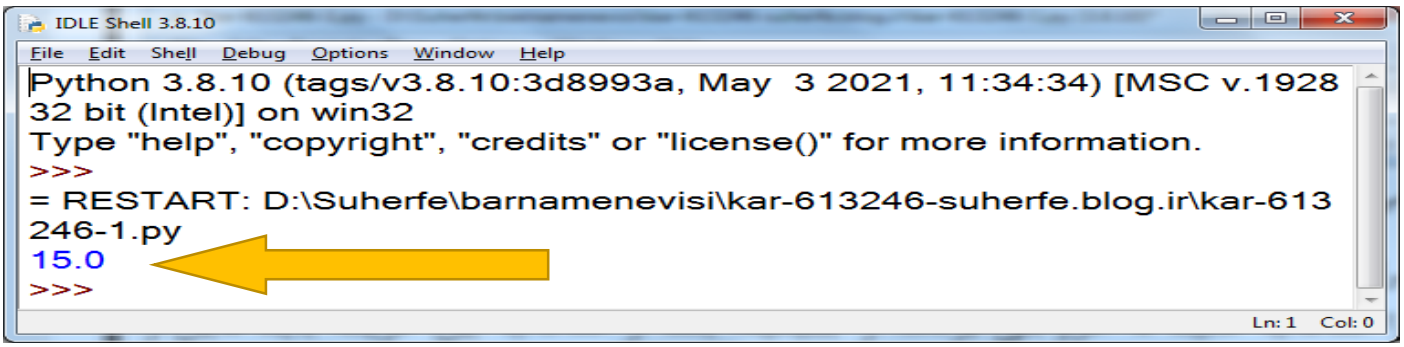
فاطمه کتاب داستانی را در ۶ ساعت مطالعه کرد و ۱۰ صفحه از آن باقی ماند. اگر این کتاب ۱۰۰ صفحه داشته باشد، فاطمه به طور متوسط در هر ساعت چند صفحه از آن را مطالعه کرده است؟

پاسخ :

ابتدا متغیری به نام p تعریف می کنیم و فرمول جواب را به آن نسبت می دهیم یعنی فاطمه در ۶ ساعت ۹۰ صفحه مطالعه کرده پس ۹۰ را بر ۶ تقسیم می کنیم تا به جواب برسیم و سپس پرینت متغیر P



برای دیدن خروجی بعد از نوشتن کدها از منوی Run گزینه Run Module را انتخاب کنید تا خروجی برنامه را ببینید. بعد از انتخاب این گزینه باید فایل را در رایانه save کنید که با انتخاب یک محل برنامه را ذخیره کنید بعد از ذخیره جواب را خواهید دید که برابر است با ۱۵ یعنی فاطمه در هر ساعت میانگین ۱۵ صفحه مطالعه کرده است.

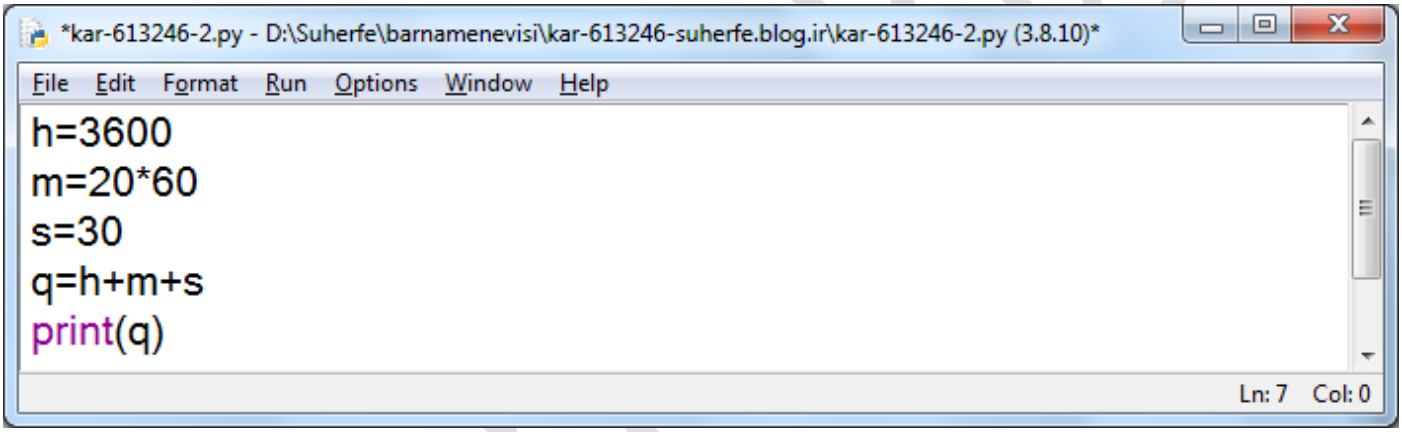


مثال ۲

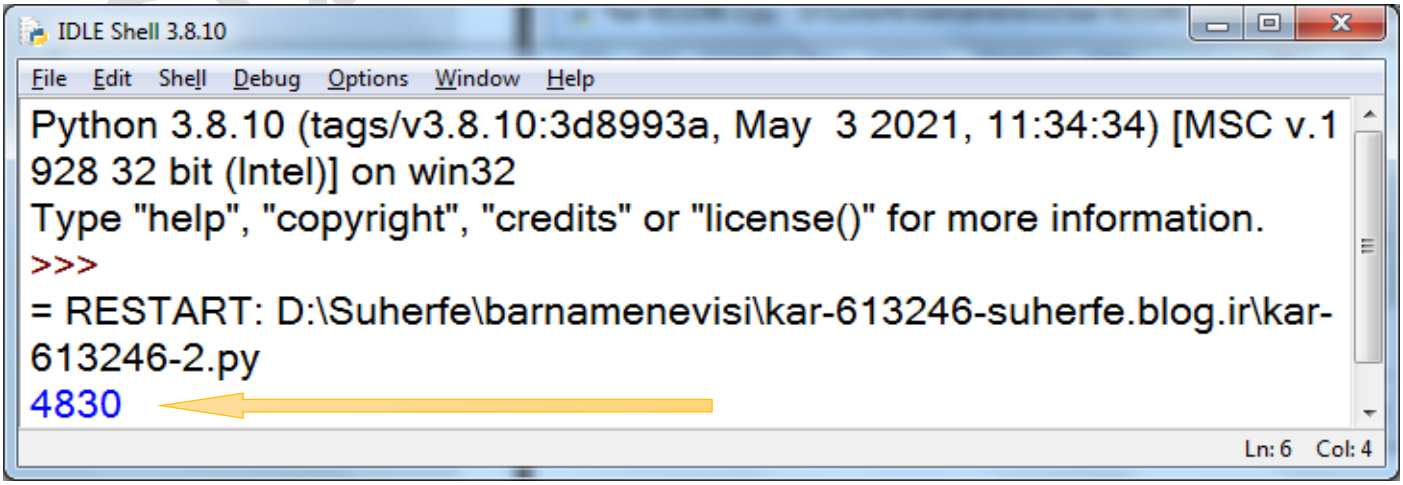
محاسبه کنید یک ساعت و بیست دقیقه و سی ثانیه، چند ثانیه است.

پاسخ :

برای حل این سوال باید ثانیه های ۱ ساعت و ۲۰ دقیقه و سی ثانیه را با هم جمع و پرینت بگیریم. در جواب پایین سه متغیر hوmوs را با هم جمع کرده تا q بدست بیاید.



حال q را چاپ گرفته تا خروجی ۴۸۳۰ ثانیه نمایش داده شود. برای دیدن خروجی بعد از نوشتن کدها از منوی Run گزینه Run Module را انتخاب کنید تا خروجی برنامه را ببینید.

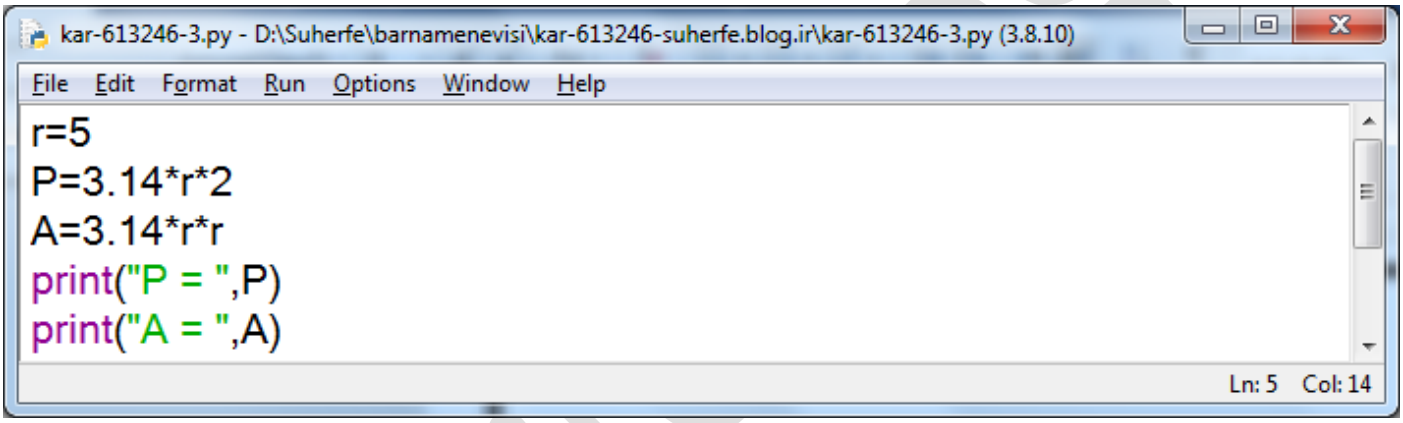


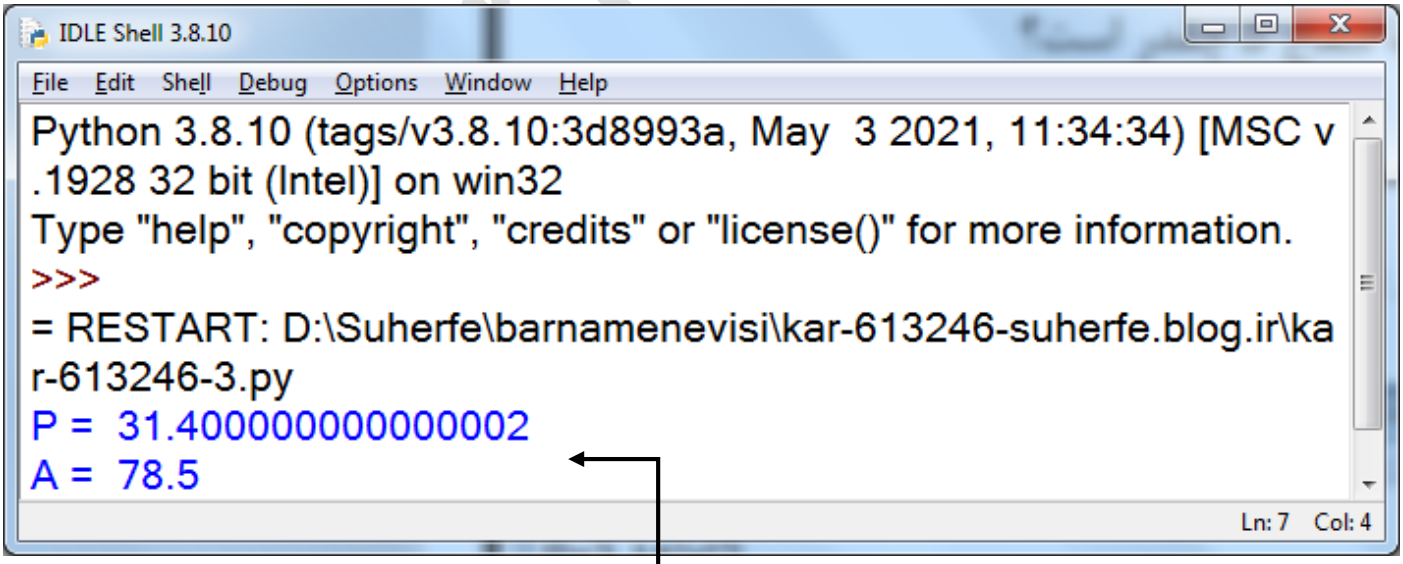
مثال ۳

محیط و مساحت یک دایره با شعاع ۵ چقدر است؟

پاسخ :

ابتدا شعاع را به متغیر r و بعد فرمول محیط و مساحت را نوشته و به دو متغیر P و A نسبت می دهیم و در پایان پرینت P و A





محیط برابر است با ۳۱/۴۰

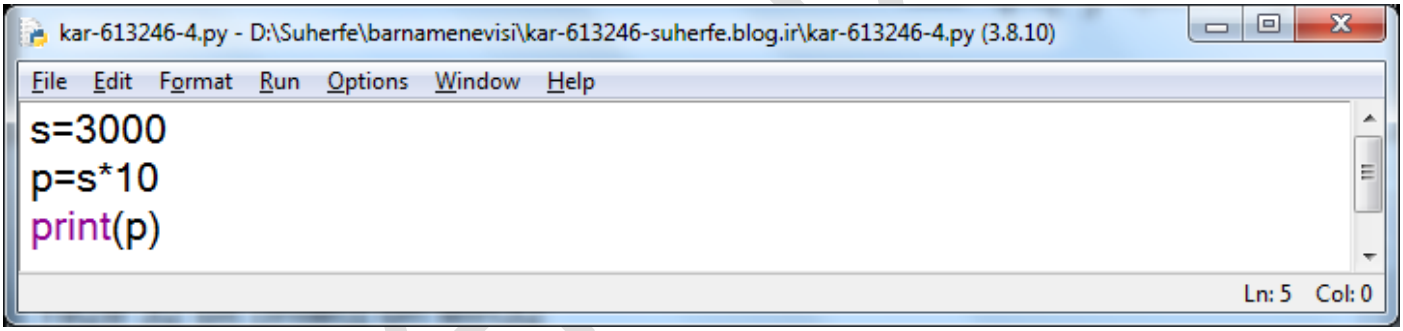
مساحت برابر است با ۷۸/۵

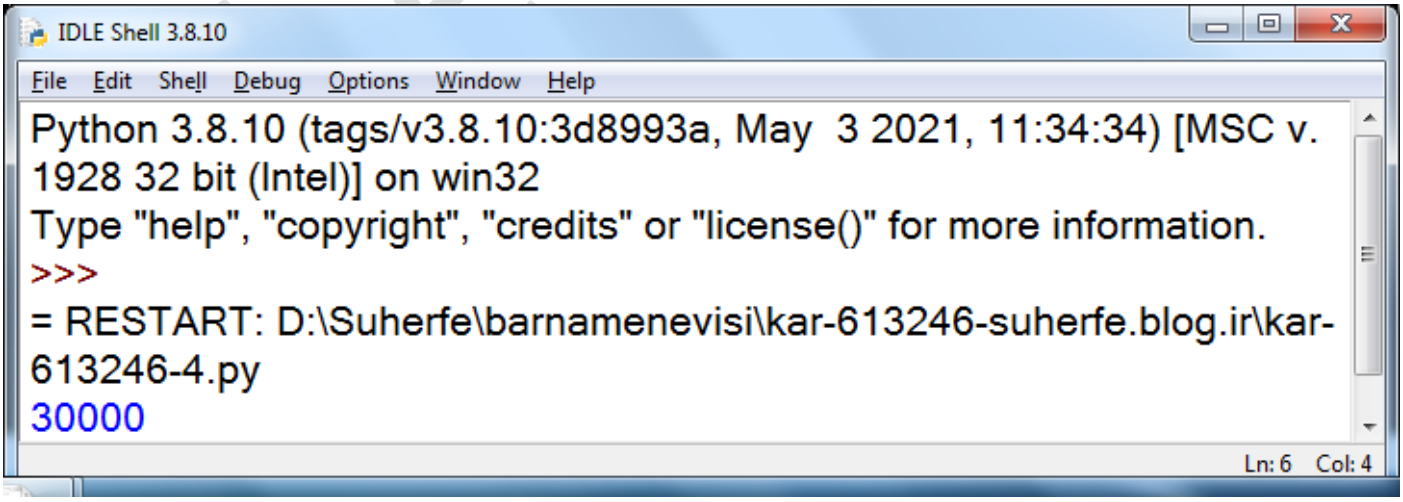
مثال ۴

پس انداز هفتگی محمد، ۳۰۰۰ تومان است. او حساب کرد ۵ هفته پس انداز او، نصف قیمت کیفی است که دوست دارد آن را بخرد. برنامه ای بنویسید که قیمت کیف را محاسبه و چاپ کند.

پاسخ :

از متن سوال نتیجه می گیریم که محمد با ۱۰ هفته پس انداز می تواند کیف بخرد.پس متغیر S که پس انداز هفتگی محمد است را ۱۰ برابر کرده و آن را به متغیر p نسبت می دهیم و نهایتا چاپ p.





جواب ۳۰۰۰۰ تومان

متن یا رشته

نوعی داده متنی است که در برنامه نویسی به آن رشته هم گفته می شود، و در کنار نوع عددی از رایج ترین انواع داده مورد استفاده در پایتون است.

این نوع داده دنباله ای از حروف، اعداد یا کاراکترها را شامل می شود که در میان دو یا چهار کوتیشن "" قرار می گیرند. مانند متن زیر

Text = "Hello Python"

تاکنون، برنامه‌هایی که نوشته شد ایستا بودند. یعنی مقدار متغیرها ثابت بودند برای انعطاف‌پذیری بیشتر ، ممکن است نیاز به گرفتن ورودی از کاربر باشد. در پایتون، تابع input() برای انجام این کار وجود دارد. برای دریافت ورودی می توان ورودی را در یک متغیر ذخیره کرد. این تابع به صورت زیر نوشته می شود.

A=input("مقدار")

در این تابع ورودی در متغیر A ذخیره می شود.

مقادیری که در ورودی توسط تابع input() دریافت می شود، به صورت متن یا رشته داخل متغیر قرار می گیرد. برای تبدیل این رشته به یک عدد صحیح یا برای دریافت عدد صحیح از ورودی می‌توان از تابع int() و برای دریافت اعداد اعشاری از تابع float() استفاده کرد.

عملگرهای ریاضی در پایتون

یکی از کاربردهای اولیه رایانه، انجام عملیات ریاضی و مقایسه ای است.

برای انجام محاسبات ریاضی در پایتون از عملگرهای زیر استفاده می کنیم.

نمونـــــه	عملگر	شرح	مثال
۵+۷=۱۲	+	جمع	a+b
۱۸−۷=۱۱	−	تفریق	a−b
۹*۸=۵۴	*	ضرب	a*b
۳۶/۳=۱۲	/	تقسیم	a/b
۲۱//۴=۵	//	تقسیم صحیح	a//b
۱۸٪۴=۲	%	باقیمانده تقسیم	a%b
۳**۴=۸۱	**	توان	a**b

تقدم عملگرهای ریاضی در برنامه نویسی

توجه داشته باشید که پرانتز می تواند اولویت عملگرها را بالا ببرد یعنی هر عملگری که داخل پرانتز باشد اولویت بالاتری دارد.

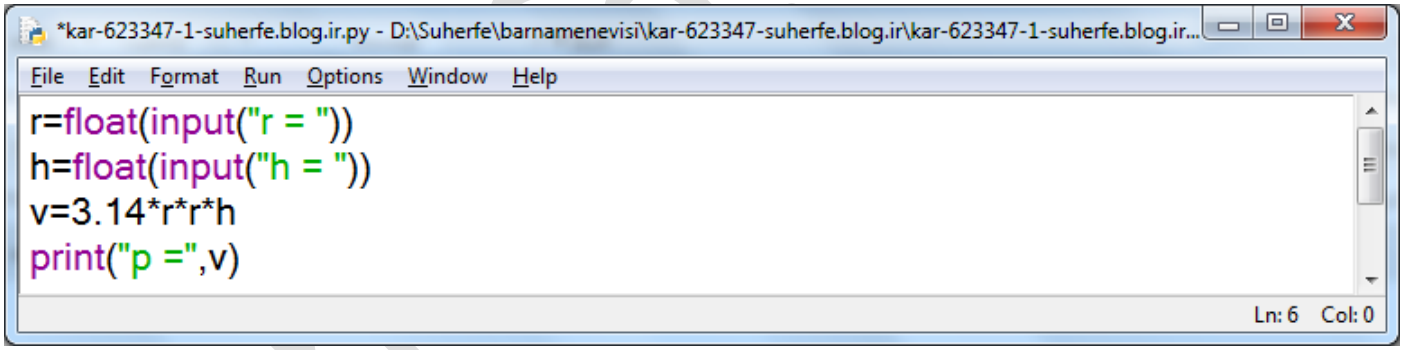
اولویت	عملگر
۱	**
۲	*, /, //, %
۳	+, −

مثال ۵

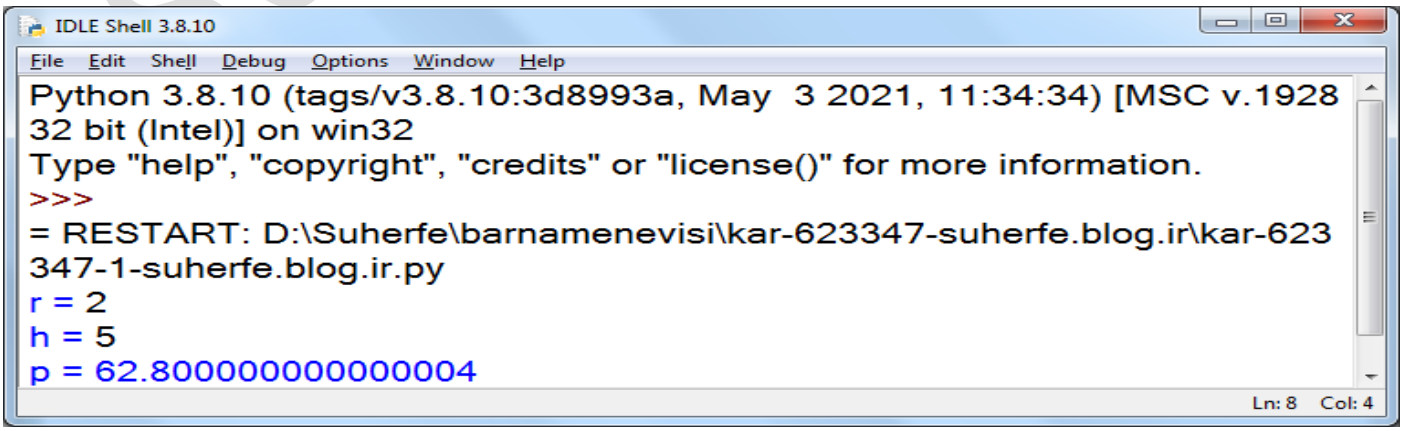
برنامه ای بنویسید که شعاع قاعده و ارتفاع منبع آب استوانه ای را از ورودی دریافت و سپس محاسبه کند که چند مترمکعب آب می گیرد.

پاسخ :

ابتدا دو متغیر r و h برای وارد کردن شعاع r و h ارتفاع تعریف می کنیم و تابع ورودی آن را می نویسیم. چون ممکن است عدد دریافت شده اعشاری باشد از تابع float استفاده می کنیم. فرمول حجم را نوشته و به متغیر v نسبت می دهیم اگر برنامه را اجرا کنیم ابتدا از ما دو عدد شعاع و ارتفاع درخواست می شود و نهایتا با اینتر زدن حجم استوانه محاسبه و در تابع print نمایش داده می شود.



اگر شعاع را ۲و ارتفاع را ۵ متر در نظر بگیریم و در برنامه وارد کنیم حجم برابر ۶۲,۸ متر مکعب نمایش داده خواهد شد.



مثال ۶

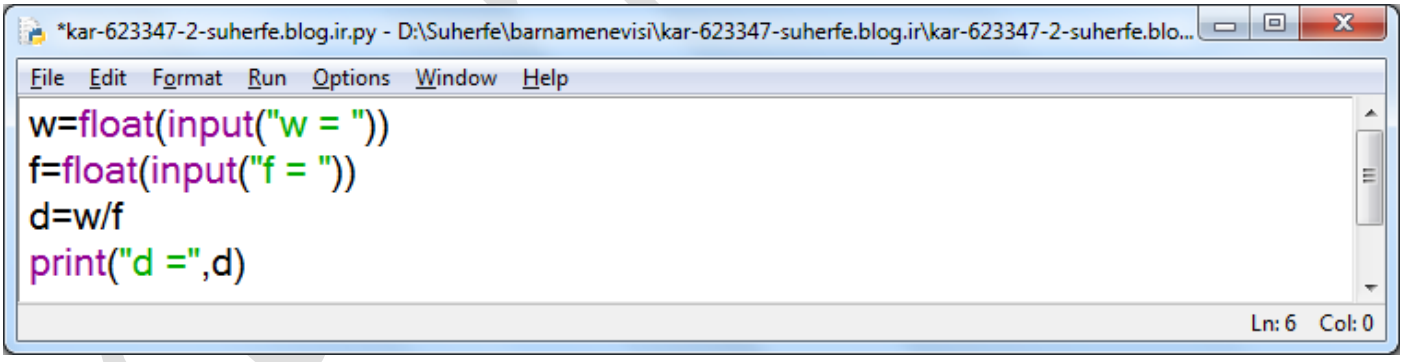
در درس علوم آموختید که کار انجام شده با مقدار نیرو در اندازه جابه جایی برابر است. این رابطه را با تساوی W=F.D نشان می دهیم. برنامه ای بنویسید که کار انجام شده و مقدار نیرو را از ورودی دریافت و سپس میزان جابه جایی را محاسبه و چاپ کند.

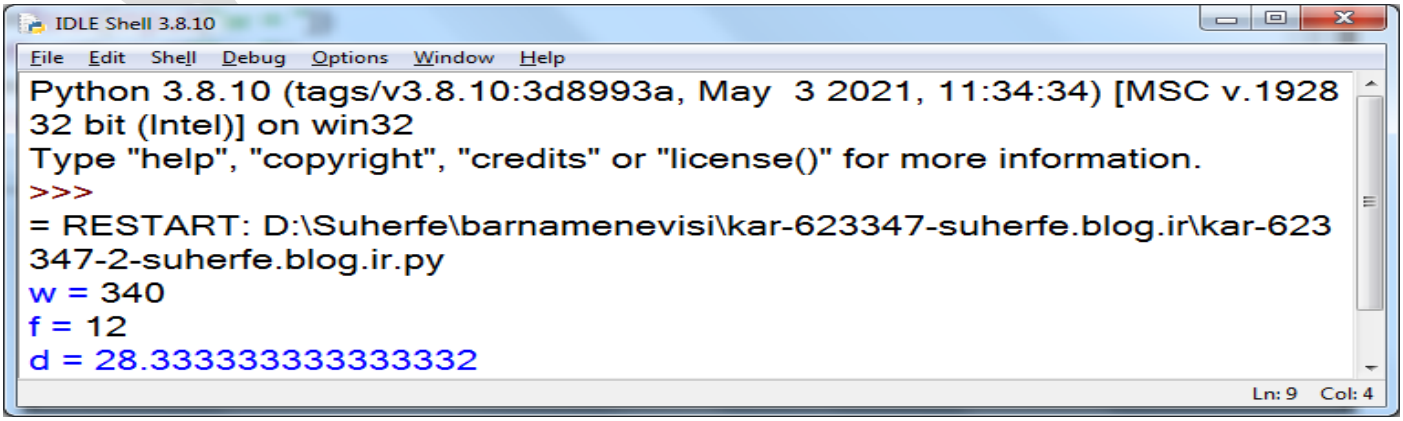
پاسخ :

میزان جابجایی از فرمول پایین محاسبه می شود.

$$d=w/f$$

پس مقدار کار و نیرو از کاربر دریافت می شود مثلا ۳۴۰ و ۱۲ و طبق فرمول بالا مقدار جابجایی ۲۸,۳ محاسبه و بوسیله تابع Print نمایش داده می شود.



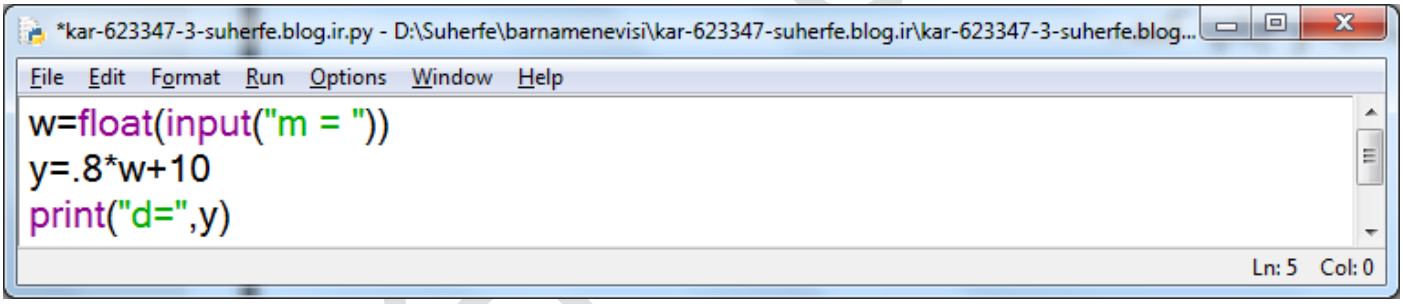


مثال ۷

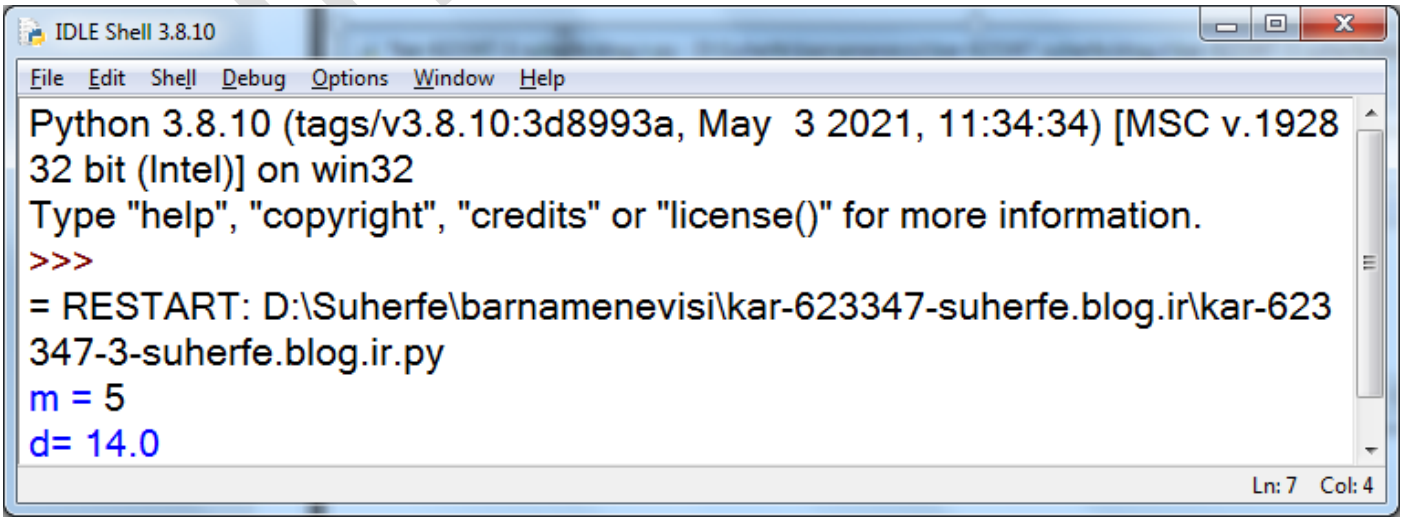
طول یک فنر 10 سانتی متر است. وقتی وزنه ای به جرم X به آن وصل شود، طول فنر از رابطه $Y=0/8\ x+10$ محاسبه می شود. برنامه ای بنویسید که از ورودی جرم وزنه ای برحسب کیلوگرم که به آن وصل شده دریافت و سپس، طول فنر را محاسبه و چاپ کند.

پاسخ :

ابتدا تابع ورودی مقدار جرم وزنه را می نویسیم و آن را در فرمول بالا قرار می دهیم و نهایتا چاپ یا نمایش خروجی در تابع print



مثلا طول فنر در صورت دریافت وزنه ۵ کیلویی ۱۴ سانتیمتر خواهد شد.



عملگرهای مقایسه ای در پایتون

رایانه ها برای مقایسه دو داده از نمادهایی استفاده می کنند. برنامه نویسان به نمادهای جدول زیر عملگرهای مقایسه ای می گویند.

عملگرهای مقایسه ای در پایتون مقادیر دو طرف را مقایسه کرده و رابطه بین آن ها را تعیین می کنند. این عملگرها به طور معمول در دستورات شرطی به کار می‌روند به این ترتیب که باعث ادامه یا توقف دستور شرطی می‌شوند. جدول زیر عملگرهای مقایسه‌ای در پایتون را نشان می‌دهد:

نتایج عملگرهای مقایسه ای درست (True) یا غلط (False) است.

علامت	شرح	مثال
==	مساوی	a = = b
!=	نامساوی	a ! = b
<	کوچک‌تر	a < b
>	بزرگ‌تر	a > b
<=	کوچک‌تر مساوی	a < = b
>=	بزرگ‌تر مساوی	a > = b

استفاده از شرط در برنامه نویسی

بسیاری از کارهایی که ما روزانه انجام می دهیم، بستگی به این دارد که شرطی برقرار باشد و یا برقرار نباشد. به کمک شرط‌ها در زبان‌های برنامه نویسی می‌توانیم تصمیم‌گیری کنیم. با استفاده از دستور if در پایتون می‌توان مشخص کرد که اگر یک یا چند شرط برقرار بود، کدهای خاصی را اجرا کند.

مثلا به پایتون بگوییم :

- اگر عدد زوج بود آن را بر ۲ تقسیم کن و اگر فرد بود آن در ۳ ضرب کن
- اگر رمز ورودی "admin" بود پیام "رمز معتبر است" را نمایش بده
- اگر عدد کمتر از ۵ بود آن را به توان ۲ برسان
- و.....

دستور شرط در یک حالت

```
If شرط :
    دستورات شرط
```

در مواقعی که شرط دارای دو حالت باشد، از دستورif-else استفاده می کنیم. در این دستور در صورتی که شرط if برقرار باشد دستورات آن اجرا می شود و در غیر این صورت دستورات else اجرا می شوند.

معنا و مفهوم else (در غیر این صورت – وگرنه – اگر نباشد) و ..... می باشد.

دستور شرط در زمانی که شرط دو حالت داشته باشد

```
If شرط :
    دستورات شرط
else:
    دستورات بعدی
```

مثال ۸

برنامه ای بنویسید که یک عدد دریافت کند حال اگر عدد بزرگتر از ۱۰ بود آن را چاپ و در غیر این صورت آن را در ۲ ضرب کند.

پاسخ :

```
h= int(input("input number = "))
if (h>10):
    print(h)
else:
    print(h*2)
```

مثال ۹

برنامه ای بنویسید که یک عدد از ورودی دریافت کند و سپس مشخص کند که این عدد زوج است یا فرد.

پاسخ :

در این بلوک شرطی ما دو بلوک شرط با if می نویسیم

شرط اول می نویسیم اگر عدد بر ۲ بخشپذیر بود "عدد زوج است" را چاپ کن

شرط دوم می نویسیم اگر عدد بر دو بخشپذیر نبود "عدد فرد است" را چاپ کن

```
h= int(input("input number = "))
if (h%2)==0:
    print("عدد زوج است")
if (h%2)==1:
    print("عدد فرد است")
```

توجه :

برای راحتی و کد نویسی کمتر بهتر است از دستورif-else استفاده می کنیم.

در صفحه بعد از این دستور برای جواب سوال استفاده می کنیم.

جواب سوال قبلی را با دستورif-else بنویسید.

```
h= int(input("input number = "))
if (h%2)==0:
    print("عدد زوج است")
else:
    print("عدد فرد است")
```

در این بلوک شرطی ما از دستورif-else استفاده کردیم

در شرط نوشتیم اگر عدد بر ۲ بخشپذیر بود "عدد زوج است" را چاپ کن و در غیر این صورت "عدد فرد است" را چاپ کن

مشاهده می کنید که این دستور ساده تر و روان تر از دستور قبلی است و در موارد زیادی هم از کد نویسی کمتری استفاده می شود.

در صفحات بعدی مثال های دیگری از دستورات شرطی را خواهید دید.

مثال ۱۰

برنامه ای بنویسید که دو عدد از ورودی دریافت و سپس عدد بزرگ تر را چاپ کند.

پاسخ :

دو عدد از ورودی دریافت و آن ها را در دو متغیر h و b ذخیره می کنیم در دستورات بعدی می نویسیم اگر عدد h از عدد b بزرگتر باشد آن را نمایش بده و در غیر این صورت عدد b را نمایش بده

```
h= int(input("number 1 = "))
b= int(input("number 2 = "))
if h>b :
    print(h,"عدد بزرگتر")
else :
    print(b," عدد بزرگتر")
```

مثال ۱۱

برنامه ای بنویسید که سه عدد از ورودی دریافت و سپس عدد بزرگ تر را چاپ کند.

پاسخ :

سه عدد از ورودی دریافت و آن ها را در متغیرهای a و b و c ذخیره می کنیم در دستورات بعدی می نویسیم اگر عدد a از دو عدد b و c بزرگتر باشد a را نمایش بده و در غیر این صورت اگر عدد b از دو عدد a و c بزرگتر باشد b را نمایش بده و در غیر این صورت ها c را نمایش بده

```
a= int(input("number1 = "))
b= int(input("number2 = "))
c= int(input("number3 = "))
if a>=b and a>=c:
  print("Max : ",a)
elif b>=a and b>=c:
   print("Max : ",b)
else:
    print("Max : ",c)
```

مثال ۱۲

برنامه ای بنویسید که نام کاربری و گذرواژه را برای ورود به سیستم دریافت کند. در صورتی که نام کاربری Admin وگذرواژه 12345678 بود پیغام "خوش آمدید" صادر شود و در غیر این صورت پیغام "دسترسی غیر مجاز است" صادر شود.

پاسخ :

در این کد نوشتیم اگر ورودی متغیر a فقط Admin و ورودی متغیر b فقط گذرواژه 12345678 باشد پیغام "خوش آمدید" چاپ و نمایش داده شود و در غیر این صورت پیغام "دسترسی غیر مجاز است" چاپ شود.

```
a= input("username = ")
b= input("password = ")
if a=="admin" and b=="12345678":
    print("خوش آمدید")
else:
    print("دسترسی غیر مجاز است")
```

مثال ۱۳

برنامه ای بنویسید که طول سه پاره خط را از ورودی دریافت و سپس مشخص کند که آیا می توان با این سه پاره خط مثلثی رسم کرد یا خیر.

پاسخ :

شرط تشکیل یک مثلث این است که طول هر ضلع مثلث از مجموع دو ضلع دیگر کمتر باشد یا مجموع طولهای دوضلع از ضلع سوم بیشتر باشدکه در کدهای پایین این شرط نوشته شده است. سه متغیر a,b,c سه ضلع مثلث هستند و در خط چهارم هم شرط ایجاد مثلث نوشته شده است.

```
a=int(input("طول ضلع اول = "))
b=int(input("طول ضلع دوم = "))
c=int(input("طول ضلع سوم = "))
if a+b>c and b+c>a and a+c>b:
    print("با این سه پاره خط می توان مثلث ساخت")
else :
    print("با این سه پاره خط نمی توان یک مثلث ساخت")
```

کتابخانه های پایتون

پایتون دارای تعداد زیادی از کتابخانه‌های آماده شده می‌باشد که این کتابخانه‌های پایتونی به ما کمک می‌کند تا بسیاری از کار ها را دیگر خودمان انجام ندیم و آنها را به کتابخانه بسپاریم وتمرکز خود را روی برنامه ای که می‌نویسیم بگذاریم. کتابخانه (ماژول) مجموعه ای از کدهای آماده و نوشته شده در زمینه های مختلف است که توسط افراد نوشته می شود بعضی از این کتابخانه ها به همراه پایتون وجود دارند و برای استفاده باید آنها را با نوشتن کد مربوطه فراخوانی کرد. کتابخانه ها می تواند در برنامه های متعددی استفاده شود که این کار باعث پرهیز از تکرار می شود که در برنامه نویسی بسیار مهم است.

در پایین نام چند کتابخانه یا ماژول در پایتون را می بینید.

Pillow - Matplotlib - Numpy - OpenCV Python - Requests - Keras - TensorFlow - Theano - NLTK - Fire - Arrow - FlashText - Scipy - SQLAlchemy - wxPython - Cirq - PyTorch - Luminoth - Delorean - Poetry - Gensim - Pandas - Pytil - Scikit Learn - NetworkX - PyGame - TextBlob - Mahotas - Turtle

یکی از کتابخانه های پایتون، کتابخانه Turtle یا لاک پشت است با کتابخانه Turtle می توانید اشکال و الگوهای مختلفی را ترسیم کنید به وسیله آن می توان یک لاک پشت ایجاد کرد که با کنترل روی آن می توان اشکال مختلفی را رسم نمود.

برای استفاده از Turtle ، آن را به پایتون import می کنیم برای این کار لازم است کد پایین را در پایتون تایپ کنید.

```
import turtle
```

بعد از وارد کردن کد بالا یک متغیر به آن نسبت می دهیم مثلا laki از اینجا به بعد نام لاکپشت ما لاکی است و هر کدی که بنویسیم باید در متغیر laki ذخیره شود.

```
import turtle
laki=turtle.Turtle()
laki.shape('turtle')
```

وقتی می خواهیم لاک پشت کاری انجام دهد، یک نقطه جلوی نام آن قرار می دهیم و سپس آن کار را می نویسیم. به کارهایی که لاک پشت انجام می دهد، متد می گوییم. متدهای زیادی از پیش برای لاک پشت تعریف شده است که به تدریج با آن ها آشنا می شویم.

تغییر شکل لاکی به لاکپشت

لاکی بعد از اجرا ۱۰۰ پیکسل به جلو برود.

```
import turtle
laki=turtle.Turtle()
laki.shape('turtle')
laki.forward(100)
```

اگر کد بالا را در پایتون کپی یا بنویسید و آن را اجرا کنید خواهید دید که لاکپشت ۱۰۰ پیکسل به جلو می رود. برای دیدن خروجی بعد از نوشتن کدها از منوی Run گزینه Run Module را انتخاب کنید تا خروجی برنامه را ببینید.

مثال ۱۴

کارت شناسایی روبرو را با لاکی طراحی کنید. پاسخ و کد برنامه در پایین


```
import turtle
laki=turtle.Turtle()
laki.color ("red")
laki.width (5)
laki.forward(200)
laki.right (90)
laki.forward(160)
laki.right (90)
laki.forward(300)
laki.right (90)
laki.forward(160)
laki.right (90)
laki.forward(100)
laki.penup()
laki.goto(-85,-35)
laki.color ("blue")
laki.write("Name : Ali",font=('tahoma',16))
laki.goto(-85,-70)
laki.write("Last Name: Bahrami",font=('tahoma',16))
laki.goto(-85,-100)
laki.write("Student Code: 1234567890",font=('tahoma',16))
laki.goto(-85,-130)
laki.write("Mobile: 09131111111",font=('tahoma',16))
laki.hideturtle ()
```

شرح کدهای صفحه قبل

laki.color ("red")

تغییر رنگ خط و متن ترسیمی به قرمز

laki.width (5)

تعیین ضخامت خط به ۵ پیکسل

laki.forward(200)
laki.right (90)

حرکت ۲۰۰ پیکسل لاکی به جلو و سپس چرخش ۹۰ درجه به راست

laki.penup()

عدم خط کشی توسط لاکی (لاکی خط نکش)

laki.goto(-85,-35)

رفتن لاکی به مختصات تعیین شده طبق محور مختصات

laki.write()

لاکی متن داخل پرانتز را تایپ کن

laki.hideturtle ()

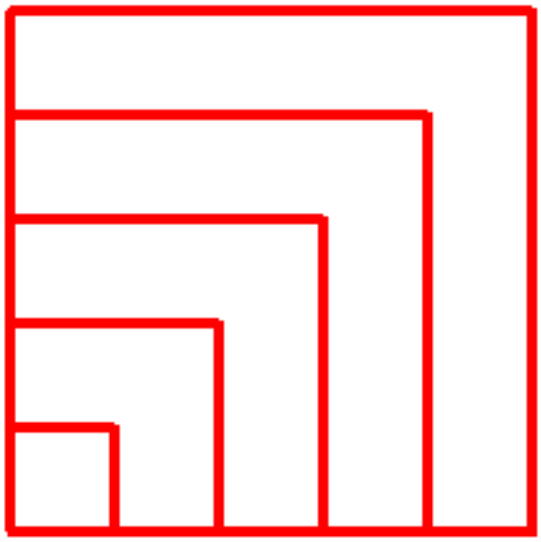
بعد از اتمام ترسیم لاکی را پنهان کن

Name : Ali
Last Name: Bahrami
Student Code: 1234567890
Mobile: 09131111111

نتیجه کد نویسی بالا طراحی این کارت است که می توان با دستکاری در کد ها کارت مشابه دیگری با متن دلخواه ایجاد کرد

مثال ۱۵

شکل زیر را با لاکی ترسیم کنید.



```
import turtle
laki=turtle.Turtle()
laki.color ("red")
laki.width (4)
laki.forward(200)
laki.left (90)
laki.forward(200)
laki.left (90)
laki.forward(200)
laki.left (90)
laki.forward(200)
laki.left (90)
laki.forward(160)
laki.left (90)
laki.forward(160)
laki.left (90)
laki.left (90)
laki.forward(160)
laki.left (90)
laki.forward(120)
laki.left (90)
laki.forward(120)
laki.left (90)
laki.forward(120)
laki.left (90)
laki.forward(120)
laki.left (90)
laki.forward(80)
laki.left (90)
laki.forward(80)
laki.left (90)
laki.forward(80)
laki.left (90)
laki.forward(80)
laki.left (90)
laki.forward(40)
laki.left (90)
laki.forward(40)
laki.left (90)
laki.forward(40)
laki.penup()
laki.hideturtle ()
```

در ترسیم این شکل به لاکی دستور می دهیم

با خط قرمز و ضخامت ۴ پیکسل ۵ مربع تو در تو بکشد

با نگاه کردن به کد ها متوجه نحوه ترسیم خواهید شد.

مربع اول ۲۰۰ پیکسل به جلو

مربع دوم ۱۶۰ پیکسل به جلو

مربع سوم ۱۲۰ پیکسل به جلو

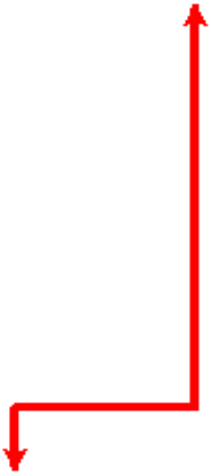
مربع چهارم ۸۰ پیکسل به جلو

و مربع پنجم ۴۰ پیکسل رو به جلو

و چرخش ۹۰ درجه به چپ در تمام خطوط

مثال ۱۶

شکل زیر را با لاکی ترسیم کنید.



```
import turtle
laki=turtle.Turtle()
laki2=turtle.Turtle()
laki.color ("red")
laki2.color ("red")
laki.width (4)
laki2.width (4)
laki.left (90)
laki.forward(200)
laki2.left (180)
laki2.forward(80)
laki2.left (90)
laki2.forward(30)
laki.penup()
```

در ترسیم این شکل از دو لاکی استفاده می کنیم

این دو لاکی با خط قرمز و ضخامت ۴ پیکسل خطوط را ترسیم می کنند.

لاکی اول خط رو به بالا و

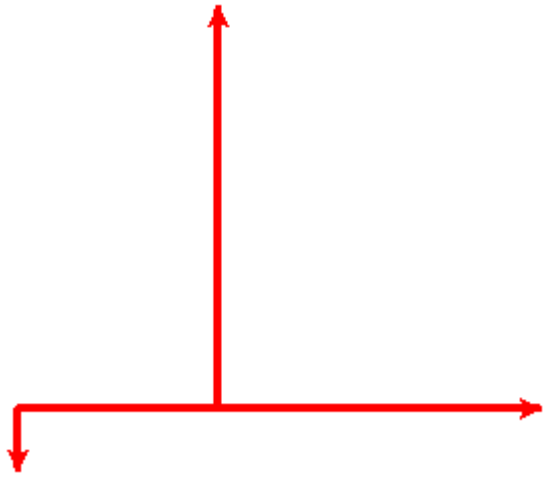
لاکی دوم خط افقی و عمودی پایین

را ترسیم می کنند.

به کدها توجه کنید

مثال ۱۷

شکل زیر را با لاکی ترسیم کنید.



```
import turtle
laki=turtle.Turtle()
laki2=turtle.Turtle()
laki3=turtle.Turtle()
laki.color ("red")
laki2.color ("red")
laki3.color ("red")
laki.width (4)
laki2.width (4)
laki3.width (4)
laki.forward(160)
laki2.left (90)
laki2.forward(200)
laki3.left (180)
laki3.forward(100)
laki3.left (90)
laki3.forward(30)
laki.penup()
```

در ترسیم این شکل از سه لاکی استفاده می کنیم

این سه لاکی با خط قرمز و ضخامت ۴ پیکسل خطوط را ترسیم می کنند.

لاکی اول خط سمت راست

لاکی دوم خط رو به بالا

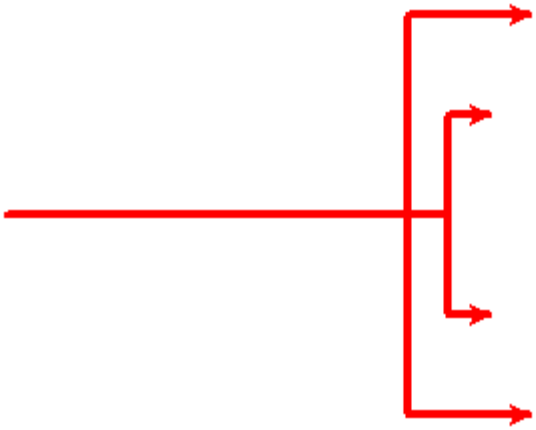
لاکی سوم خط سمت چپ و خط رو به پایین

را ترسیم می کنند.

به کدها توجه کنید

مثال ۱۸

شکل زیر را با لاکی ترسیم کنید.



```
import turtle
laki=turtle.Turtle()
laki2=turtle.Turtle()
laki3=turtle.Turtle()
laki4=turtle.Turtle()
laki.color ("red")
laki2.color ("red")
laki3.color ("red")
laki4.color ("red")
laki.width (4)
laki2.width (4)
laki3.width (4)
laki4.width (4)
laki.forward(200)
laki.left (90)
laki.forward(100)
laki.right (90)
laki.forward(60)
laki2.forward(200)
laki2.right (90)
laki2.forward(100)
laki2.left (90)
laki2.forward(60)
laki3.forward(220)
laki3.left (90)
laki3.forward(50)
laki3.right (90)
laki3.forward(20)
laki4.forward(220)
laki4.right (90)
laki4.forward(50)
laki4.left (90)
laki4.forward(20)
laki.penup()
```

در ترسیم این شکل از چهار لاکی استفاده می کنیم

این چهار لاکی با خط قرمز و ضخامت ۴ پیکسل خطوط را ترسیم می کنند.

هر لاکی یک خط را ترسیم می کند.

به کدها توجه کنید

حلقه تکرار در پایتون

در کد نویسی با پایتون ممکن است یک کد یا دستور چندین بار تکرار شود. که باعث شلوغ شدن صفحه کد نویسی و همچنین سردرگمی کد نویس خواهد شد. برای جلوگیری از این وضعیت باید از حلق تکرار در کد نویسی استفاده کرد.

حلقه for در پایتون برای تکرار کردن کاری در اشیای قابل تکرار، مورد استفاده قرار می‌گیرد. دستور این کد به همراه تابع range به صورت زیر است.

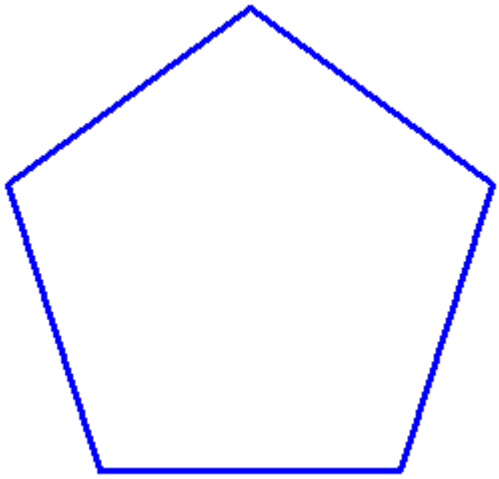
```
for i in range():
```

برای تعداد تکرار متغیر i در داخل پرانتز یک عدد قرار می دهیم. این تابع به تعداد عدد داخل پرانتز دستور را تکرار می کند.

مثال:

کد ترسیم ۵ ضلعی

```
import turtle
laki=turtle.Turtle()
laki.color ("blue")
laki.width (3)
for i in range(5):
    laki.forward(150)
    laki.left(72)
    laki.hideturtle()
```



در این کد به لاکی دستور می دهیم با رنگ آبی و ضخامت خط ۳ پیکسل ۵ بار عمل ۱۵۰ پیکسل رو به جلو و چرخش ۷۲ درجه ای به سمت چپ را انجام دهد که نتیجه این عمل رسم ۵ ضلعی است.

مثال ۱۹

با دستور for کدهای رسم شش ضلعی و هشت ضلعی را بازنویسی کنید.

کد ۶ ضلعی

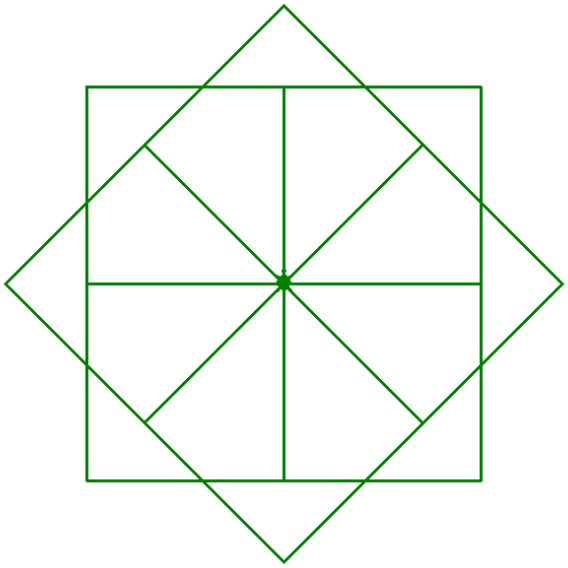
```
import turtle
laki=turtle.Turtle()
laki.color ("blue")
laki.width (3)
for i in range(6):
    laki.forward(150)
    laki.left(60)
    laki.hideturtle()
```

کد ۸ ضلعی

```
import turtle
laki=turtle.Turtle()
laki.color ("blue")
laki.width (3)
for i in range(8):
    laki.forward(100)
    laki.left(45)
    laki.hideturtle()
```

مثال ۲۰

با دستور for شکل زیر را رسم کنید.



```
import turtle
laki=turtle.Turtle()
laki.shape("turtle")
laki.color("green")
laki.width(3)
for a in range(2):
 for b in range(4):
  for c in range(4):
   laki.forward(200)
   laki.left(90)
  laki.left(90)
 laki.left(45)
```

در این کد از سه حلقه تو در تو استفاده کردیم و به لاکی گفتیم۲ سری مربع رسم کن ۴ مربع معمولی و ۴ مربع با زاویه ۴۵ درجه نسبت به مربع های قبلی

مثال ۲۱

برنامه ای بنویسید که تعداد اضلاع شکل و تعداد تکرار آن را از ورودی دریافت و سپس طرح کامل را ترسیم کند.

```
import sys
import turtle
laki=turtle.Turtle()
laki.shape("turtle")
laki.color("green")
laki.width(3)
laki.speed(100)
n=int(turtle.textinput("N.Side","Number of sides: "))
if x<3 :
    print("error")
    sys.exit("تعداد ضلع باید ۳ یا بیشتر از ۳ باشد")
m=int(turtle.textinput("repeat-polygon","The number of repetitions: "))
for i in range(m):
  for d in range(1):
   for e in range(n):
    laki.forward(50)
    laki.left(360/n)
  laki.left(360/m)
```

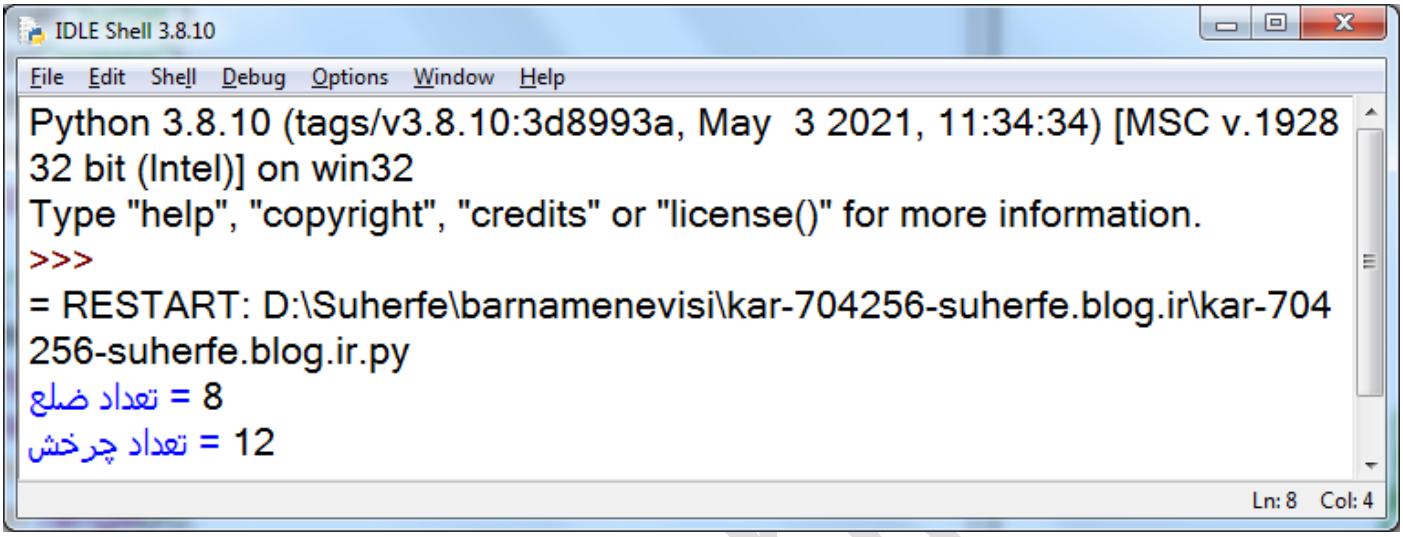
سرعت حرکت لاکی

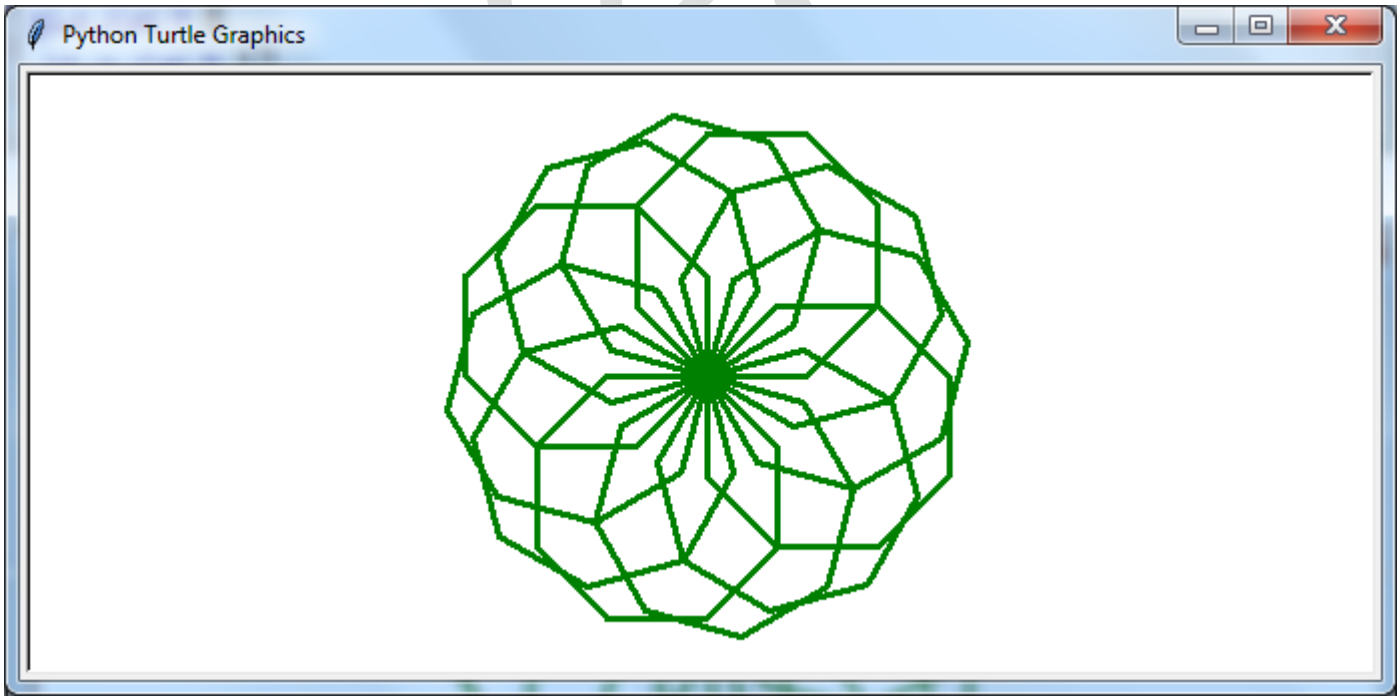
اگر تعداد ضلع کمتر از سه دریافت شود تابع sys پیغام خطا صادر کرده و برنامه را متوقف می کند.

متغیر n تعداد ضلع و متغیر m تعداد چرخش را از ورودی دریافت می کند و بر اساس محیط ۳۶۰ درجه ای دایره زاویه چرخش را محاسبه کرده و چندضلعی را ترسیم می کند.

نحوه اجرا در صفحه بعد.

کد بالا را در پایتون کپی یا بنویسید برای دیدن خروجی بعد از نوشتن کدها از منوی Run گزینه Run Module را انتخاب کنید برنامه از شما تعداد ضلع و چرخش را دریافت می کند و آن را ترسیم می کند.





مثال های ترسیم با ترتل در پایتون

دوستان در صفحات قبل با مباحثی همچون متغیرها، عملگرها، ساختارهای شرطی if-elif، if-else و حلقه یا همان ساختار تکرار for آشنا شدید. و چگونگی دریافت مقادیر از ورودی و چاپ مقدار متغیرها در خروجی را دیدید. همان طورکه می دانید متغیرها، خانه ای از حافظه اند که مقداری را به طور موقت در خود ذخیره می کنند. ازاین رو برنامه نویسان با دو پرسش مواجه می شوند.

۱- آیا می توان یک ساختار داده ای را ایجاد کرد که چندین مقدار را داخل خودش ذخیره کند؟

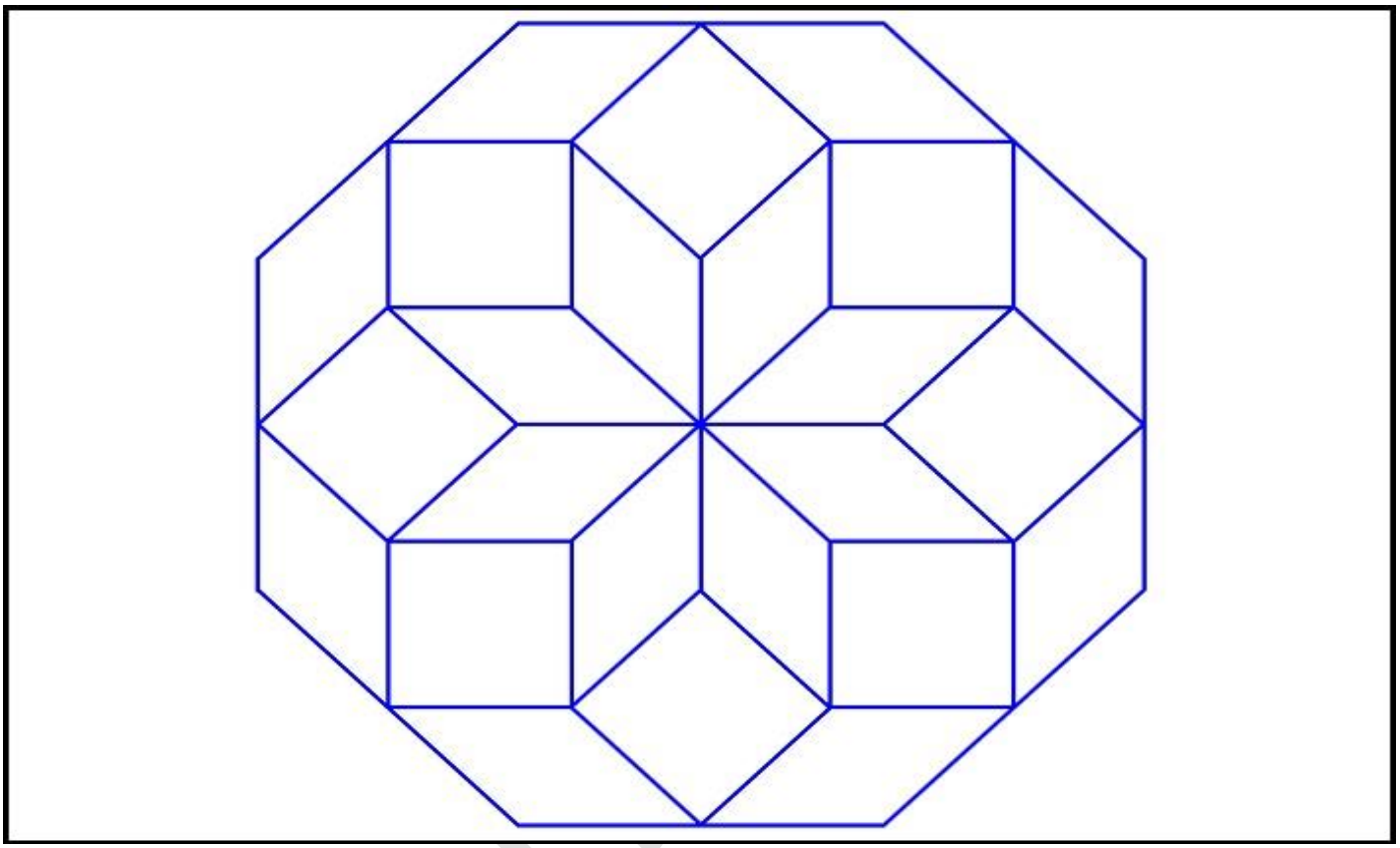
۲- چگونه می توان داده های ورودی از کاربر را به صورت دائمی در حافظه رایانه ذخیره کرد؟

برای پاسخ به پرسش اول باید از ساختارهای آرایه ای استفاده کرد و در پاسخ به پرسش دوم باید متغیرهای برنامه را روی حافظه دائمی رایانه مثل هارد دیسک ذخیره کرد. به همین دلیل در ادامه به بررسی مبحث آرایه ها و فایل ها می پردازیم. همچنین از مفهوم تابع برای کاهش تعداد خطوط برنامه و خوانایی بهتر آن استفاده خواهیم کرد. در ضمن می توان یک فایل پایتونی (ماژول)، ایجاد کرده و داخل آن مواردی مانند تابع function فهرست list چندتایی tuple، مجموعه set و دیکشنری dictionary را نوشت و در سایر برنامه های دیگر از آن ماژول استفاده کرد.

ابتدا مثال صفحه بعد را بررسی کنید.

مثال ۲۲

قطعه برنامه زیر را که توسط کتابخانه turtle نوشته شده است، در محیط IDLE بنویسید. خروجی آن به شکل زیر است: خطوط برنامه را تحلیل نموده و شرح دهید.



suherfe.blog.ir

File Edit Format Run Options Window Help

```
1 import turtle
2 t=turtle.Turtle()
3 t.shape("turtle")
4 t.color("blue")
5 t.speed("fast")
6 screen = turtle.Screen()
7 t.width(3)
8 for i in range(8):
9     for j in range(8):
10        t.forward(100)
11        t.right(45)
12    t.right(45)
13 t.hideturtle()
```

Ln: 15 Col: 0

پاسخ:

خط ۱ :فراخوانی کتابخانه ترتل turtle برای ترسیم

خط ۲ :نسبت دادن کتابخانه ترتل به یک متغیر دلخواه مانند t

خط ۳ :تعیین شکل ترسیم کننده به صورت لاک پشت turtle

خط ۴ :تعیین رنگ آبی blue برای ترسیم خطوط

خط ۵ :تعیین سرعت ترسیم به صورت سریع fast

خط ۶ :ساخت صفحه ترسیم جهت ترسیم شکل

خط ۷ :تعیین ضخامت ۳ پیکسل برای ترسیم خطوط

خط ۸ :ایجاد حلقه تکرار چرخش ۸ ضلعی

خط ۹ :ایجاد حلقه تکرار ترسیم ۸ ضلعی

خط ۱۰ :ترسیم خط راست به طول ۱۰۰ پیکسل

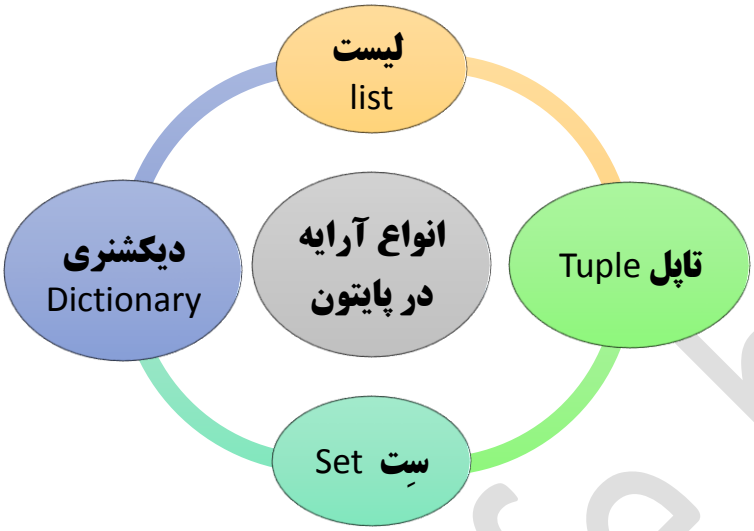
خط ۱۱ :چرخش ۴۵ درجه ترتل به طرف راست جهت ترسیم ۸ ضلعی

خط ۱۲ :چرخش ۴۵ درجه ۸ ضلعی به طرف راست و ترسیم دوباره آن

خط ۱۳ :مخفی کردن ترسیم کننده لاک پشت از صفحه بعد از ترسیم

آرایه ها

در زبان پایتون برای نگه داری و ذخیره چندین داده در یک متغیر از آرایه ها استفاده می شود. در پایتون، چهار نوع آرایه وجود دارد که می توان از آنها برای ذخیره انواع داده استفاده کرد این آرایه ها عبارتند از:



لیست، list

تاپل، Tuple

مجموعه یا سِت،Set

دیکشنری،Dictionary

آرایه های لیست و تاپل شباهت زیادی به یکدیگر دارند، در لیست، عناصر آن را می توان تغییر داد. در حالی که در تاپل این کار امکان پذیر نیست.

دو آرایه سِت و دیکشنری هم با هم شباهت دارند. در دیکشنری، عناصر و آیتم ها از ترکیب (کلید : مقدار) key:value ایجاد می شوند. اما در آرایه سِت، عناصر فقط دارای مقدارvalue هستند. در ادامه در مورد هر کدام از این آرایه ها بیشتر توضیح داده می شود.

لیست list

لیست یک آرایه مهم در پایتون است. به کمک لیست‌ها می‌توانیم دنباله‌ای از داده‌ها را در یک متغیر ذخیره کرده و روی آن‌ها عملیات‌های مختلفی را اجرا کنیم. لیست می تواند مجموعه ای از آیتم های مختلف را در خود ذخیره کند.
لیست شامل مجموعه‌ای از عناصر به صورت ترتیبی و قابل تغییر است. لیست در پایتون با کروشه به شکل [] مشخص می‌شود؛ به طوری که با علامت کروشه باز لیست آغاز شده و تا علامت کروشه بسته ادامه می‌یابد. همچنین هر عنصر درون لیست به وسیله کاما (ویرگول) به شکل , از یکدیگر جدا شده و می‌تواند از هر نوع داده دلخواهی باشد.

لیست های پایتون چند ویژگی مهم دارند:

آیتم های لیست در داخل کروشه [ ] قرار می گیرند.
لیست ها منظم و دارای ترتیب هستند. یعنی آیتم ها در لیست ترتیب ورودی را حفظ می کنند، بنابراین می توانید از ایندکس (شماره گذاری) برای دسترسی به آیتم ها استفاده کنید.
لیست ها تغییر پذیر هستند یعنی می توانید آیتم های آن را کم یا زیاد کنید.
آیتم های لیست می توانند تکراری باشند.
آیتم های لیست می توانند از انواع مختلفی از داده ها باشند.

چند مثال از ساختارهای لیست

```
myList1= [60, 26,'ahmad',39, 17,True,'zahra',405]
myList2= ["Omid", 217, 22.5, "Sistan"]
myList3= ['u', 'p', 'a', 'o', 'd', 'm', 'n', 'e', 'f', 'h']
myList4= ["shiva", "omran", "nahid", "reza" ,"ariya"]
```

دسترسی به عناصر لیست

دسترسی به اعضای لیست بر اساس اندیس‌های آن صورت می‌گیرد. اندیس‌ها اعدادی هستند که موقعیت هر عنصر در لیست را مشخص می‌کنند.

اگرمبنای شماره‌گذاری عناصر لیست از چپ به راست باشد، باید از صفر شروع کنیم. **مثال :**

myList = [12,15,23,8,43, 560,12,36,9]

۱۲	۱۵	۲۳	۸	۴۳	۵۶۰	۱۲	۳۶	۹
۰	۱	۲	۳	۴	۵	۶	۷	۸

myList[2] ⟶ 23 myList [5] ⟶ 560

اگر مبنای شماره‌گذاری عناصر لیست از راست به چپ باشد. باید از ۱- شروع کنیم. **مثال :**

myList =[12 ,15,''omid'',8, 43,True,12, 36,9]

۱۲	۱۵	'omid'	۸	۴۳	True	۱۲	۳۶	۹
-۹	-۸	-۷	-۶	-۵	-۴	-۳	-۲	-۱

myList [-7] ⟶ ''omid'' myList[-8]⟶15

کار با عناصر لیست - عملیات روی لیست ها

علاوه بر دسترسی به عناصر یک لیست، نیاز به وارد کردن، خارج کردن یا تغییر داده‌ها در لیست خواهیم داشت.برای هر کدام از این کارها روش‌ها یا توابع مختلفی در پایتون وجود دارد. در ادامه به مهم‌ترین و کاربردی‌ترینِ آن‌ها را بررسی می‌کنیم.

در پایتون، عملیات روی عناصر لیست ها عبارت اند از:

اضافه کردن، حذف، مرتب سازی و پیمایش عناصر.

افزودن عنصر جدید به لیست

برای افزودن یک عنصر جدید به لیست فعلی، سه روش وجود دارد. در دو روش اول، عنصر به انتهای لیست اضافه می‌شود؛ اما به کمک روش سوم می‌توانیم عنصر را در اندیس دلخواه خود قرار دهیم. به مثال پایین توجه کنید:

ابتدا لیست example را با ۴ مقدار اولیه ایجاد کرده و سپس با سه روش به آن عناصری را اضافه می‌کنیم

```
example = [802, 18, 24, 90]
```

۱- افزودن عنصر جدید به لیست با تابع append

تابع append() ، مقدار ورودی خود را به عنوان عنصر جدید در انتهای لیست درج می‌کند. این تابع صرفاً یک ورودی گرفته و نوع داده‌ای آن اهمیتی ندارد.

```
example = [802, 18, 24, 90]
example.append(42)
print(example)
output: [802, 18, 24, 90, 42]
```

خروجی

۲- افزودن سریع عنصر به لیست پایتون با علامت جمع(+)

این روش می‌تواند یکی از سریع‌ترین روش‌های افزودن عنصر یا عناصر جدید به لیست مورد نظرمان باشد.

```
example = [802, 18, 24, 90]
example = example + [405]
print(example)
output: [802, 18, 24, 90, 405]
```
خروجی ←

۳- استفاده از متد insert

متد ()insert روی لیست دو ورودی می‌گیرد:

آرگومان اول : شماره ایندکس مورد نظر به صورت عدد صحیح integer

آرگومان دوم : عنصر مورد نظر برای افزودن به لیست

برای شماره ایندکس وارد شده دو حالت وجود دارد:

۱- ایندکس داخل بازه ایندکس‌های فعلی است؛ پس عنصر در خانه مورد نظر درج شده و عنصر خانه‌های بعد از آن یکی به جلو رانده می‌شوند.

```
example = [802, 18, 24, 90]
example.insert(1, 110)
print(example)
output: [802, 110, 18, 24, 90]
```
خروجی ←

۲- ایندکس خارج از محدوده و اندازه لیست است؛ در این‌صورت عنصر در انتهای لیست اضافه خواهد شد.

```
example = [802, 18, 24, 90]
example.insert(70, 110)
print(example)
output: [802, 18, 24, 90, 110]
```
خروجی ←

حذف عناصر از لیست

حذف عنصر به کمک دستور del

به کمک دستور del ، می‌توان کل لیست یا یک عنصر از آن را حذف کرد.
اگر نام لیست یا لیست همراه به ایندکس را در جلوی کلمه کلیدی del قرار دهیم، لیست یا عنصر مورد نظر به طور کامل حذف خواهد شد.

```
myList= ['u', 'p', 'a', 'o', 'd', 'm', 'n', 'e', 'f', 'h']
del myList [5]
print(myList)
output: ['u', 'p', 'a', 'o', 'd', 'n', 'e', 'f', 'h']
```

خروجی

استفاده از متد remove برای حذف عنصر از لیست

این متد بر روی لیست یک ورودی می‌گیرد. آنچه که به عنوان ورودی گرفته را در لیست جستجو کرده و اولین موردی که مطابقت داشت را حذف می‌کنید.در لیست زیر، ما سه بار حرف n را داریم. با صدا زدن متد remove('n') اولین موردی که در لیست پیدا می‌شود (یعنی ایندکس۳) حذف خواهد شد.

```
myList= ['u', 'p', 'a', 'n', 'd', 'm', 'n', 'e', 'f', 'n']
myList.remove('n')
print(myList)
output: ['u', 'p', 'a', 'd', 'm', 'n', 'e', 'f', 'n']
```

خروجی

حذف آخرین عنصر لیست با متد pop

با کمک تابع pop() می‌توانیم آخرین عنصر را از لیست خارج و حذف کنیم.

```
mylist=["shiva", "omran", "nahid","ariya","reza"]
mylist.pop()
print(mylist)
```

output: ["shiva", "omran", "nahid", " ariya "] ← خروجی

حذف تمام عناصر لیست با متد clear

اگر بخواهید تمام اعضای لیست را حذف کنید، با متد clear() روی یک لیست، می توانید تمام اعضای آن را حذف و در نهایت یک لیست خالی خواهید داشت.

```
names = ["sara", "omid", "amir", "roya" ,"ehsan"]
names.clear()
print( names )
```

output: [] ← خروجی

مرتب سازی عناصر لیست

با مِتُد sort می توان عناصر لیست را به طور صعودی یا نزولی مرتب کرد

```
myList=['orange','mango','kiwi','banana']
myList.sort()
print(myList)
```

خروجی قطعه برنامه بالا یک لیست به شکل زیر است که به ترتیب حروف الفبای انگلیسی و به صورت صعودی مرتب شده است:

output: ['banana' ,'kiwi' ,'mango' ,'orange']

همچنین برای مرتب سازی به ترتیب نزولی از روش زیر استفاده می کنیم:

```
myList=['banana','mango','kiwi','orange']
myList.sort(reverse = True)
print(myList)
```

خروجی قطعه برنامه بالا یک لیست به شکل زیر است که به صورت نزولی بر اساس حروف انگلیسی مرتب شده است:

output: ['orange' ,'mango' ,'kiwi' ,'banana']

پیمایش عناصر لیست

پیمایش عناصر لیست به معنای حرکت روی عناصر است.

فرض کنید که یک لیست به شکل زیر داریم:

```
myList=[80,549,300, 5,303]
for i in myList:
print(i)
```

در قطعه برنامه بالا توسط یک حلقهfor روی عناصر لیست حرکت کرده و آن ها را در یک ستون چاپ می کند. در اینجا i به معنی عناصر لیست است.

```
80
549
300
5
303
```

اگر بخواهیم روی عناصر لیست حرکت کرده و آن ها را در یک سطر چاپ کنیم دستور چاپ را در خط سوم به صورت پایین تغییر می دهیم

```
myList=[80,549,300, 5,303]
for i in myList:
print(i,end=' ')
output: 80 549 300 5 303
```

خروجی

مثال ۲۳

چگونه می توان توسط یک ساختار شرطی در داخل حلقه for عناصر زوج را در لیست صفحه قبل نمایش داد؟

جواب

```
myList=[100,213,350,625,140]
for i in myList:
    if i%2==0:
        print(i,end=' ')
```

در این کد ما در حلقه for یک شرط به کار بردیم و در آن نوشتیم که اگر اعداد لیست را بر ۲ تقسیم کنیم و باقی مانده صفر شود یعنی آن عدد زوج است پس آن را نماش بده

شباهت های لیست با تاپل

- هر دو نوع داده یک دنباله هستند
- لیست و تاپل می‌توانند به عنوان نگهدارنده ی از دیتا های دیگر باشند
- لیست و تاپل هر دو مرتب ، و از طریق ایندکس قابل دسترسی هستند

تفاوت های لیست با تاپل

- تاپل برخلاف لیست پس از ایجاد شدن دیگر قابل تغییر نیست
- تاپل کمی از لیست سریع تر است
- عناصر تاپل به جای کروشه در داخل دو پرانتز قرار می گیرند

تفاوت ها وشبا هت های لیست با تاپل در یک نگاه

لیست (list)		چندتایی(tuple)	
انتساب	myList= [10,15,36,15] myList [2] =100 print(myList) ⟶ [10,15,100,15]	انتساب	myTuple = (10 ,15 ,36,15) myTuple [2] =100 ⟶ Error print(myTuple)
محدوده	myList= [10,15,36,15] print(myList [1: 3]) ⟶ [15,36] print (myList [1:]) ⟶ [15,36,15]	محدوده	myTuple= (10,15,36,15) print(myTuple [1: 3]) ⟶ (15,36) print (myTuple [: 2]) ⟶ (10,15)
حذف	myList = [10,15,36,15] myList.remove(15) print(myList) ⟶ [10,36,15]	حذف	myTuple = (10,15,36,15) myTuple.remove(15) ⟶ Error
بازکردن	fruits = ["apple", "banana", "cherry"] [green , yellow , red]= fruits print(green , yellow , red) ⟶ apple banana cherry	بازکردن	fruits = ("apple", "banana", "cherry") (green , yellow , red) = fruits print(green , yellow, red) ⟶ apple banana cherry

نکته :

با توجه به جدول بالا، عملیات حذف و انتساب در آرایه تاپل امکان پذیر نیست. زیرا تاپل ، به طور ذاتی قابل تغییر نیست، ولی می توان آن را به لیست تغییر داد. پس از آن، تغییرات لازم مثل حذف و انتساب را روی آن انجام داده و دوباره آن را به تاپل تغییر می دهیم.

حذف عنصر از تاپل

در مثال پایین ابتدا تاپل را به لیست تبدیل کرده و سپس یک عنصر از آن را حذف می کنیم بعد آن را به تاپل تبدیل می کنیم

```
myTuple = ('sara','zahra','majid')
myList = list(myTuple)
myList.remove('sara')
myTuple = tuple(myList)
print(myTuple)
output:('zahra','majid')
```

خروجی برنامه : ←

در کدهای بالا، ابتدا در خط دوم تاپل myTuple به لیست myList تبدیل شده، سپس در خط سوم، عنصر sara را حذف کردیم، آنگاه دوباره در خط چهارم آن را به تاپل تبدیل کرده و درنهایت در خط پنجم آن را چاپ کردیم.

تغییر عناصر تاپل

```
x = ('sara','zahra','majid')
y = list(x)
y[0] ='reza'
x = tuple(y)
print(x)
output:('reza',' zahra ','majid')
```

خروجی برنامه : ←

در کدهای بالا، ابتدا در خط دوم تاپل x به لیست y تبدیل شده، سپس در خط سوم، مقدار عنصر شماره 0 را به reza تغییر دادیم، آنگاه دوباره در خط چهارم آن را به تاپل تبدیل کرده و سرانجام در خط پنجم آن را چاپ کردیم.

متدهای قابل اجرا روی لیست و تاپل

متد len

با این متد میتوان تعداد عناصر لیست و تاپل را شمرد.

تعداد عناصر لیست

```
myList= [99,28,95,75]
print(len(myList))
```

4

تعداد عناصر تاپل

```
myTuple= (100,20,35,62,75,225,305)
print(len(myTuple) )
```

7

متد min

با این متد میتوان کمترین مدار عناصر لیست و تاپل را بدست آورد.

کمترین مقدار عناصر در لیست

```
myList= [99,28,95,75]
print(min(myList))
```

28

کمترین مقدار عناصر در تاپل

```
myTuple= (100,20,35,62,75,225,305)
print(min(myTuple) )
```

20

متد max

با این متد میتوان بیشترین مقدار عناصر لیست و تاپل را بدست آورد.

بیشترین مقدار عناصر در لیست

```
myList= [99,28,95,75]
print(max(myList))
```
99

بیشترین مقدار عناصر در تاپل

```
myTuple= (100,20,35,62,75,225,305)
print(max(myTuple) )
```
305

متد reverse

با این متد میتوان عناصر لیست را معکوس چاپ کرد. اما به علت تغییر ناپذیری تاپل نمی توان عناصر آن را معکوس چاپ کرد.

چاپ معکوس عناصر لیست

```
myList= ['red', 'yellow', 'black', 'blue']
myList.reverse()
print(myLis)
output: ['blue', 'black', 'yellow', 'red']
```

خطا در چاپ معکوس عناصر در تاپل

```
myTuple = ('red', 'yellow', 'black', 'blue')
myTuple.reverse()
print(myTuple)
output:  Error
```

دیکشری در پایتون

در بین ساختارهای داده، دیکشنری پایتون امکان ذخیره‌سازی آیتم‌ها را به صورت جفت کلید - مقدار (Key-Value) فراهم می‌کند. عناصر دیکشنری در داخل آکولاید{}نوشته می شود.
دفترچه تلفن می‌تواند نمونه خوبی برای فهم دیکشنری پایتون باشد. در دفترچه تلفن، اطلاعات بر اساس نام اشخاص و شماره تماس آن‌ها سازمان‌دهی شده است.

مثال ۲۴ :

```
teldict ={'ali':' 0911','hasan':'0937','reza':'0916'}
print(teldict)
```

در برنامه بالا، ابتدا یک دیکشنری به نام teldict با سه آیتم تعریف کردیم.هر آیتم دارای یک **کلید:مقدار** است سپس توسط دستورprint دیکشنری را چاپ می کنیم که خروجی به شکل زیر است:

```
output: {'ali': ' 0911 ', 'hasan': '0937', 'reza': '0916'}
```

مجموعه (set) در پایتون

ست‌ها در پایتون معادل مجموعه‌های ریاضی هستند ست‌ها در این زبان امکان استفاده از عناصر تکراری را نمی‌دهند. برای تعریف مجموعه ها در پایتون همانند دیکشنری باید عناصر را داخل آکولاید {} نوشت . با تعریف یک ست و قرار دادن یک نام برای آن، چون ایندکس عناصر مشخص نیست با پرینت آن در خروجی ممکن است ترتیب قرارگیری با ترتیب تعریف یک ست متفاوت باشد.

```
Set = {"apple", "banana", "cherry"}
print(Set)
Output: {'apple', 'cherry', 'banana'}
```
خروجی ←

حال اگر چندین بار برنامه را اجرا کنید ممکن است خروجی های جدید با خروجی اجرای های دیگر متفاوت باشد و ترتیب چاپ عناصر تغییر کنند مانند:

```
Output: {'apple', 'banana', 'cherry'}
Output: {'cherry', 'banana', 'apple'}
Output: {'cherry', 'apple', 'banana'}
Output: {'banana', 'cherry', 'apple'}
Output: {'banana', 'apple', 'cherry'}
```

به یاد داشته باشید که اگر از عناصر تکراری در داخل ست استفاده کنید هنگام چاپ، عنصر تکراری فقط یک بار نمایش داده می شود مانند :

```
Set = {'apple','banana','cherry','apple','banana','apple'}
print(Set)
Output:{'apple', 'cherry', 'banana'}
```
← یکبار چاپ عناصرتکراری

آشنایی با تابع

تابع چیست؟

به برنامه ای که یک بار نوشته شده و نامی دلخواه برای آن انتخاب می شود، تابع می گویند .تابع را می توان هر چند بار که لازم باشد فراخوانی کرد .این کار باعث کمتر شدن تعداد خطوط برنامه و راحت تر شدن کار برنامه نویس می شود.
هرچه برنامه بزرگ و بزرگ‌تر شود، تابع‌ها به سازمان‌یافته‌تر و قابل مدیریت شدن آن کمک می‌کنند. علاوه بر این، توابع مانع از تکرار برنامه‌نویسی برای یک کار واحد می‌شوند و کدها را قابل استفاده مجدد می‌کنند.
برای تعریف یک تابع باید از دستور def استفاده کرد.
بعد از دستور def نامی دلخواه برای تابع می نویسیم
در داخل پرانتز نام ورودی را نوشته اگر چند ورودی داشتیم بین آنها از علامت , استفاده می کنیم و سپس پرانتز بسته و دو نقطه :
برای تعریف خروجی تابع در پایتون، از کلمه کلیدی return استفاده می‌شود return هر مقداری که در جلویش قرار داشته باشد را به عنوان خروجی تابع به ما باز می‌گرداند.
و سپس نوشتن دستورات دیگر :

برای یادگیری بهتر به خط کدهای صفحه بعد توجه کنید:

مثال ۲۵

```
def sub( a, b ):
    return a / b
a = int(input("Enter First Number: "))
b = int(input("Enter Second Number: "))
result = sub(a,b)
print('result=',result)
```

دراین برنامه یک تابع نوشتیم که دو عدد را از ورودی بگیرد و عدد اول را بر عدد دوم تقسیم کند و نهایتا نتیجه را نمایش دهد.

در خط اول از تابع def استفاده کرده و نام آن را sub قرار دادیم سپس در پرانتز نام ورودی ها را نوشتیم و پرانتز بسته و دو نقطه

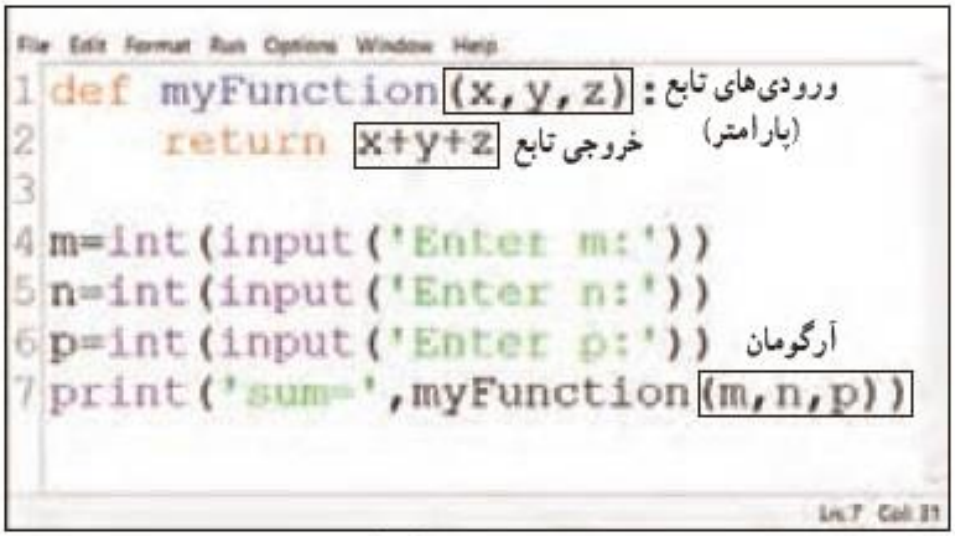
برای تعریف خروجی تابع از کلمه کلیدی return استفاده کردیم و در ادامه با کدهای input ورودی ها را دریافت می کنیم

در خط پنجم نام تابع را نوشته و به یک متغیر به نام result نسبت می دهیم و در پایان چاپ نتیجه تقسیم دو عدد:

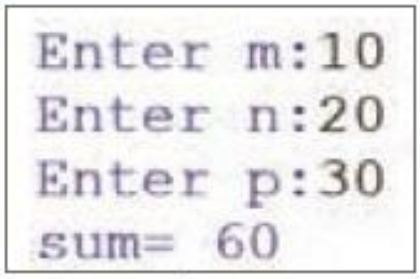
برای تمرین بیشتر مثال صفحه بعد را ببینید

مثال ۲۶

برنامه ای بنویسید که تابعی با سه ورودی ایجاد کند و حاصل جمع آن ها را با دستور return برگرداند.



روش تعریف تابع



خروجی برنامه

در قطعه برنامه بالا با استفاده از دستور def یک تابع در خط ۱ به نام myFunction ایجاد نمودیم و ورودی های تابع را با z و y و x مشخص کردیم و توسط دستور return در خط ۲ حاصل جمع آن ها را محاسبه کردیم و مقدار آن را بر گرداندیم. سپس در خطوط ۴ و ۵ و ۶ سه مقدار m , n , p را از ورودی دریافت و آنگاه در خط ۷ تابع را فراخوانی کردیم.

سپس با دستور print نتیجه را چاپ می کنیم.

چند نکته در مورد کار با تابع در پایتون

مثال ۲۷

تابعی بنویسید که دو عدد را دریافت کرده و عدد بزرگ تر را چاپ کند.

جواب

```
def maximum():
    a=int(input('Number 1 : '))
    b=int(input('Number 2 : '))
    print(max(a,b))
maximum()
```

همراهان گرامی برای آشنایی بیشتر با توابع در پایتون فیلم زیر را ببینید.

دانلود فیلم آشنایی با تابع در پایتون

ماژول در پایتون

ماژول (Module) به فایلی گفته می‌شود که حاوی دستورات و تعاریف پایتون است. به فایل پایتون با پسوند py. ماژول گفته می شود. داخل ماژول می توان تابع، لیست، تاپل، ست، دیکشنری و ... ایجاد کرد. ماژول نوشته شده را می توان ابتدا ذخیره و سپس در ابتدای هر برنامه ای وارد import کرد. و از عناصر داخل آن استفاده نمود. ماژول‌ها قابلیت استفاده مجدد از کد را فراهم می‌کنند. کاربران می‌توانند به جای کپی کردن کدها در برنامه‌های گوناگون، توابع پر استفاده خود را تعریف و وارد برنامه کنند

برای وارد کردن ماژول ها به برنامه از دو روش زیر استفاده می شود:

۱- دستور ..... Import

می‌توان یک ماژول را با استفاده از دستور import وارد کرد و به تعاریف درون آن با استفاده از عملگر dot(.) دسترسی داشت.در این روش هنگام استفاده از عناصر داخل ماژول، لازم است نام ماژول قبل از آن نوشته شود.

به مثال پایین توجه فرمایید

```
import math
print("The value of pi is", math.pi)
output: The value of pi is 3.141592653589793
```

در مثال بالا با استفاده از دستور import ماژول ریاضی math که یک ماژول داخلی پایتون است را فراخوانی کردیم و درخط دوم با دستور پرینت گفتیم از ماژول ریاضی فقط عدد پی را نمایش بده math.pi

۲- استفاده از دستور from ... import...

در این روش می‌توان نام‌های خاصی را از یک ماژول بدون وارد کردن کل ماژول، وارد کرد. در ادامه، مثالی از آنچه بیان شد آورده شده است.

```
from math import pi
print("The value of pi is", pi)
output: The value of pi is 3.141592653589793
```

با استفاده از قطعه کد بالا، فقط مشخصه pi از ماژول math فراخوانی شد. در چنین مثال‌هایی، از عملگر dot استفاده نمی‌شود.

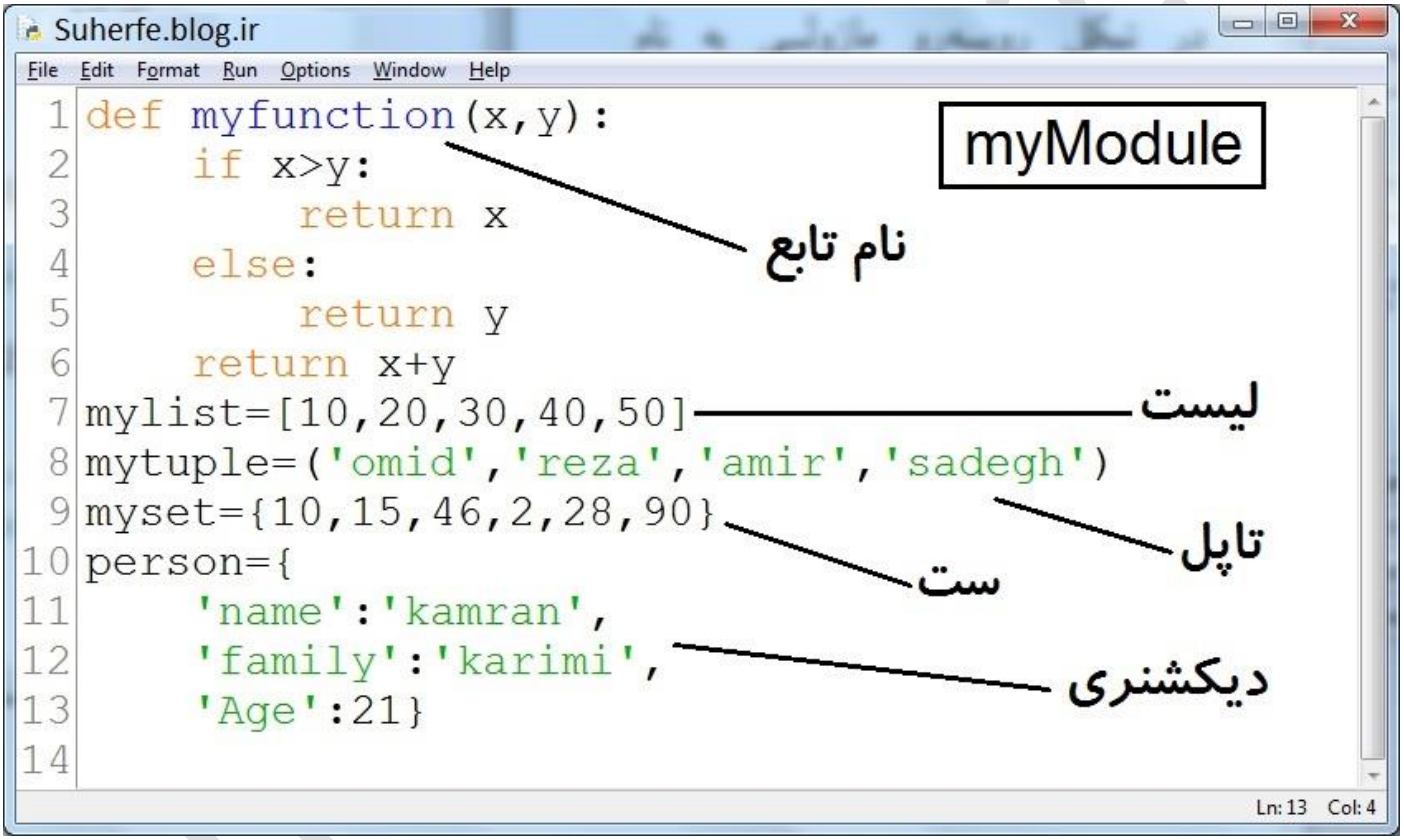
نکته : در روش دوم اگر بخواهیم همه عناصر داخل یک ماژول را فراخوانی کنیم لازم نیست اسم تک تک آنها را بنویسیم فقط کافیست بعد از دستور import از کاراکتر ستاره * استفاده شود منظور از ستاره یعنی می‌توان همه اسامی (تعاریف) را از یک ماژول وارد برنامه کرد.

```
from math import *
print("The value of pi is", pi)
output: The value of pi is 3.141592653589793
```

در صفحه بعد مثال دیگری را با هم بررسی می کنیم

مثال ۲۸

در شکل زیر ماژولی به نام myModule نوشته شده است که شامل تابع،لیست، تاپل، سِت، و دیکشنری است. این ماژول را به دو روش گفته شده در برنامه دیگری وارد و از عناصر داخل آن استفاده کنید.



جواب در صفحه بعد

ابتدا کدهای زیر را در محیط IDLE پایتون نوشته و آن را با نام دلخواهی مانند mymodule در یک پوشه ذخیره کنید .

```
def myfunction(x,y):
    if x>y:
        return x
    else:
        return y
    return x+y
mylist=[10,20,30,40,50]
mytuple=('omid','reza','amir','sadegh')
myset={10,15,46,2,28,90}
person={'name':'kamran','family':'karimi','Age':21}
```

الف) دستور ..... import

یک محیط برنامه نویسی جدید در پایتون باز کرده و کدهای زیر را با دستور import بنویسید.سپس برنامه را در همان پوشه با نام دلخواه ذخیره کنید حال آن را با زدن کلید f5 اجرا کنید.

```
import myModule
print(myModule.myfunction(10,20))
print(myModule.mylist[4])
print(myModule.mytuple[2])
print(myModule.myset)
print(myModule.person['family'])
```

در این روش هنگام استفاده از عناصر داخل ماژول، لازم است نام ماژول قبل از نام عناصر نوشته شود.

ب) دستور from … import…

محیط برنامه نویسی پایتون را باز کنید.

کدهای زیر را با دستور from … import… بنویسید.سپس برنامه را در همان پوشه با نام دلخواه ذخیره کنید حال آن را با زدن کلید f5 اجرا کنید.

```
from mymodule import *
print(myfunction(10,20))
print(mylist[4])
print(mytuple[2])
print(myset)
print(person['family'])
```

در روش دوم ، لازم نیست نام ماژول قبل از نام عناصر نوشته شود. و از عملگرdot استفاده نمی‌شود همچنین اگر بخواهیم همه عناصر داخل یک ماژول را فراخوانی کنیم لازم نیست اسم تک تک آنها را بنویسیم فقط کافیست بعد از دستور import از کاراکتر ستاره * استفاده شود منظور از ستاره یعنی می‌توان همه اسامی (تعاریف) را از یک ماژول وارد برنامه کرد.

خروجی: با اینکه از دو روش متفاوت استفاده کردیم اما خروجی به صورت یکسان و مانند جواب پایین می باشد.

```
20
50
amir
{2, 10, 46, 15, 90, 28}
karimi
```

نکته : افزون بر ماژول هایی که برنامه نویس می نویسد، دسته دیگری از ماژول های آماده نیز وجود دارند که از قبل در پایتون تعریف شده اند و می توان از آن ها استفاده کرد، مانند ماژول های turtle،numpy، pygame و...که برای کار با آرایه ها و اشکال هندسی و ... به کار می روند.

برای دیدن لیست ماژول های پایتون کد پایین را در پایتون تایپ و اجرا کنید پس از اجرا ده ها ماژول لیست خواهد شد.

کد لیست ماژول های پایتون

```
help('modules')
```

برای کد نویسی و شروع کار از منوی File در شل پایتون گزینه New File را انتخاب کنید تا پنجره جدیدی برای نوشتن برنامه ها باز شود حال در این پنجره شروع به نوشتن برنامه می کنیم.

بعد از نوشتن کدها از منوی Run گزینه Run Module را انتخاب کنید تا خروجی برنامه را ببینید.

مدیریت استثناء در پایتون

برنامه پایتون با برخورد به یک ارور متوقف می شود. در پایتون، ارورها می توانند از نوع خطاهای نحوی یا syntax error یا از نوع خطاهای منطقی یا exception باشند.

در ادامه با نوشتن یک کد هر دو خطا را بررسی می کنیم

خطاهای نحوی syntax زمانی رخ می دهد که مفسر پایتون یک دستور نادرست را تشخیص دهد. به مثال زیر توجه کنید:

```
a=4/0
print(a))
```

در این مثال در تابع پرینت از ۲ پرانتز استفاده کردیم. بدیهی است که پس از اجرا پایتون پیغام خطا را نمایش می دهد. پس لازم است به کدهای نوشته شده مراجعه و کد را اصلاح کنیم.

```
a=4/0
print(a)
ZeroDivisionError: division by zero
```

پس از اصلاح کد ، برنامه را دوباره اجرا می کنیم. اما این بار با یک خطای exception مواجه می شویم. این نوع خطا زمانی رخ می دهد که کد پایتون از نظر سینتکس syntax درست بوده اما به یک ارور منطقی برخورد می کند.

خط آخر پیام نشان می دهد که با چه نوع خطای استثنا مواجه شده اید. در این جا پایتون اعلام می کند که تقسیم عدد بر صفر بی معنی و غلط می باشد.

در چنین مواقعی برنامه نویس باید با استفاده از روش های مدیریت استثنا Exception Handling این مسئله را حل نماید.

مدیریت خطا در پایتون با try و except و finally

دستورهای try و except برای گرفتن و مدیریت خطا در پایتون استفاده میشود. اگر کد مشکوکی دارید که ممکن است استثنا ایجاد کند، می توانید با قرار دادن کد مشکوک در بلاک try از برنامه خود محافظت کنید. بعد از بلاک try یک بلاک except خواهد بود که مشخص میکند در زمان خطا چه اتفاقی بیفتد. بلاک های except مشخص میکنند که برنامه در زمان بروز خطا چطور واکنش دهد. میتوانید یک یا چند بلاک except برای یک try داشته باشید. دقت کنید که وجود حداقل یک بلاک except برای هر try اجباری است.

به مثال پایین توجه کنید.

مثال ۲۹

برنامه ای بنویسید که دو عدد صحیح از ورودی دریافت نماید و هنگام تقسیم دو عدد، حالت استثنا (تقسیم یک عدد بر صفر) را مدیریت کند.

جواب در صفحه بعد

```
num1=int(input('Enter number1:'))
num2=int(input('Enter number2:'))
try:
  print(num1/num2)
except:
  print('ZeroDivisionError')
else:
  print('Division')
finally:
  print('End')
```

توضیحات :

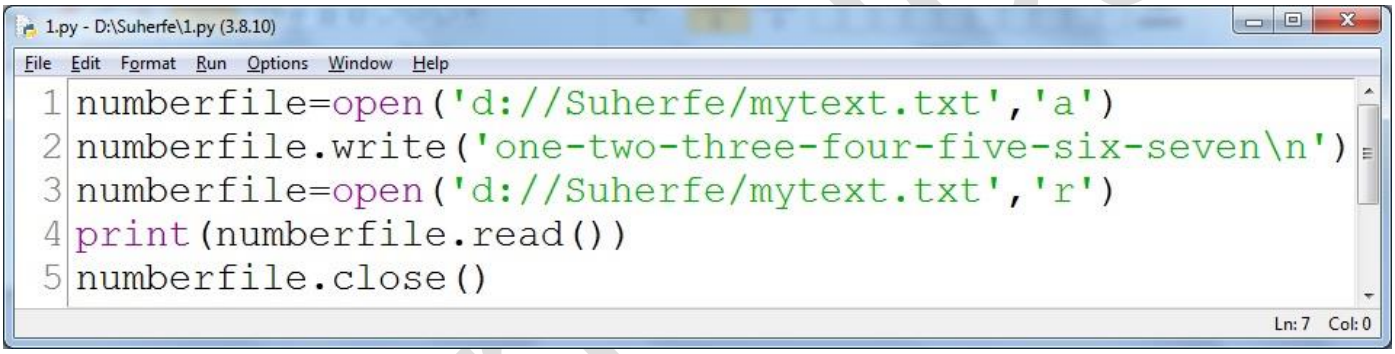
مفسر پایتون کدی که در بلاک try وجود دارد را به شکل کاملا عادی اجرا خواهد کرد. اگر این کد بدون خطا اجرا شود، بلاک except اجرا نخواهند شد. اما اگر کد try باعث بروز خطا شود بلاک except فعال خواهد شد. اگر بلاک except نتواند خطای ایجاد شده را مدیریت کند، در نهایت فارق از نوع خطا، بلاک else اجرا خواهد شد. دستور finally اختیاری بوده و همیشه اجرا خواهد شد، مهم نیست که خطا رخ داده باشد یا خیر. دقت کنید که برای هر بلاک try فقط میتوانید یک بلاک finally داشته باشید.

ایجاد فایل متنی و ویرایش آن در پایتون

برای ساخت فایل متنی می توان با برنامه Notpad، یک فایل txt ایجاد کرده، و آن را در رایانه ذخیره کرد شما می توانید همین کار را با پایتون هم انجام دهید.

ابتدا یک پوشه به نام suherfe در درایو D بسازید.

حال ویرایشگر پایتون را باز کرده و خطوط زیر را در آن بنویسید. و آن را در همان پوشه و درایو با نام دلخواه ذخیره کنید.



توضیح خطوط:

در خط ۱ پایتون، فایل متنی به نام mytext.txt را در فولدر suherfe واقع در درایوD می سازد. با کاراکتر 'a' می توانیم متنی را در فایل text اضافه کنیم. مسیر و آدرس فایل متنی را در جلوی متغیر دلخواهی به نام numberfile قرار می دهیم

numberfile=open('d://Suherfe/mytext.txt','a')

در خط ۲ از پایتون می خواهیم متن انگلیسی اعداد ۱ تا ۷ را در فایل متنی ما اضافه کند.

```
numberfile.write('one-two-three-four-five-six-seven\n')
```

منظور از کاراکترn\در انتهای رشته ، یعنی با هر بار اجرا، متن اعداد در خط جدیدی در فایل متنی text اضافه شود.

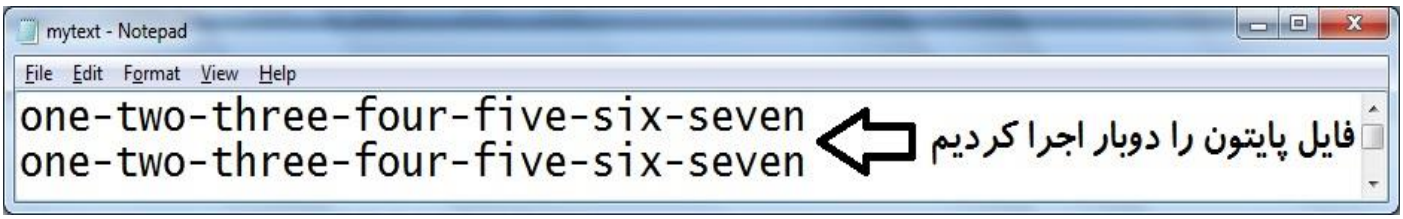
در خط سوم دستور باز شدن فایل را تایپ می کنیم

```
numberfile=open('d://Suherfe/mytext.txt','r')
print(numberfile.read())
```

برای خواندن r ابتدا فایل را باز و سپس آن را خوانده و چاپ می کنیم. و نهایتا دستور بستن فایل را تایپ می کنیم.

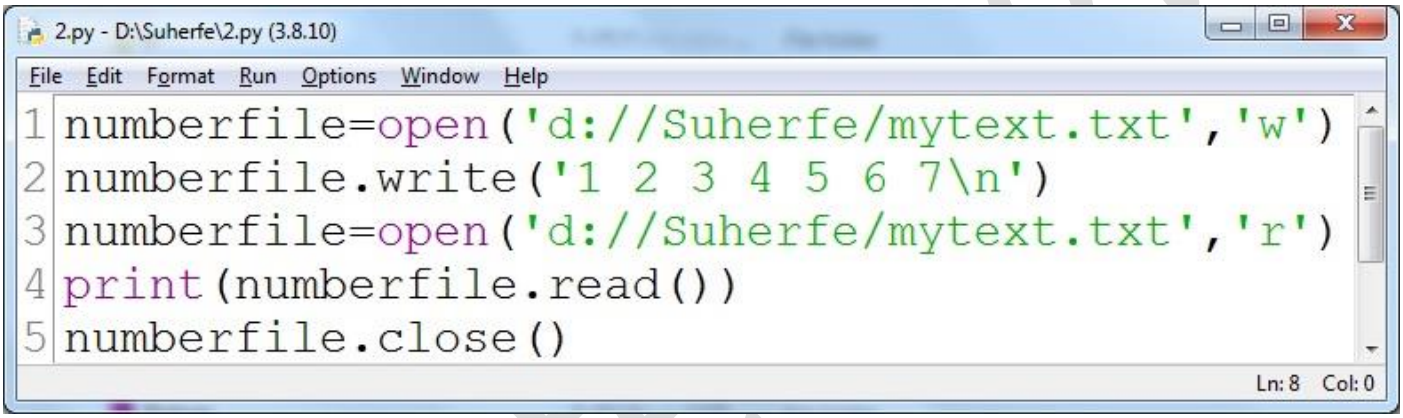
```
numberfile.close()
```

فایل پایتون را با کلید f5 اجرا کنید. خواهید دید که متن داخل فایل text نمایش داده می شود شما اگر چند بار فایل را اجرا کنید به همان اندازه متن تکرار و در فایل متنی ما درج می شود. حال فایل پایتون را بسته و فایل متنی text را باز کنید مشاهده می کنید که متن انگلیسی اعداد ۱ تا ۷ تایپ شده است.



نوشتن فایل متنی با کاراکتر 'w' در پایتون

در مثال قبل با کاراکتر "a" توانستیم یک رشته یا متنی را به فایل text اضافه کنیم و این کار را چندین بار می توانستیم تکرار کنیم. اما با کاراکتر w هم که به معنای write است نیز می توانیم در یک فایل متنی رشته ای را بنویسیم و آن را ذخیره کنیم. مانند تصویر پایین:



توضیح خطوط:

در خط ۱ پایتون، فایل متنی به نام mytext.txt را در فولدر suherfe واقع در درایوD می سازد. با کاراکتر 'w' می توانیم متنی را در فایل text بنویسیم. مسیر و آدرس فایل متنی را در جلوی متغیر دلخواهی به نامnumberfile قرار می دهیم

```
numberfile=open('d://Suherfe/mytext.txt','w')
```

در خط ۲ از پایتون می خواهیم اعداد ۱ تا ۷ را در فایل متنی ما بنویسد.

```
numberfile.write('1 2 3 4 5 6 7\n')
```

در خط سوم دستور باز شدن فایل را تایپ می کنیم

```
numberfile=open('d://Suherfe/mytext.txt','r')
print(numberfile.read())
```

برای خواندن r ابتدا فایل را باز و سپس آن را خوانده و چاپ می کنیم. و نهایتا دستور بستن فایل را تایپ می کنیم.

```
numberfile.close()
```

فایل پایتون را با کلید f5 اجرا کنید. خواهید دید که متن داخل فایل text نمایش داده می شود حال فایل پایتون را بسته و فایل متنی text را باز کنید مشاهده می کنید که تمام محتویات قبلی فایل text ما پاک و اعداد ۱ تا ۷ تایپ شده اند.

فرق کاراکتر "a" و"w" در کار با فایل در پایتون

با کاراکتر "a" به معنای append (اضافه کردن) می توانیم چندین بار یک متن یا رشته را تکرار و به فایل متنی اضافه کرد. در حالی که هنگام استفاده از کاراکتر "w" تمام محتویات قبلی فایل متنی پاک ورشته جدید در فایل text نوشته می شود.

پاک کردن فایل

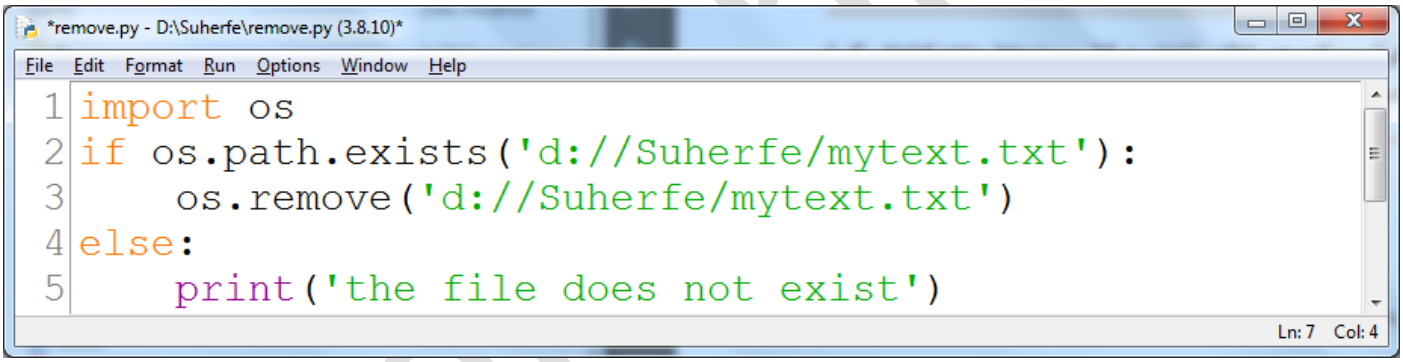
اگر بخواهیم فایل متنی text نوشته شده در صفحات قبلی را برای همیشه از حافظه رایانه پاک کنیم باید ماژول os را در ابتدای برنامه import کنیم.

توجه کنید که ماژول os جزو ماژول های استاندارد در پایتون است.

کدهای زیر را در پایتون نوشته و در همان مسیر فایل text ذخیره کنید حال با کلیک روی فایل پایتون فایل متنی برای همیشه از رایانه حذف می شود

توجه : پس از حذف، فایل به سطل بازیافت منتقل نمی شود.

طبق خط کدهای زیر می توانید فایل تکست را از رایانه حذف کنید



برای ساخت و ایجاد دوباره فایل text توضیحات صفحات قبلی، فایل های پایتونی که در پوشه نوشته و ذخیره کرده اید را با دوبار کلیک اجرا کنید.

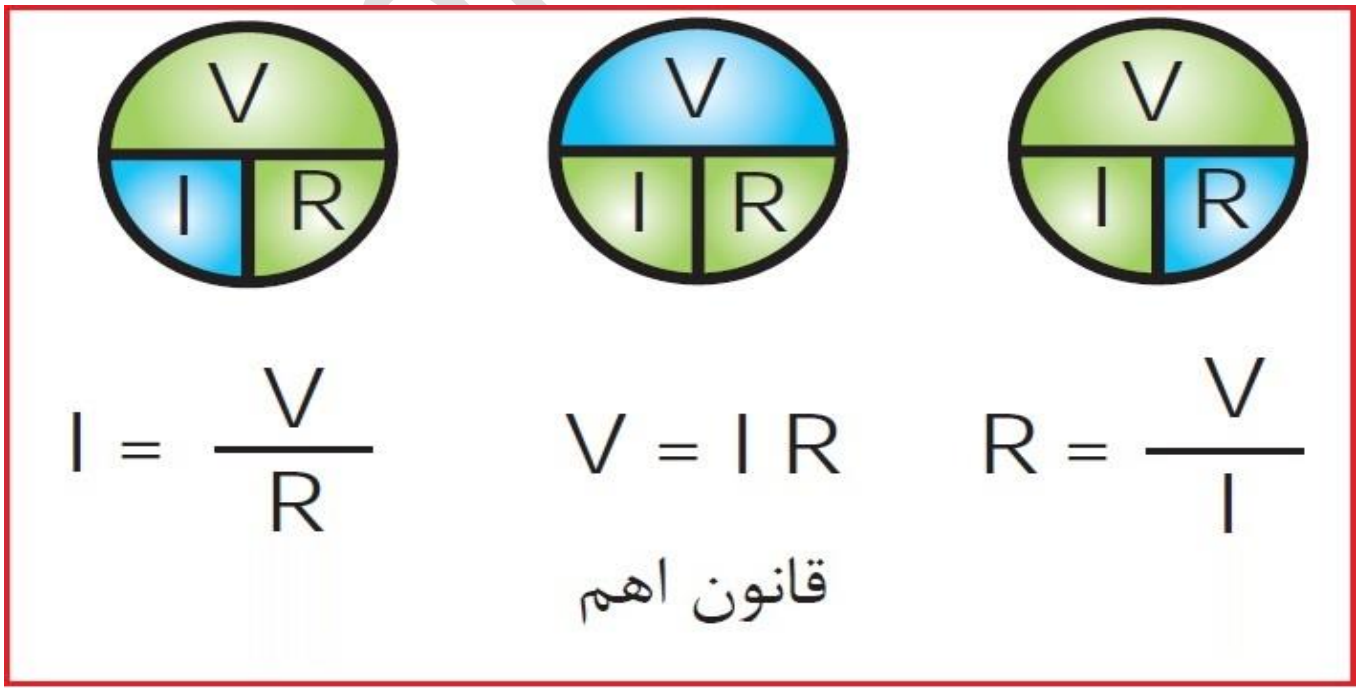
برای مرور و یادگیری بهتر در ادامه با طرح چند مثال دیگر در خدمت شما هستیم

مثال ۳۰

تابعی بنویسید که ولتاژ دو سر یک لامپ رشته ای و مقاومت آن را دریافت کرده و شدت جریانی که از آن عبور می کند را برگرداند. سپس آن را فراخوانی کنید.

جواب

```
def amper():
    V=float(input('voltage: '))
    R=float(input('resistance: '))
    I=V/R
    print('Amper =',I)
amper()
```



مثال ۳۱

با رابطه فیثاغورس، تابعی بنویسید که سه عدد را به عنوان اضلاع مثلث، دریافت کرده و تشخیص دهد که آیا با این سه ضلع می توان یک مثلث قائم الزاویه تشکیل داد یا خیر؟

جواب

```
def qaem():
    a=int(input('side length1= '))
    b=int(input('side length2= '))
    c=int(input('side length3= '))
    if a^2==b^2+c^2 or b^2==a^2+c^2 or
c^2==a^2+b^2:
        print('Right triangle')
    else:
        print('not Right triangle')
qaem()
```

مثال ۳۲

با استفاده از رابطه فشار، تابعی بنویسید که نیرو برحسب نیوتون و مساحت سطح برحسب مترمربع را دریافت کند و فشار را برحسب پاسکال برگرداند.

$$\text{فشار} = \frac{\text{نیرو}}{\text{سطح}} \quad \text{یا} \quad P = \frac{F}{A}$$

جواب

```
def Pressure():
    force=float(input('force: '))
    area=float(input('area: '))
    p=force/area
    print('Pressure =',p)
Pressure()
```

مثال ۳۳

تابعی بنویسید که معادله خطی به صورت y=2x+5 را شبیه سازی کند. سپس آن را با مقادیر x به ترتیب با ۵ و ۸ فراخوانی کنید.

جواب

```
def khat(x1,x2):
    y1=2*x1+5
    y2=2*x2+5
    print('x=5: y=',y1)
    print('x=8: y=',y2)
khat(x1=5,x2=8)
```


دانلود فیلم های آموزش پایتون

موفق باشید

لیست تولیدات سایت کاروفناوری کلاله

بسته طرح درس سالانه کاروفناوری

طرح درس روزانه کاروفناوری هفتم

طرح درس روزانه کاروفناوری هشتم

طرح درس روزانه کاروفناوری نهم

پاسخنامه کاروفناوری هفتم

پاسخنامه کاروفناوری هشتم

پاسخنامه کاروفناوری نهم

بسته پاورپوینت کاروفناوری هفتم

بسته پاورپوینت کاروفناوری هشتم

بسته پاورپوینت کاروفناوری نهم

پاورپوینت های آموزش مدارات برق

جزوه آموزش نرم افزار python

جزوه آموزش نرم افزار paint

جزوه آموزش نرم افزار Movie Maker

جزوه آموزش نرم افزار BTVC

جزوه آموزش نرم افزار Photo.Collage

جزوه آموزش نرم افزار AV.Audio.Editor

جزوه آموزش نرم افزار Toon.Boom

جزوه آموزش نرم افزار Edraw.Max

جزوه آموزش نرم افزار Bricscad

مطالب کتب کاروفناوری را در سایت کاروفناوری کلاله دنبال کنید.

Suherfe.blog.ir

پایان