
برنامه نویسی پایتون برای کودکان و نوجوانان به زبان ساده

تالیف:

مهندس علیرضا اسمعیلی
مهندس یوسف عباس نژاد ورزی



فن آوری نوین

سرشناسه	: اسمعیلی، علی رضا، ۱۳۵۱ -
عنوان و نام پدیدآور	: برنامه نویسی پایتون برای کودکان و نوجوانان به زبان ساده / تالیف علیرضا اسمعیلی، یوسف عباس نژادورزی.
مشخصات نشر	: بابل: فناوری نوین، ۱۳۹۹.
مشخصات ظاهری	: ۱۱۲ ص.: مصور (رنگی)، جدول.
شابک	: ۳۵۰۰۰۰ ریال : 9-01-7393-622-978
وضعیت فهرست نویسی	: فیبا
موضوع	: پایتون (زبان برنامه نویسی کامپیوتر)
موضوع	: Python (Computer program language)
شناسه افزوده	: عباس نژادورزی، یوسف، ۱۳۶۴ -
رده بندی کنگره	: QA۷۶/۷۳
رده بندی دیویی	: ۰۰۵/۱۳۳
شماره کتابشناسی ملی	: ۶۲۱۲۸۷۶



www.fanavarinovin.net

تلفن: ۰۱۱-۳۲۲۵۶۶۸۷

بابل، صندوق پستی ۷۳۴۴۸-۴۷۱۶۷

فن آوری نوین

برنامه نویسی پایتون برای کودکان و نوجوانان به زبان ساده

تألیف: مهندس علیرضا اسمعیلی، یوسف عباس نژادورزی

ناشر: فن آوری نوین

چاپ اول: بهار ۱۳۹۹

جلد: ۲۰۰

شابک: 978-622-7393-01-9

نام جایگاه و صحافی: دفتر فنی سورنا

طراح جلد: کانون آگهی و تبلیغات آبان (احمد فرجی)

قیمت: ۳۵۰۰۰ تومان

حروفچینی و صفحه آرایی: فن آوری نوین

تلفکس: ۶۶۴۰۰۱۴۴-۶۶۴۰۰۲۲۰

تهران، خ اردیبهشت، نبش وحید نظری، پلاک ۱۴۲

فهرست مطالب

فصل اول: آماده‌سازی محیط پایتون	۸
۱-۱. داتلود و نصب پایتون	۸
۱-۲. بررسی صحت نصب نرم‌افزار پایتون	۱۲
۱-۳. اجرای نرم‌افزار پایتون	۱۳
۱-۴. در این فصل چه چیزی یاد گرفتید	۱۴
فصل دوم: محاسبات در پایتون	۱۵
۲-۱. مقدمه	۱۵
۲-۲. محاسبات با پایتون	۱۵
۲-۳. عبارت محاسباتی	۱۶
۲-۴. عملگرها در پایتون	۱۶
۲-۵. اولویت عملگرها	۱۷
۲-۶. متغیرها	۱۸
۲-۷. چه چیزی یاد گرفتید؟	۲۱
فصل سوم: رشته‌ها، لیست‌ها، چندتایی‌ها و دیکشنری‌ها	۲۲
۳-۱. رشته یا string	۲۳
۳-۱-۱. ایجاد رشته	۲۳
۳-۱-۲. قرار دادن مقدار بین رشته (Embedding Values in Strings)	۲۴
۳-۱-۳. علامت ضرب در رشته	۲۵
۳-۱-۴. علامت + (جمع) با رشته	۲۵
۳-۲. لیست‌ها	۲۶
۳-۲-۱. اضافه کردن عنصر جدید به لیست	۲۷
۳-۲-۲. حذف عنصر از لیست	۲۸
۳-۲-۳. لیستی از اعداد و رشته	۲۸
۳-۲-۴. عملگر جمع در لیست	۲۹
۳-۲-۵. عملگر ضرب در لیست	۲۹
۳-۳. چندتایی	۳۰
۳-۴. دیکشنری‌ها (Maps)	۳۰
۳-۵. در این فصل چه چیزهایی یاد گرفتید	۳۱
فصل چهارم: ترمیم با لاک پشت	۳۲

- ۳۲..... ۴-۱. استفاده از مازول لاک پشت در پایتون
- ۳۳..... ۴-۲. ایجاد بوم نقاشی
- ۳۳..... ۴-۳. حرکت دادن لاک پشت
- ۳۶..... ۴-۴. رسم مربع
- ۳۶..... ۴-۵. پاک کردن صفحه
- ۳۷..... ۴-۶. نمایش لاک پشت به جای علامت پیکان «▶»
- ۳۸..... ۴-۷. کنترل سرعت حرکت لاک پشت
- ۳۸..... ۴-۸. رسم نقطه
- ۳۹..... ۴-۹. مخفی کردن لاک پشت
- ۳۹..... ۴-۱۰. رسم دایره
- ۳۹..... ۴-۱۱. رسم کمان
- ۴۰..... ۴-۱۲. رسم n ضلعی منظم
- ۴۰..... ۴-۱۳. تغییر رنگ زمینه
- ۴۰..... ۴-۱۴. تغییر رنگ قلم
- ۴۰..... ۴-۱۵. تغییر رنگ داخل شکل بسته
- ۴۱..... ۴-۱۶. کلونی از لاک پشت ها
- ۴۱..... ۴-۱۷. در این فصل چه چیزی یاد گرفتید
- ۴۲..... ۴-۱۸. تمرین ها

فصل پنجم: چگونه یک سؤال پیرسیم (ساختارهای تصمیم) ۴۵

- ۴۷..... ۵-۱. Do this or Else (این را انجام بده ... یا در غیر این صورت)
- ۴۸..... ۵-۲. دستور elif (Do THIS OR DO THIS OR ELSE!!!)
- ۴۹..... ۵-۳. ترکیب شرط ها
- ۵۰..... ۵-۴. تھی
- ۵۲..... ۵-۵. تفاوت چیست ...؟

فصل ششم: دوباره و دوباره (ساختارهای تکرار) ۵۵

- ۵۸..... ۶-۱. بلاک چیست و چه زمانی باید از آن استفاده کرد؟
- ۶۳..... ۶-۲. حلقه While
- ۶۵..... ۶-۳. چه چیزی یاد گرفتید؟
- ۶۵..... ۶-۴. تمرین

فصل هفتم: چیزهای بازیافتی برای استفاده مجدد (توابع) ۶۷

- ۷۰..... ۷-۱. Scope چیست؟
- ۷۳..... ۷-۲. استفاده از modules (ماژول ها)
- ۷۴..... ۷-۳. ماژول ها
- ۷۷..... ۷-۴. در این فصل چه چیزهایی یاد گرفتید.

فصل هشتم: برنامه نویسی در پایتون ۷۸

۷۸.....	۸-۱. تابع، ذخیره و اجرای برنامه در پایتون
۸۰.....	۸-۲. تابع print()
۸۲.....	۸-۳. خواندن داده
۹۰.....	۸-۴. چه چیزی یاد گرفتید؟
۹۰.....	۸-۵. تمرین‌ها.
۹۲.....	فصل نهم: مسائل حل شده با بسته Turtle
۱۱۱.....	منابع:
۱۱۳.....	پیوست الکترونیکی:

مقدمه

آموزش برنامه‌نویسی به کودکان و نوجوانان موضوع مهمی است که این روزها در جهان بسیار مورد توجه قرار گرفته است. یادگیری سریع‌تر در سنین پایین‌تر یکی از ویژگی‌های دوران کودکی و نوجوانی است. از این نظر آموزش مهارت‌های کلید زندگی به کودکان در این سنین اهمیت ویژه‌ای دارد. در این میان، آموزش برنامه‌نویسی به کودکان و نوجوانان، به‌عنوان یکی از مهم‌ترین مهارت‌های کلیدی قرن ۲۱، مورد توجه کشورهای پیشرفته‌ی جهان قرار گرفته است. نکته‌ای که وجود دارد این است که آموزش برنامه‌نویسی به کودکان تنها معطوف به آموزش یک مهارت ویژه برای زندگی در عصر فناوری‌های دیجیتال نیست، بلکه از نظر روانشناسی و علوم شناختی مزایای زیادی دارد. در حقیقت کارشناسان اعتقاد دارند که در دنیای معاصر، به دلایل مختلف، برنامه‌نویسی باید به کودکان و نوجوانان نیز آموزش داده شود که چند دلیل اصلی آن در زیر آمده است:

۱. افزایش خلاقیت، متخصصین اعتقاد دارند که برنامه‌نویسی، بیش‌تر بر پایه‌ی فنون خلاقیت است تا علوم کامپیوتری. در حقیقت، برنامه‌نویسان همانند هنرمندان، خالق و سازنده‌ی یک اثر هستند و نتیجه‌ی کار آن‌ها مرتبط با طرز بیان‌شان است. برنامه‌نویسی کودکان را مجاب به تجربه و آزمایش ایده‌های جدید می‌کند. پیترو نظرش را در این زمینه این‌گونه بیان می‌کند:

کد نویسی، کاری بسیار خلاقانه است. خلاقیت نه از بُعد هنرمندانه؛ بلکه از این بُعد که شما چیزی را خلق کرده‌اید که قبلاً وجود نداشته است. با استفاده از چندین ابزار، نرم‌افزاری را خلق کرده‌اید که توسط جامعه مورد استفاده قرار می‌گیرد و یا کاری را بهبود می‌بخشد.

۲. تکنیک حل مشکل، دانستن علم برنامه‌نویسی به کودکان دیدگاه یک مهندس نرم‌افزار را می‌دهد و به کودکان می‌آموزد که چگونه دید کامپیوتری داشته باشند و مشکلات را از آن منظر حل کنند. زبان برنامه‌نویسی بر پایه‌ی منطق، الگوریتم و ریاضیات است و به کودکان آموزش می‌دهد که برای حل مشکلات بزرگ، لازم است که آن‌ها را به بخش‌های کوچک‌تر تقسیم کنید تا آسان‌تر حل شوند. پیترو می‌گوید:

شما هیچ‌وقت یک برنامه را صرفاً برای ساخت یک برنامه نمی‌سازید. شما باید به مشکلی که می‌خواهید حل کنید توجه داشته باشید که آیا این مشکل برای مردم از اهمیت ویژه‌ای برخوردار است که بخواهند از برنامه شما استفاده کنند؟! و اگر این مشکل همگانی است، برای حل آن باید به دنبال بهترین راه‌حل باشید.

۳. درک بهتری از تکنولوژی، در عصر حاضر که تکنولوژی در حال بلعیدن همه چیز است، درک بهتر از آن مزیت بزرگی محسوب می‌شود. دانستن تکنولوژی (به‌طور مثال استفاده‌ی کاربردی از تلفن‌های هوشمند و برنامه‌ها) سبب عوض شدن نگرش انسان به این دنیا می‌شود. کد نویسی نه تنها به کودکان طریقه‌ی ساخت یک برنامه را شرح می‌دهد، بلکه شما را از سرعت بالای توسعه تکنولوژی آگاه می‌سازد و به کودکان می‌آموزد چرا یک برنامه

ساخته شده است؟ چرا سازندگان، این برنامه را ساخته‌اند و چرا دیگران این موقعیت را برای ساخت این برنامه درک نکرده‌اند؟ پیترو در این باره می‌گوید:

وقتی که علم کد نویسی را فرا می‌گیرید، دید جدیدی به دنیا خواهید داشت. زمانی که می‌فهمید چه کارهایی را می‌توان با برنامه‌نویسی انجام داد، درک بهتری از تکنولوژی که در تمام زندگی شما تأثیر می‌گذارد، خواهید داشت. بعد از آن می‌دانید که وقتی وارد صفحه‌ی فیس‌بوک می‌شوید، چه چیزی‌هایی در پشت پرده اتفاق می‌افتد.

۴. توانایی در گفتار تخصصی، یادگیری یک زبان برنامه‌نویسی برقراری ارتباط کودکان را با توسعه‌دهنده‌ها آسان‌تر می‌کند. همچنین یادگیری برنامه‌نویسی به درک و حل الگوریتم‌های پیچیده، ارتباط و انتقال مفاهیم پایه و اساسی کمک شایانی می‌کند. اهمیت این موضوع را پیترو این‌گونه بیان می‌کند:

اگر مدیران اجرایی بتوانند ایده‌های خود را به صورتی به تیم توسعه‌دهنده انتقال دهند که در حد و سطح دانایی (به زبان خودشان، زبان برنامه‌نویسی) آن‌ها باشد، توسعه‌دهندگان بهتر و سریع‌تر خواسته‌های مدیران را درک می‌کنند. همچنین مدیران با اطلاعات و زبان تخصصی آشنا می‌شوند و توسعه‌دهندگان هم بهتر می‌توانند به مدیران مراحل ساخت، طراحی و پیکره‌ی برنامه را به زبان تخصصی شرح دهند.

استیو جابر هم این‌گونه اهمیت برنامه‌نویسی را بیان می‌کرد:

تمام مردم باید برنامه‌نویسی یاد بگیرند؛ چون برنامه‌نویسی به شما یاد می‌دهد که چگونه فکر کنید

۵. تفکر خلاقانه در لحظات بحرانی، در حال حاضر، اکثر مدیران اجرایی در تلاش برای پیدا کردن راه‌حلی برای افزایش سوددهی و کاهش هزینه‌های جانبی هستند. باید این موضوع را در نظر گرفت که کد نویسی و تکنولوژی، تأثیر بسزایی در سازمان‌دهی و کنترل بخش‌های مختلف یک کسب‌وکار دارد و ورود آن، سبب می‌شود که فرد با دید حساس‌تری پروژه‌های جاری را دنبال کند و به دنبال راه‌حل‌های خلاقانه‌تری برای بهبود راندمان کاری باشد. پیترو این موضوع را به روشی ساده‌تر بیان می‌کند:

حسابدار شرکت ما، مجبور بود که برای نشان دادن گزارشات به مشتری‌ها، اطلاعات چندین سیستم را ثبت و ضبط کند و این اطلاعات را به روی کاغذ بیاورد و ارقام را با دست جابه‌جا کند. او همیشه به دنبال راهی آسان‌تر برای انجام این کار بود و بعد از آشنایی با ساخت نرم‌افزار، اهمیت برنامه‌نویسی برای او مشخص شد که چگونه در افزایش روند کارها به او کمک می‌کند. از تمامی اساتید و دانشجویان عزیز تقاضا داریم، هرگونه اشکال، ابهام در متن کتاب، پیشنهاد و انتقادات را به آدرس پست الکترونیک fanavarienovin@gmail.com ارسال نمایند.

در پایان امیدوارم این اثر مورد توجه جامعه انفورماتیک کشور، اساتید، دانشجویان و به‌ویژه کودکان و نوجوان عزیز قرار گیرد.

مولفین

fanavarienovin@gmail.com



آماده‌سازی محیط پایتون

۱-۱. دانلود و نصب پایتون

اولین قدم جهت برنامه‌نویسی، دانلود و نصب نرم‌افزار پایتون است. پایتون نسخه‌های مختلفی دارد که در حال حاضر (تألیف این کتاب) آخرین نسخه آن ۳,۸,۳ Python است. برای دانلود این نسخه پایتون مراحل زیر را انجام دهید:

۱. یکی از مرورگرهای Crome, Firefox یا IE را اجرا کرده، در نوار آدرس آن آدرس زیر را وارد کرده و کلید Enter را بزنید:

<https://www.python.org/downloads/>

۲. اکنون سایت دانلود نرم‌افزار پایتون باز می‌شود (شکل زیر):





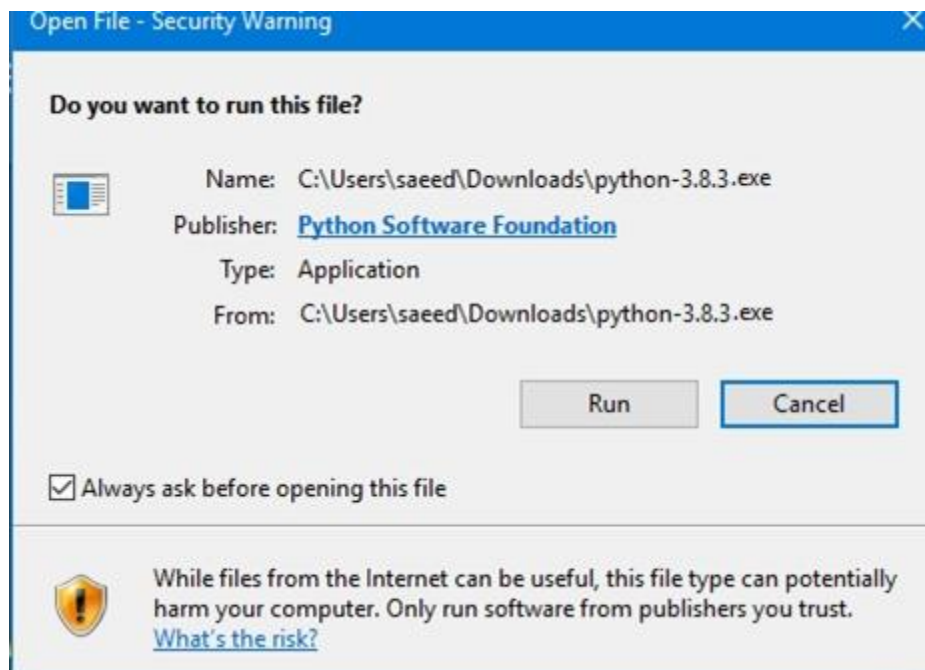
۳. دکمه **Download Python 3.8.3** را کلیک کنید تا شکل زیر ظاهر شود:



۴. دکمه **Save** را کلیک کنید و منتظر بمانید تا فایل نصبی پایتون دانلود شود. اکنون شکل زیر ظاهر می‌شود:



۵. دکمه **Run** را کلیک کنید. پس از چند لحظه ممکن است شکل زیر ظاهر شود:



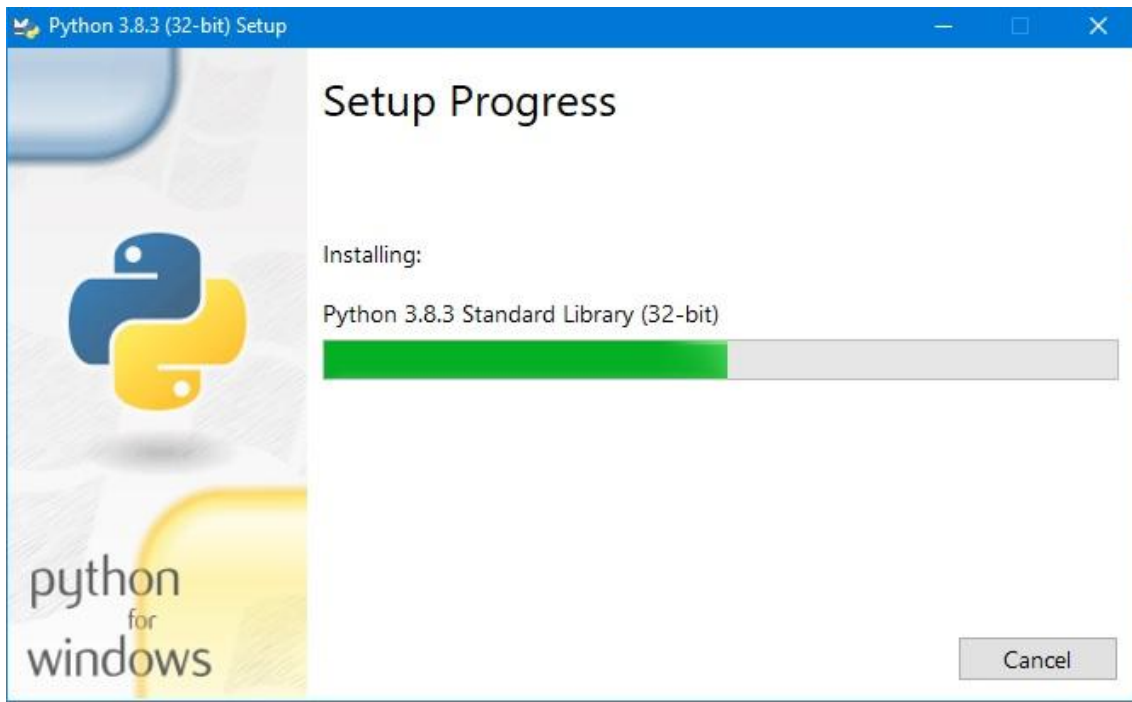
۶. دکمه **Run** را کلیک کنید تا برنامه پنجره نصب نرم‌افزار پایتون به شکل زیر ظاهر شود:



۷. گزینه `Add Python 3.8 to PATH` را انتخاب کنید (مانند شکل زیر):

Add Python 3.8 to PATH

۸. دکمه `Install Now` را کلیک کنید تا نصب نرم‌افزار پایتون شروع شود و شکل زیر ظاهر گردد:



۹. پس از چند لحظه نرم‌افزار پایتون نصب گردیده، شکل زیر ظاهر می‌شود:

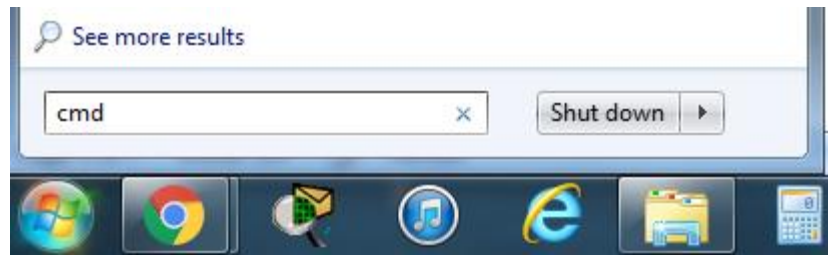


۱۰. دکمه Close را کلیک کنید.

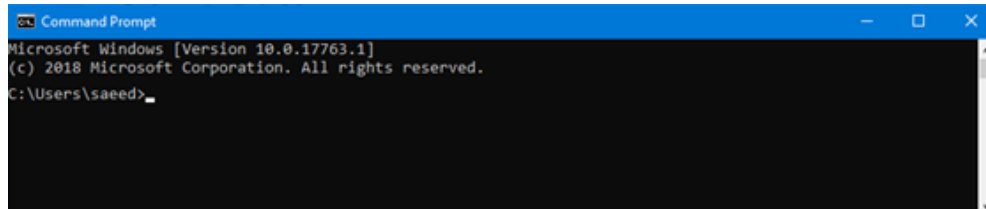
۲-۱. بررسی صحت نصب نرم‌افزار پایتون

در صورتی که نسخه ۳ پایتون را روی ویندوز نصب کرده باشید، برای اطمینان از نصب صحیح پایتون، مراحل زیر را انجام دهید:

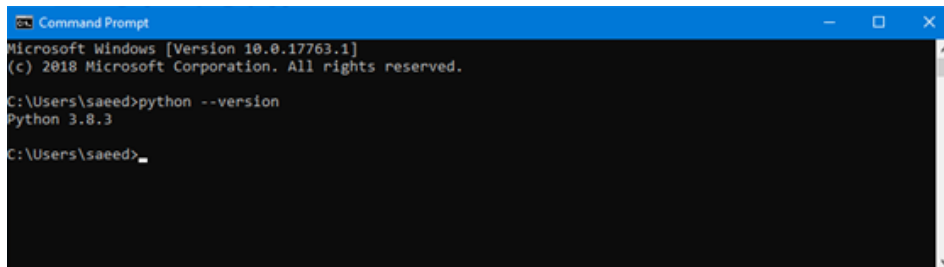
۱. در منوی Start کادر جست‌وجو ویندوز، کلمه `cmd` را تایپ کنید (مانند شکل زیر):



۲. اکنون بر روی فایل cmd.exe کلیک کنید تا شکل زیر ظاهر شود:



۳. دستور Python --version را اجرا کنید تا شکل زیر ظاهر شود:



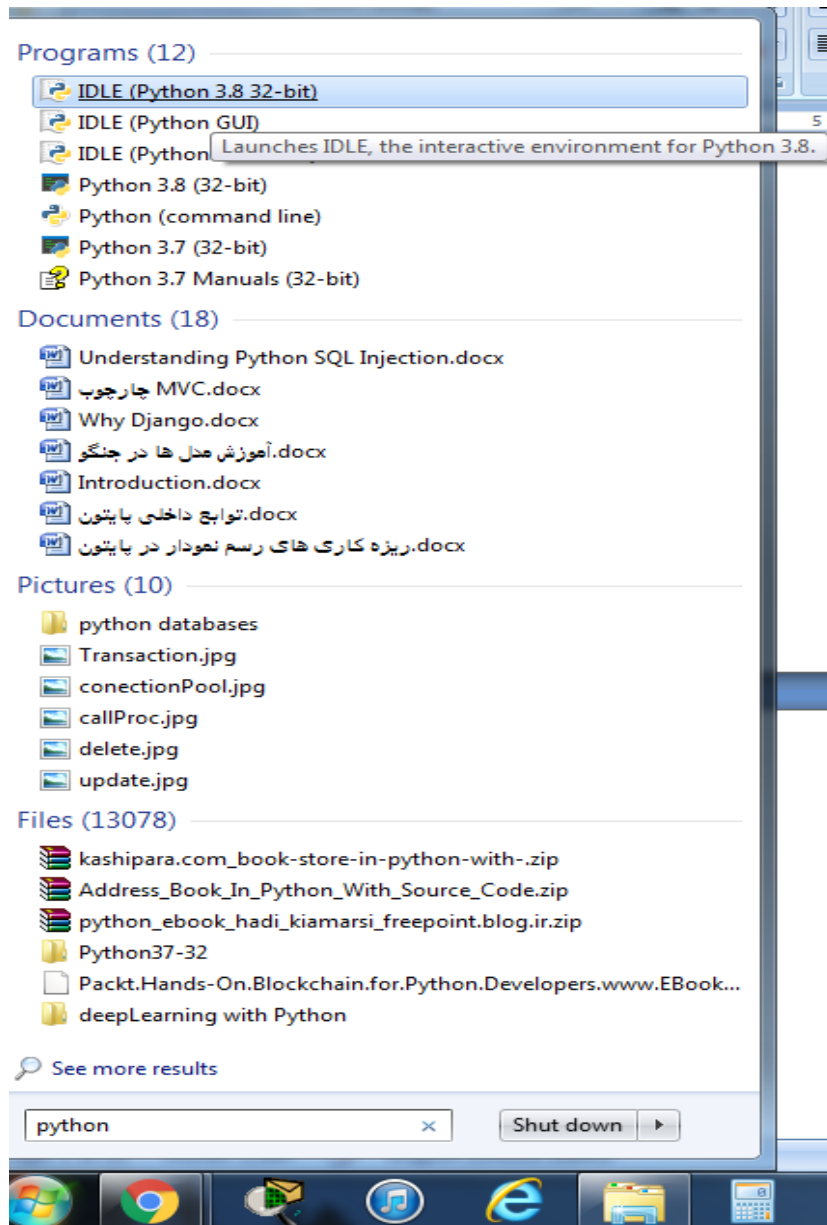
۴. پنجره Command Prompt را ببندید.

۳-۱. اجرای نرم‌افزار پایتون

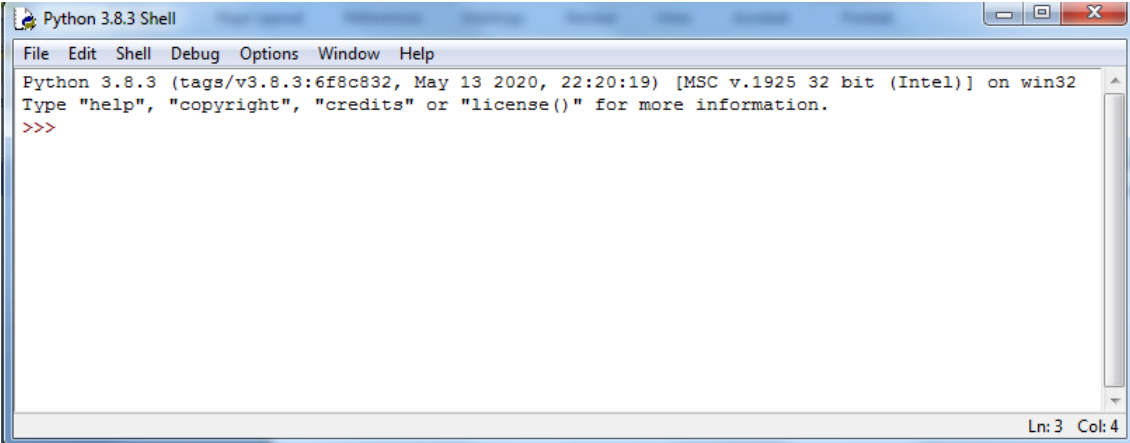
برای اجرای نرم‌افزار پایتون مراحل زیر را انجام دهید:

۱. در منوی Start، در باکس Search، عبارت Python را تایپ کرده تا لیست برنامه‌های

Python ظاهر شوند (شکل زیر):

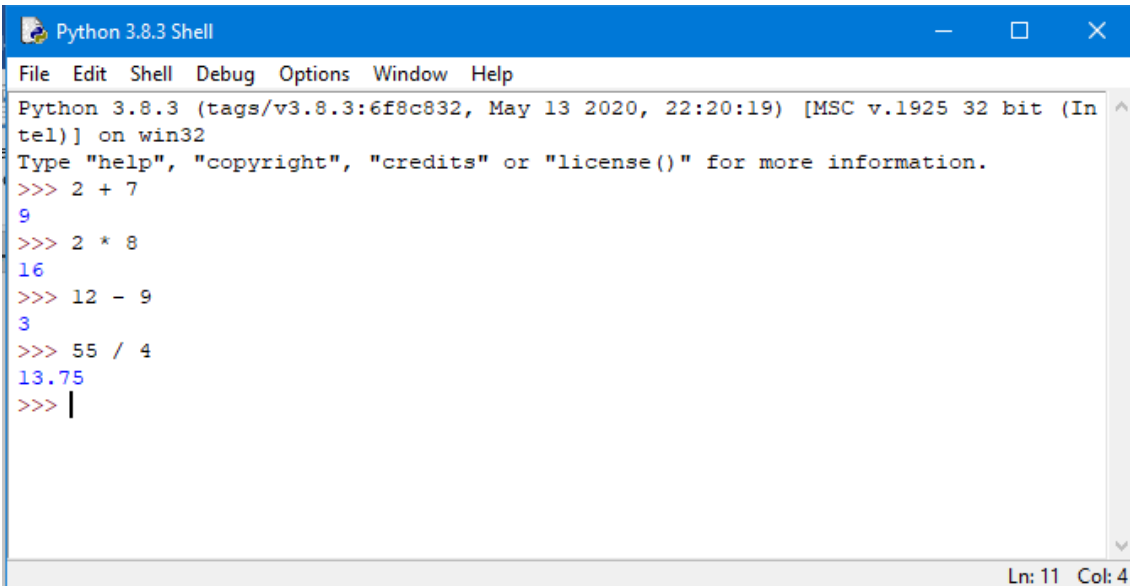


۲. برنامه (python ۳.۸ ۳۲-bit) IDLE را اجرا کنید تا شکل زیر ظاهر شود:



```
Python 3.8.3 Shell
File Edit Shell Debug Options Window Help
Python 3.8.3 (tags/v3.8.3:6f8c832, May 13 2020, 22:20:19) [MSC v.1925 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
```

۳. چند دستور را تایپ نمایید. پس از تایپ هر دستور کلید Enter را فشار دهید (مانند شکل زیر):



```
Python 3.8.3 Shell
File Edit Shell Debug Options Window Help
Python 3.8.3 (tags/v3.8.3:6f8c832, May 13 2020, 22:20:19) [MSC v.1925 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> 2 + 7
9
>>> 2 * 8
16
>>> 12 - 9
3
>>> 55 / 4
13.75
>>> |
```

۴-۱. در این فصل چه چیزی یاد گرفتید

در این فصل آموختید چگونه نرم‌افزار پایتون را دانلود کرده و نصب کنید. سپس بررسی نمایید که آیا نرم‌افزار پایتون به درستی نصب شده است یا خیر. در نهایت، اجرای نرم‌افزار پایتون و انجام چند دستور ساده را در آن آموختید.



۲

محاسبات در پایتون

۲-۱. مقدمه

بچه‌های عزیز بعد از نصب پایتون و در ابتدا با Python shell کار می‌کنیم تا بعضی از کارهای اولیه را با آن انجام دهیم. با مثال‌های ساده محاسباتی شروع می‌کنیم سپس با متغیرها آشنا می‌شویم. متغیرها یک روش برای ذخیره داده‌ها در برنامه‌های کامپیوتری هستند و به شما برای نوشتن یک برنامه مفید کمک می‌کنند.

۲-۲. محاسبات با پایتون

معمولاً وقتی می‌خواهید نتیجه ضرب دو عدد مانند $۶۸/۵ \times ۲۴$ را به دست بی‌آورید باید از ماشین حساب یا کاغذ و خودکار استفاده کنید. نظر شما چیه از Python shell برای انجام کارهای محاسباتی استفاده کنیم؟ خوبه که این روش را هم امتحان کنیم.

برای شروع Python shell، بر روی نشانه IDLE (Python 3.7 32-bit) در desktop کلیک

کنید. در مقابل >>> عبارت $۶۸/۵ \times ۲۴$ را

وارد نمائید. عدد 16440 را نمایش می‌دهد.

به صورت * 68.5

جالب بدانید به جای علامت ضرب (×) در برنامه‌نویسی از علامت (*) استفاده می‌شود و به جای علامت اعشار (/) از علامت (.) استفاده می‌گردد.

```
Python 3.8.3 Shell
File Edit Shell Debug Options Window Help
Python 3.8.3 (tags/v3.8.3:6f8c832, May 13 2020, 22:20:19) [MSC v.1925 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> 68.5 * 24
1644.0
>>>
```

۲-۳. عبارت محاسباتی

در برنامه‌نویسی هر معادله را عبارت محاسباتی می‌گویند که شامل عملگرها و عملوندها است. به‌عنوان مثال، در $A+5$ به علامت + عملگر و به A و ۵ عملوند می‌گویند.

۲-۴. عملگرها در پایتون

می‌توانید هر معادله‌ای که در آن عمل جمع، تفریق، ضرب و تقسیم، توان و باقی‌مانده تقسیم صحیح داشته باشد را در محیط پایتون انجام دهید. به هریک از این اعمال اصلی در برنامه‌نویسی اصطلاحاً عملگر محاسباتی گفته می‌شود. لیست عملگرهای محاسباتی را در جدول ۱-۲ مشاهده می‌کنید.

جدول ۱-۲: عملگرهای محاسباتی		
مثال	عملگر	نوع عمل
>>> 7 + 2 9	+	جمع
>>> 7 - 2 5	-	تفریق
>>> 7 * 2 14	*	ضرب
>>> 7 / 2 3.5	/	تقسیم اعشاری
>>> 7 \ 2 3	\	تقسیم صحیح
>>> 7 % 2 1	%	باقی‌مانده تقسیم
>>> 7 ** 2 49	**	توان

۵-۲. اولویت عملگرها

نکته مهمی که در برنامه‌نویسی باید در نظر داشته باشید، اولویت عملگرها است. یعنی، در یک عبارت محاسباتی که چندین عملگر وجود دارد کدام عملگر زودتر اجرا می‌شود؟ و چگونه می‌توان این اولویت را تغییر داد؟ جدول ۲-۲ اولویت عملگرها را مشخص می‌کند.

جدول ۲-۱: اولویت عملگرهای محاسباتی		
اولویت	عملگر	نوع عمل
اول	**	توان
دوم	*	ضرب
دوم	/	تقسیم اعشاری
دوم	\	تقسیم صحیح
دوم	%	باقی‌مانده تقسیم
سوم	+	جمع
سوم	-	تفریق

عبارت محاسباتی	>>> 6 + 3 * 4
نتیجه	18

در این مثال چون بیش‌ترین اولویت با عملگر * می‌باشد، پس ۳ در ۴ ضرب شده نتیجه ۱۲ می‌گردد، و سپس ۶ با ۱۲ جمع شده، نتیجه ۱۸ می‌شود.

از پرانتز برای تغییر اولویت عملگرها استفاده می‌شود. به مثال زیر توجه نمائید.

عبارت محاسباتی	>>> (6 + 3) * 4
نتیجه	36

در این مثال، چون از پرانتز استفاده شده بیش‌ترین اولویت با عبارت داخل پرانتز و عملگر جمع می‌باشد، پس ۶ با ۳ جمع شده، نتیجه ۹ می‌شود، و سپس ۹ در ۴ ضرب شده، نتیجه ۳۶ می‌شود.

بعضی از عملگرها اولویت یکسان دارند. در این حالت، بیش‌ترین اولویت با عملگری است که در سمت چپ قرار دارد.

عبارت محاسباتی	>>> 6 / 3 * 4
نتیجه	8

در این مثال ضرب و تقسیم دارای اولویت یکسانی هستند. در این حالت، بیش‌ترین اولویت با عملگری است که در سمت چپ قرار دارد. بنابراین، ۶ تقسیم بر ۳ شده نتیجه ۲ می‌شود. سپس ۲ در ۴ ضرب شده ۸ می‌شود.

عبارت محاسباتی	>>>6 + 3 * 4 ** 2
نتیجه	54

در این مثال ، بیشترین اولویت با عملگر ****** یا «توان» است. بنابراین، ۴ به توان ۲ شده نتیجه ۱۶ می‌شود. سپس ۱۶ در ۳ ضرب شده ۴۸ می‌شود و در نهایت ۴۸ با ۶ جمع شده نتیجه ۵۴ به دست می‌آید. اولویت عملگرها در این مثال به ترتیب توان ، ضرب و جمع می‌باشد.

مثال زیر را در نظر بگیرید اعداد و عملگرها مشابه هم، اما در بخش "ب" با استفاده از پرانتز اولویت عملگرها را تغییر داده و نتیجه خروجی نیز متفاوت است:

الف	ب
$\ggg 100 - 2 * 30 + 10$ <p>اولویت اول * اولویت دوم - اولویت سوم + نتیجه</p>	$\ggg 100 - 2 * (30 + 10)$ <p>اولویت اول داخل () + ضرب * تفریق - نتیجه</p>
$100 - 2 * 30 + 10$ $100 - 60 + 10$ $40 + 10$ 50	$100 - 2 * (30 + 10)$ $100 - 2 * 40$ $100 - 80$ 20

۶-۲. متغیرها

متغیر در برنامه‌نویسی، محلی برای ذخیره داده و اطلاعات می‌باشند. داده‌ها مانند اعداد، متن یا لیستی از اعداد، متن و غیره می‌باشد.

شما می‌توانید برای ایجاد یک متغیر به نام **ali**، از یک علامت (=) استفاده کنید و سپس به پایتون بگویید چه اطلاعاتی باید متغیر داشته باشد؛ به‌عنوان مثال، پول علی ۱۰۰۰ تومان است.

```
>>> ali = 1000
```

بعد از مقدار دادن به یک متغیر باید امکان نمایش محتویات متغیر نیز وجود داشته باشد. برای نمایش محتویات یک متغیر از دستور **print** استفاده نمایید. مانند دستور زیر:

```
>>> print (ali)
1000
```

به راحتی می‌توانید به پایتون بگویید محتویات متغیر **ali** را تغییر دهد. برای مثال، محتویات متغیر **ali** را به ۲۰۰۰ تغییر می‌دهیم. یعنی پول علی به ۲۰۰۰ تومان تغییر کرد.

```
>>> ali = 2000
>>> print (ali)
2000
```

در خط اول، به پایتون گفتیم محتویات `ali` برابر با ۲۰۰۰ است، و در خط دوم، خواستیم محتویات متغیر `ali` را نمایش دهد و در خط سوم، پایتون ۲۰۰۰ را نمایش داد.

از متغیرهای بیش‌تری می‌توانید در برنامه استفاده کنید. برای مثال، پول `reza` برابر با پول `ali` است. پول `reza` چقدر است؟

```
>>> ali = 2000
>>> reza = ali
>>> print (reza)
2000
```

در این مثال، در خط دوم، به پایتون گفتیم پول رضا برابر با پول علی شود و در خط سوم، از او خواستیم محتویات متغیر `reza` را نمایش دهد که همان ۲۰۰۰ را در خط چهارم نمایش داد.

البته، `ali` احتمالاً یک نام بسیار مفید برای یک متغیر نیست. زیرا به احتمال زیاد به ما چیزی درباره متغیر استفاده‌شده نمی‌گوید. بهتر است به جای متغیر `ali` از یک نام بهتری استفاده کنیم که نشان‌دهنده محتویات متغیر باشد. به عنوان مثال، `Pocket_money` «پول توجیبی»

```
>>> Pocket_money = 2000
>>> print (Pocket_money)
2000
```

مشخص است این برنامه در مورد ۲۰۰۰ تومان پول توجیبی صحبت می‌کند. چون نام متغیر مناسب انتخاب شده است.

نام متغیر می‌تواند از حروف، اعداد و زیرخط (`_`) باشد، ولی نمی‌تواند با یک عدد شروع شوند. برای نام متغیر می‌توانید از حروف تک مانند `a` یا یک جمله طولانی استفاده کنید. ولی بین کلمات نباید فاصله باشد.

به نظر شما، نام کدام متغیر درست به کار گرفته شده است؟

نام متغیر
P
5p
Pm
Pocket money
Pocket_money
PocketMoney

جواب‌هایی که دادید را با پاسخ‌های جدول زیر مقایسه نمایید:

نام متغیر	جواب	توضیحات
P	درست	نام متغیر می‌تواند تک‌حرفی باشد.
5p	نادرست	نام متغیر نمی‌تواند با رقم شروع شود.
Pm	درست	نام متغیر می‌تواند بیش از یک حرف باشد.
Pocket money	نادرست	بین کلمات در نام متغیر نمی‌تواند فاصله باشد.

Pocket_money	درست	بین کلمات در نام متغیر می‌تواند علامت زیر خط «_» باشد.
PocketMoney	درست	اگر نام متغیر شامل چند کلمه است بهتر است حرف اول آن‌ها بزرگ نوشته شود.

علی و پول توجیبی‌اش را به خاطر بی‌آوردی او هر سال عید ۴۰۰۰۰۰ تومان از خانواده‌اش عیدی می‌گیرد قصد دارد تا سال آینده پول‌های خود را پس‌انداز کند. اگر او هر روز ۲۰۰۰ تومان از پول توجیبی خود را پس‌انداز کند و روزهای جمعه ۷۰۰ تومان از پس‌انداز خود بردارد در طی یک سال چقدر می‌تواند پس‌انداز کند؟ «در ضمن علی با نگاه به تقویم متوجه شد هر سال ۳۶۵ روز و ۵۲ هفته می‌باشد پس ۵۲ روز آن جمعه است.»

علی این محاسبات را به صورت زیر انجام داد.

```
>>> 400000 + 2000 * 365
1130000
>>> 700 * 52
36400
>>> 1130000 - 36400
1093600
```

علی می‌تواند تمامی محاسبات را در یک خط به صورت زیر بنویسد:

```
>>> 400000 + 2000 * 365 - 700 * 52
1093600
```

بچه‌های خوب، چطور می‌توانید از متغیرها برای نگه‌داری این اعداد استفاده کنید؟ برای مثال قبلی، متغیرهای زیر پیشنهاد می‌شود.

نام متغیر مناسب	نام متغیر ساده	معادل فارسی متغیر	مقدار
Rewarded	r	عیدی	۴۰۰۰۰۰ تومان
PocketMoney	p		
DaySaving	d	پس‌انداز روزانه	۲۰۰۰ تومان
YearSaving	y	پس‌انداز در طی سال	باید حساب شود
getMySaving	g	برداشت از پس‌انداز در روزهای جمعه	۷۰۰ تومان

چون شروع کارتان است برای آن‌که درگیر واژه‌های انگلیسی نشوید، حرف اول هر یک از این کلمات را نام متغیر در نظر بگیرید و به این متغیرها مقدار دهید.

```
>>> r = 400000
>>> d = 2000
>>> g = 700
```

حالا می‌توانید معادله را به صورت زیر مجدداً وارد کنید:

```
>>> r + d * 365 - g * 52
```

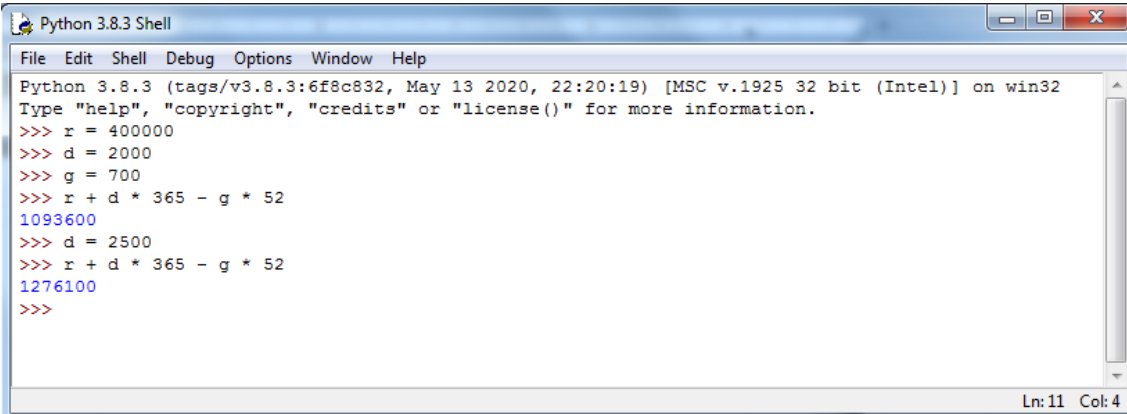
1093600

می‌توانید ببینید پاسخ یکسانی به دست می‌آید.

مهم‌ترین مزیت متغیرها، تغییر محتویات با مقادیر جدید است. مثلاً علی با دیدن نتیجه به دست آمده می‌گوید: «اگر روزی ۲۵۰۰ پس‌انداز کنم، نتیجه چه تغییری می‌کند.» کافی است مقدار متغیر **d** که پس‌انداز روزانه است را تغییر دهید.

```
>>> d = 2500
```

سپس معادله را مجدداً بنویسید (یا با **copy** و **paste** کردن معادله به راحتی می‌توانید با مقدار جدید نتیجه را مجدداً محاسبه کنید. در ضمن حتی نیاز به نوشتن معادله نیست کافی است مکان نما را در ابتدا معادله قرار داده، کلید **Enter** را فشار دهید).



```
Python 3.8.3 Shell
File Edit Shell Debug Options Window Help
Python 3.8.3 (tags/v3.8.3:6f8c832, May 13 2020, 22:20:19) [MSC v.1925 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> r = 400000
>>> d = 2000
>>> g = 700
>>> r + d * 365 - g * 52
1093600
>>> d = 2500
>>> r + d * 365 - g * 52
1276100
>>>
```

علی با مشاهده نتیجه جدید به فکر فرو رفت. شما چطور؟

۷-۲. چه چیزی یا گرفتید؟

در این فصل استفاده از عملگرهای محاسباتی، اولویت آن‌ها و استفاده از پرانتز برای تغییر اولویت عملگرها را یاد گرفتید. همچنین ایجاد متغیر و استفاده از آن‌ها در عبارتهای محاسباتی آموزش داده شد.



در فصل ۲ عملیات محاسباتی و کار با متغیرها را در پایتون یاد گرفتید. در این فصل با دیگر عناصر پایتون در برنامه‌نویسی مانند رشته، لیست و غیره آشنا می‌شوید.

۱-۳ رشته یا string

در برنامه‌نویسی معمولاً متن یا پیغامی نمایش داده می‌شود که به این متن یا پیغام رشته گفته می‌شود. بنابراین، مجموعه‌ای از حرف، علامت یا رقم تشکیل رشته را می‌دهند. مانند نام شما، آدرس منزل شما، یا دربازی‌های رایانه‌ای پیغام "Game Over".

۱-۱-۳ ایجاد رشته

برای ایجاد رشته کافی است عبارت رشته‌ای را در داخل " " یا ' ' قرار دهید. برای نمایش محتویات متغیر از دستور print استفاده می‌شود.

```
>>> msg = "Game Over"
>>> print(msg)
Game Over
```

در مثال زیر نام Ali ابتدا در یک متغیر قرار می‌گیرد، سپس نمایش داده می‌شود:

```
>>> firstName = 'Ali'
>>> print(firstName)
Ali
```

اگر یک عبارت رشته‌ای با علامت (") یا (') آغاز شود، با همان علامت نیز باید خاتمه یابد. در غیر این صورت، خطا نشان داده می‌شود.

به مثال‌های زیر توجه نمایید که خطای املایی (Syntax Error) یا خطا و علت آن را نشان می‌دهد.

```
Python 3.8.3 Shell
File Edit Shell Debug Options Window Help
Python 3.8.3 (tags/v3.8.3:6f8c832, May 13 2020, 22:20:19) [MSC v.1925 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> msg = "Game over
SyntaxError: EOL while scanning string literal
>>> msg = "Game over'
SyntaxError: EOL while scanning string literal
>>> msg = 'Game over"
SyntaxError: EOL while scanning string literal
>>> msg = Game over"
SyntaxError: invalid syntax
>>>
```

اگر بخواهید رشته‌ای را در چند خط بنویسید از سه تا علامت تک کوتیشن (''') استفاده کنید.

```
>>> msg = '''what is your name?
My name is AliReza.'''
```

سؤال: اگر خود رشته شامل (") و (') باشد، چه کاری انجام دهیم؟
در هریک از حالت‌های زیر خطا رخ می‌دهد:

```
>>> msg1 = 'He said, "Aren't can't shouldn't wouldn't." '
SyntaxError: invalid syntax
```

```
>>> msg2 = "He said, "Aren't can't shouldn't wouldn't." "
SyntaxError: invalid syntax
```

روش اول: از سه تا علامت تک کوتیشن ("") استفاده کنید.

```
>>> msg1 = ''' He said, "Aren't can't shouldn't wouldn't." '''
```

روش دوم: قبل از علامت (") و (') در داخل رشته از علامت \ استفاده نمایید. مانند: (\') و (\")

```
>>> msg1 = 'He said, "Aren\'t can\'t shouldn\'t wouldn\'t.'" '
>>> msg2 = "He said, \"Aren't can't shouldn't wouldn't.\" "
>>> print(msg1)
He said, "Aren't can't shouldn't wouldn't."
>>> print(msg2)
He said, "Aren't can't shouldn't wouldn't."
```

۲-۱-۳. قرار دادن مقدار بین رشته (Embedding Values in Strings)

اگر می‌خواهید مقادیر ثابت یا محتویات متغیر داخل رشته نشان داده شود، از %S درون رشته استفاده نمایید تا محل درج مقدار یا محتویات متغیر مشخص گردد.

در مثال زیر در پیغام "I scored points" جای نقطه‌چین می‌خواهیم امتیاز ۲۰۰ را قرار دهیم، باید در هنگام کد نویسی به جای نقطه‌چین از %S استفاده گردد. به آن جا نگهدار (placeholder) می‌گویند.

```
>>> msg = 'I scored %s points'
>>> print(msg % 200)
I scored 200 points
```

همان طوری که مشاهده می‌شود عدد ۲۰۰ جایگزین %S در خروجی چاپ شد.

کد زیر همان مثال قبلی است، اما به جای عدد از متغیر استفاده شده است:

```
>>> myScore = 200
>>> msg = 'I scored %s points'
>>> print(msg % myScore)
I scored 200 points
```

همان طوری که مشاهده می‌شود محتویات متغیر myScore که عدد ۲۰۰ است جایگزین %S در خروجی شد.

در یک رشته می‌توان از چند جا نگهدار استفاده کرد و مقادیر آن‌ها را در داخل پرانتز قرار داد. به همان ترتیب به جای علامت درصد قرار می‌گیرند.

```
>>> myName="Ali"
>>> myScore = 200
>>> msg = 'The %s scored %s points'
>>> print(msg %(myName, myScore))
The Ali scored 200 points
```

همان طوری که مشاهده می‌شود محتویات متغیر myName یا "Ali" به جای اولین %S و محتویات myScore که عدد ۲۰۰ است جایگزین %S دوم در خروجی چاپ شد.