



هر برنامه کاربردی که ایجاد می‌کنید، با کامپیوتر و داده‌هایی که درون کامپیوتر دارید در تعامل است. برنامه‌ها از آن جهت روی داده‌ها متمرکز هستند که بدون داده‌ها، هیچ دلیل موجهی برای ساخت یک برنامه وجود ندارد. هر برنامه‌ای که از آن استفاده می‌کنید حتی یک بازی ساده مثل شطرنج داده‌ها را به روش خاصی دستکاری می‌کند. اما در این بخش می‌خواهیم با محیط خط فرمان پایتون که قابلیت‌های بیشتری در اختیار ما قرار می‌دهد بیشتر آشنا شویم.

برای مطالعه بخش دوم آموزش رایگان پایتون [اینجا](#) کلیک کنید

بیشتر برنامه‌های کاربردی با رویکرد ساخت، خواندن، به‌روزرسانی و حذف (CRUD) سرنام Create, Read, Update Delete ساخته می‌شوند. اما پیش از آنکه برنامه شما به کامپیوترتان دسترسی پیدا کنید، شما باید از طریق یک زبان برنامه‌نویسی به تعامل با کامپیوتر پرداخته و فهرستی از وظایفی که کامپیوتر باید انجام دهد را بر مبنای یک زبان قابل فهم برای کامپیوتر ایجاد کنید. **پایتون** یکی از زبان‌های برنامه‌نویسی محبوبی است که برای انجام چنین کاری ایده‌آل است.

از فایل README سابق و NEWS جدید غافل نشوید

بیشتر برنامه‌های کاربردی همراه با فایل به نام README عرضه می‌شوند. در حالت کلی، این فایل شامل اطلاعات به‌روز شده‌ای است که قبل از آنکه برنامه در یک وضعیت پایدار همراه با مستندات آن منتشر شود در اختیار کاربران قرار می‌گیرد. متأسفانه اکثر مردم از فایل README چشم‌پوشی کرده و برخی حتی نمی‌دانند چنین فایل وجود دارد. این افراد هیچ‌گاه متوجه نخواهند شد در ویرایش جدید یک محصول چه ویژگی‌های شاخصی ممکن است وجود داشته باشد. **پایتون** یک فایل README دارد که به شکل متنی درون پوشه اصلی Python قرار دارد. اگر از نسخه‌های جدید **پایتون** استفاده می‌کنید این فایل به نام NEWS.txt درون این پوشه قرار دارد. در این فایل موارد زیر قرار دارد:

- اطلاعاتی در ارتباط با ویژگی‌های جدید اضافه شده به **پایتون**
- نحوه نصب **پایتون** روی سیستم‌های لینوکسی
- نحوه دریافت جدیدترین نسخه از مستندات **پایتون**
- نحوه تبدیل پروژه‌های قدیمی **پایتون** به نسخه x.3.3

- نحوه نصب نسخه‌های مختلف **پایتون** روی یک سیستم
- نحوه دنبال کردن باگ‌ها و بررسی مشکلات
- نحوه به‌روزرسانی **پایتون**
- نحوه آگاه شدن از عرضه نسخه بعدی **پایتون**
- نحوه پیاده‌سازی یک آزمایش سفارشی **پایتون** و... از جمله این موارد است. مطمئن باشید با خواندن این فایل اطلاعات ارزشمندی به دست خواهید آورد که همگان را شگفت‌زده خواهید کرد.

باز کردن و اجرای محیط خط فرمان

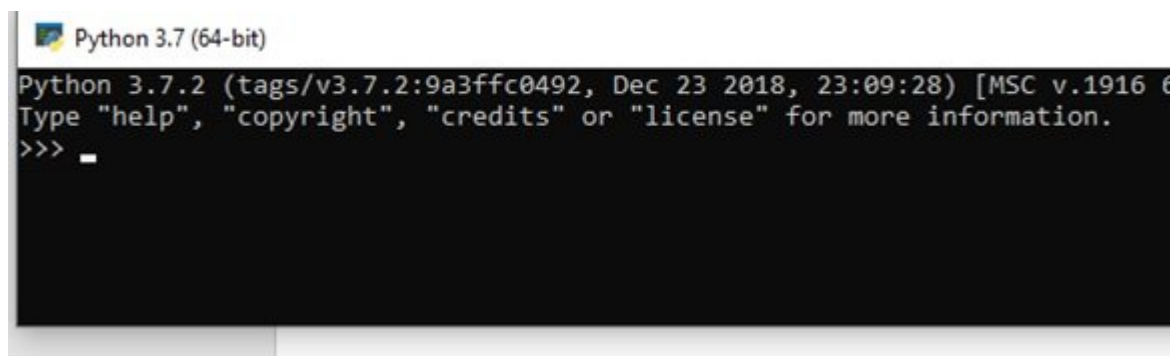
پایتون به شما اجازه می‌دهد بر مبنای راهکارهای مختلفی با این زبان در تعامل باشید. در شماره قبل به شکل مختصر با محیط توسعه یکپارچه گرافیکی **پایتون** کار کردید. IDLE برای طراحی و توسعه برنامه‌های کاربردی ابزارهای کاملی در اختیارتان قرار می‌دهد، با این وجود، گاهی اوقات شما می‌خواهید تجربه بیشتری به دست آورده یا برنامه‌هایی را اجرا کنید که ورودی‌هایی را دریافت می‌کنند. دقت کنید در برخی از موارد نسخه خط فرمان **پایتون** عملکردی بهتر از محیط گرافیکی دارد، زیرا به شما اجازه می‌دهد از سویچ‌های خط فرمان استفاده کرده و کنترل بیشتری روی **پایتون** داشته باشید و مهم‌تر از آن به دلیل عدم استفاده از عناصر گرافیکی منابع سیستمی را کمتر استفاده می‌کند.

اجرای پایتون

بسته به پلتفرمی که از آن استفاده می‌کنید راه‌های مختلفی برای اجرای **پایتون** از طریق خط فرمان وجود دارد که سه راهکار زیر جزء مرسوم‌ترین روش‌ها هستند.

- رفتن به پوشه‌ای که پایتون درون آن نصب شده و اجرای فایل Python. با این کار محیط خط فرمان **پایتون** با تنظیمات پیش‌فرض اجرا می‌شود.
- باز کردن ترمینال خط فرمان ویندوز (اجرای دستور cmd) و تایپ Python برای اجرای پایتون در حالت متنی. از این گزینه زمانی استفاده کنید که به دنبال بیشترین انعطاف‌پذیری هستید. این روش به شما اجازه می‌دهد **پایتون** را بر مبنای نیاز کاری خود پیکربندی کرده و اجرا کنید.

مهم نیست از چه روشی برای اجرای **پایتون** در محیط خط فرمان استفاده می‌کنید، در نهایت شما صفحه زیر را مشاهده خواهید کرد. در این پنجره نگارش **پایتونی** که از آن استفاده می‌کنید، سیستم‌عامل میزبان و نحوه دسترسی به اطلاعات اضافی‌تر نشان داده شده است.



چگونه از خط فرمان به شکل پیشرفته استفاده کنیم؟

برای اجرای پایتون در محیط خط فرمان شما می‌توانید در کادر جست‌وجوی ویندوز 10 کلمه Python را تایپ کرده و کلید اینتر را فشار دهید، اما در بیشتر موارد نیاز دارید تا پایتون به شیوه خاص در محیط خط فرمان اجرا شود. شما می‌توانید در زمان اجرا، **پایتون** را همراه با اطلاعات ویژه‌ای فراخوانی کنید که سه مورد زیر جزء رایج‌ترین موارد هستند:

- **گزینه‌ها:** یک گزینه یا سویچ خط فرمان همراه با یک علامت منفی آغاز شده و پس از آن یک یا چند حرف می‌آیند. به‌طور مثال، در نظر دارید از راهنمای پایتون استفاده کنید؟ عبارت python -h را تایپ کرده و کلید

اینتر را فشار می‌دهید. با اینکار اطلاعات اضافی درباره این‌که چگونه از **پایتون** در خط فرمان استفاده کنید ظاهر می‌شود. این سوئیچ در شماره آینده بررسی خواهیم کرد.

• **نام فایل:** در این روش **پایتون** همراه با نام فایل فراخوانی می‌شود. در این حالت **پایتون** به‌طور خودکار برنامه‌ای را فراخوانی کرده و اجرا می‌کند. شما می‌توانید هر یک از برنامه‌هایی که در ادامه ایجاد خواهیم کرد را به این روش در محیط خط فرمان فراخوانی کنید. به‌طور مثال، اگر فایلی به نام SayHello.py دارید برای فراخوانی آن در محیط خط فرمان کافی است عبارت Python SayHello.py را وارد کرده و کلید اینتر را فشار دهید.

• **پارامترها:** یک برنامه می‌تواند اطلاعات اضافی را به عنوان ورودی دریافت کند. اطلاعاتی که نحوه اجرای برنامه را کنترل می‌کنند. این اطلاعات اضافی به نام آرگومان نامیده می‌شوند. درباره این‌که آرگومان‌ها چه هستند و چگونه استفاده می‌شوند نگران نباشید. در این سری از آموزش‌ها با آن‌ها آشنا شده و حسابی از آن‌ها استفاده خواهید کرد.

دقت کنید **پایتون** نسبت به بزرگ و کوچکی گزینه‌ها حساسیت دارد. به‌طور مثال سوئیچ `-s` معنای کاملاً متفاوتی با `S` دارد. از مهم‌ترین سوئیچ‌های خط فرمان **پایتون** می‌توان به موارد زیر اشاره کرد:

`b-` زمانی که برنامه کاربردی شما از **پایتون** همراه با قابلیت‌های ویژه‌ای همچون `str(bytes_instance)`، `str(bytes_array_instance)` و مقایسه `bytes` یا `bytearray` با `str()` استفاده می‌کند، پیغام‌های هشدار را اضافه می‌کند.

`bb-` زمانی که برنامه شما از قابلیت‌های ویژه **پایتون** همچون `str(bytes_instance)`، `str(bytes_array_instance)` و مقایسه `bytes` یا `bytearray` با `str()` استفاده می‌کند پیغام‌های خطایی را به خروجی اضافه می‌کند.

`B-` زمانی که در حال وارد کردن یک مازول هستید اجازه نمی‌دهد چیزی در فایل‌های `py` یا `pyco` بنویسید.

`c cmd-` از اطلاعات `cmd` برای اجرای یک برنامه استفاده می‌شود.

`d-` ابزار اشکال‌زدایی را فراخوانی می‌کند. (این سوئیچ مکان خطاهای رخ داده را مشخص می‌کند.)

`E-` از همه متغیرهای محیطی **پایتون** همچون `PYTHONPATH` که برای پیکربندی **پایتون** استفاده می‌شوند صرف‌نظر می‌کند.

`h-` راهنمای دسترسی به گزینه‌ها و متغیرهای محیطی **پایتون** را نشان می‌دهد و **پایتون** را برای استفاده در اختیاراتان قرار می‌دهد.

`i-` **پایتون** را مجبور می‌کند که پس از اجرای یک اسکریپت به شما اجازه دهد به صورت تعاملی از آن استفاده کنید.

`m mod-` یک مازول کتابخانه که از طریق `mod` در قالب یک اسکریپت مشخص شده است را اجرا می‌کند. این گزینه همچنین به **پایتون** اعلام می‌دارد که فرآیند پردازش سایر اطلاعات مرتبط با سوئیچ‌های دیگر را متوقف کند.

`O-` بایت‌کدهای تولید شده را بهینه‌سازی می‌کند تا سرعت افزایش پیدا کند.

`OO-` یک بهینه‌سازی مضاعف انجام داده و `doc-strings` را حذف می‌کند.

`q-` به **پایتون** می‌گوید که پیام‌های مربوط به نسخه و کپی‌رایت را در زمان شروع تعاملی نشان ندهد.

`s-` مانع از آن می‌شود تا `user site directory` به `sys.path` اضافه شود. (متغیری که به **پایتون** اعلام می‌دارد در چه مکان‌هایی باید به دنبال پیدا کردن مازول‌ها باشد.)

`S-` اجازه مقداردهی اولیه `import site` را نمی‌دهد. فعال بودن این گزینه مانع از آن می‌شود که **پایتون** در

جست و جوی مسیرهایی باشد که ممکن است شامل ماژول‌های مورد نیاز در آن مسیرها باشند.

V- : نسخه جاری **پایتون** را نشان می‌دهد.

W arg- : سطح نمایش پیغام‌های هشدار **پایتون** را تعدیل کرده و آن‌ها را کمتر یا بیشتر می‌کند. مقادیر معتبر برای arg عبارتند از action, message, category, module و ineno

اجرای یک فرمان

پس از آن که محیط خط فرمان را اجرا کردید، قادر به تایپ دستورات هستید. دستورات به شما اجازه می‌دهند کار خاصی را انجام داده، ایده‌ای را آزمایش کرده و در نهایت برنامه خود را بنویسید. در مثال آماده‌سازی نان توست به شما گفتیم که هدف نهایی شما آماده کردن نان توست بر مبنای یک روال بود. زمانی که کره را روی نان توست می‌مالید روال شما خاتمه پیدا می‌کند. روال‌ها در کامپیوتر نیز دقیقاً به همین شکل کار می‌کنند. آن‌ها یک نقطه شروع و یک نقطه پایان دارند. زمانی که دستوری را تایپ می‌کنید، نقطه پایان کار زمانی است که کلید اینتر را فشار می‌دهید. شما با فشار کلید اینتر به کامپیوتر می‌گویید که دستور را اجرا کند. به مرور زمان که با **پایتون** بیشتر آشنا می‌شوید، با راهکارهای مختلفی که **پایتون** برای کامل کردن یک مرحله از آن‌ها استفاده می‌کند آشنا خواهید شد. مهم نیست یک برنامه کاربردی چه مقدار پردازش در پس‌زمینه انجام می‌دهد، در نهایت هر برنامه کاربردی یک نقطه توقف دارد. اکنون می‌دانید که یک فرمان گامی در یک روال است و هر فرمان نقطه شروع و پایانی متفاوت از دیگری دارد. دقت کنید که گروهی از فرمان‌ها یا یک برنامه کاربردی ممکن است نقطه شروع و پایان خاص خود را داشته باشند. اجازه دهید این موضوع را با ذکر مثالی بررسی کنیم.

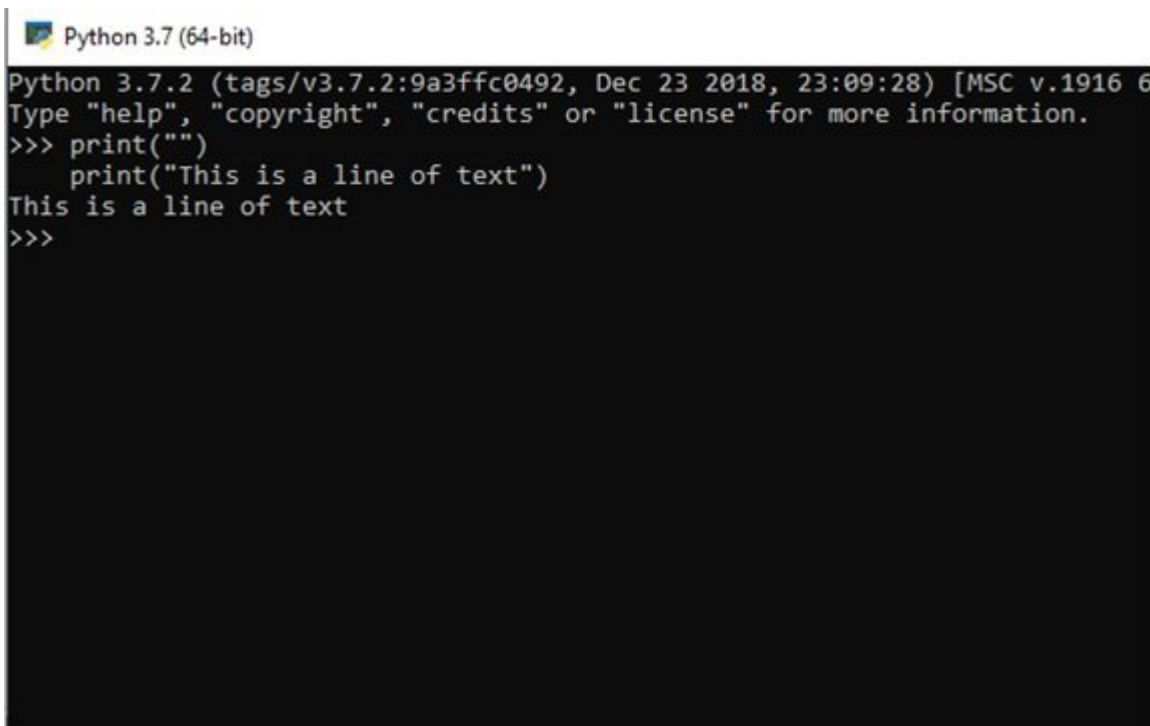
1. **پایتون** را در محیط خط فرمان اجرا کنید.

2. فرمان زیر را تایپ کنید (دقت کنید با تایپ دستور هیچ اتفاقی رخ نمی‌دهد، زیرا فرمان شما هنوز کامل نشده است).

```
print ("This is a line of text")
```

3. کلید اینتر را فشار دهید.

خروجی فرمان فوق همانند شکل زیر است.



```
Python 3.7 (64-bit)
Python 3.7.2 (tags/v3.7.2:9a3ffc0492, Dec 23 2018, 23:09:28) [MSC v.1916 6
Type "help", "copyright", "credits" or "license" for more information.
>>> print("")
print("This is a line of text")
This is a line of text
>>>
```

این مثال به شما نشان می‌دهد که فرمان‌ها در پایتون چگونه کار می‌کنند. هر فرمان تایپ شده در پایتون برای اجرای کار خاصی استفاده می‌شود، اما هر فرمان تنها زمانی اجرا می‌شود که فرآیند تکمیل شدن یک فرمان به صراحت به پایتون اعلام شده باشد. فرمان print () برای نمایش اطلاعات روی صفحه نمایش استفاده می‌شود. در محیط‌های تعاملی پس از نوشتن یک دستور و فشار کلید اینتر دستورات بلافاصله اجرا می‌شوند. اما به مرور زمان متوجه خواهید شد، همه برنامه‌ها بلافاصله نتایج را روی صفحه نشان نمی‌دهند، بلکه باید یکسری پردازش‌هایی در پس‌زمینه انجام دهند و پس از کامل شدن پردازش‌ها نتایج را به شما نشان می‌دهند.

در شماره آینده آموزش رایگان پایتون به سراغ نحوه استفاده از راهنما در پایتون خواهیم رفت.

برای مطالعه تمام قسمت‌های منتشر شده آموزش‌های رایگان پایتون

اینجا کلیک کنید

لطفا نظرات خود در مورد این آموزش و ادامه آن را در بخش دیدگاه در انتهای صفحه اعلام نمایید و نظرات سایر کاربران را نیز ببینید.

معرفی کانال آموزش برنامه‌نویسی و پایتون:

برنامه‌نویسی | پایتون: [@Python_0to100](https://t.me/Python_0to100)

(کانال سری آموزش های رایگان پایتون)

آموزش پایتون: [@learnpy](https://t.me/learnpy)

تاریخ انتشار:

02 بهمن 1397

نشانی منبع:

<https://www.shabakeh-mag.com/workshop/14473/%D8%A2%D9%85%D9%88%D8%B2%D8%B4-%D8%B1%D8%A7%DB%8C%DA%AF%D8%A7%D9%86-%D9%BE%D8%A7%DB%8C%D8%AA%D9%88%D9%86-python-%E2%80%93-%D8%A2%D8%B4%D9%86%D8%A7%DB%8C%DB%8C-%D8%A8%D8%A7-%>

%D9%85%D8%AD%DB%8C%D8%B7-%D8%AE%D8%B7-
%D9%81%D8%B1%D9%85%D8%A7%D9%86-
%D9%BE%D8%A7%DB%8C%D8%AA%D9%88%D9%86-%D9%88-
%D8%B3%D9%88%DB%8C%DB%8C%DA%86%E2%80%8C%D9%87%D8%A7