



قسمت اول

# این مار خوش خط و خال

## آشنایی و کار با زبان برنامه‌نویسی پایتون

« احمد شریف پور

اول: زمانی به برنامه‌نویسی به دید امری عجیب، پیچیده و حتی جادویی نگر بسته می‌شد که تنها در توان خوره‌های کامپیوتر و نواخ علوم الکترونیک و کامپیوتر بود. با ساده‌تر شدن زبان‌های برنامه‌نویسی و کاربرد روزافزون کامپیوتر در زندگی روزمره این دیدگاه به سرعت در حال منسوخ شدن است. تقریباً هر کاربر ساده کامپیوتر ممکن است در شرایطی قرار گرفته باشد که آرزو کند کاش برنامه‌ای برای انجام عملی خاص وجود داشت یا چه خوب می‌شد سیستم را برای انجام چنین عملی برنامه‌ریزی کرد. امروزه، این آرزو بیش از هر زمان دیگری قابل تحقق است.

دوم: می‌گویند یادگیری هر زبان تازه، بعد جدیدی به شخصیت فرد می‌افزاید. اگر چه این مثل ممکن است درباره زبان‌های برنامه‌نویسی درست نباشد، اما آشنا شدن با هر زبان برنامه‌نویسی جدید و پارادایم‌های حاکم بر آن، به یقین به تعالی منطق برنامه‌نویس، افزایش قابلیت‌های وی و از همه مهم‌تر بهبود روش حل مسئله توسط وی منجر خواهد شد. در این میان، پایتون انتخابی مناسب برای کاربران مبتدی و حرفه‌ای خواهد بود (کمی بعد با دلایل این انتخاب آشنا خواهید شد). این مقاله گزیده‌ای است از مجموعه مقالاتی در شماره‌های مختلف مجله Full Circle Magazine (مجله رسمی توزیع لینوکس اوپن‌تو) که به صورت بسیار ساده و در عین حال کاربردی به معرفی این زبان و قابلیت‌های آن خواهد پرداخت. یکی دو قسمت ابتدایی این مجموعه به معرفی زبان پایتون و آشنایی ابتدایی با آن اختصاص داده شده است. مطالب سایر قسمت‌ها کاملاً مستقل از یکدیگر بوده و هر یک با ارائه مثال‌های کاربردی به بررسی جنبه‌ای خاص از پایتون خواهد پرداخت.

## انتخاب اول

مجموعه عظیم زبان‌های برنامه‌نویسی موجود را براساس ویژگی‌ها و قابلیت‌هایشان به دسته‌های مختلفی طبقه‌بندی کرده‌اند. دسته‌بندی‌هایی نظیر شی‌اگرا در برابر روال‌گرا یا این‌سورس در برابر منبع‌بسته، مستقل از پلتفرم در برابر وابسته به پلتفرم، ترجمه‌ای در برابر تفسیری و... هر یک از این دسته‌ها نشانگر ویژگی‌ها و امکاناتی هستند که ممکن است از یک دیدگاه خاص بسیار مفید و ارزنده محسوب شده و از دیدگاهی دیگر به‌عنوان محدودیت شناخته شوند. همین امر انتخاب یک زبان را برای شروع یادگیری به انتخابی دشوار تبدیل می‌کند. در این میان، زبان‌های مشهور یا زبان‌های پرکاربرد لزوماً انتخاب‌های مناسبی نخواهند بود. به‌عنوان مثال، زبان‌هایی نظیر C یا ++C به برنامه‌نویس این امکان را می‌دهند که برنامه‌های خود را با جزئیات بسیار بالا نوشته و با سرعت فوق‌العاده‌ای اجرا کنند. اما این سرعت بالا و کنترل شدید برنامه‌نویس بر تمام جزئیات به قیمت صرف زمان بیشتر برای کدنویسی و رفع اشکال کد تمام خواهد شد. نکته جالب توجه این‌که در بسیاری از موارد سرعت اجرای برنامه‌ها چندان مهم نیست.

در حال حاضر نیز با ارزان شدن مداوم قدرت پردازش سیستم‌ها، بسیاری از برنامه‌نویسان معتقدند، هدر رفتن درصدی از نیروی پردازش پردازنده در اثر استفاده از زبان‌های سطح بالا و در عوض فراهم شدن زمان بیشتر برای تمرکز برنامه‌نویس بر فعالیت اصلی‌اش یعنی کدنویسی، معامله‌ای مقرون به صرفه است. زبانی که برای شروع یادگیری انتخاب می‌شود در ابتدا باید ساده باشد، به برنامه‌نویس تفکر سیستمی و منطقی را بیاموزد، اصول کدنویسی درست را به کاربر آموزش دهد، قادر به پاسخ‌گویی به غالب نیازهای کاربر باشد و در نهایت این‌که بهترین به سیستم‌عامل یا ابزار خاص و پیچیده‌ای وابسته نباشد.

بر همین اساس، بسیاری پایتون را بهترین زبان برای شروع یادگیری برنامه‌نویسی می‌دانند. پایتون زبانی سطح بالا، دینامیک، اسکرپتی، تفسیری، شی‌اگرا، آزاد، این‌سورس و مستقل از پلتفرم است که در سال ۱۹۹۱ توسط گوئی‌دو وان روسوم ابداع شد. باید توجه داشت که معرفی پایتون به‌عنوان زبان مناسب مبتدیان به معنای کم بودن قابلیت‌های آن نیست. هم‌اکنون پایتون به‌عنوان یکی از قدرتمندترین ابزارهای برنامه‌نویسی سیستمی، برنامه‌نویسی وب و... مورد استفاده قرار می‌گیرد.

## چرا پایتون؟

اگرچه ممکن است بسیاری از برنامه‌نویسان قدیمی که با زبان‌هایی نظیر جاوا و C سروکار داشته‌اند، برنامه‌نویسی با پایتون را چندان جدی نگیرند و حتی آن را کودکانه بدانند، اما این امر از قابلیت‌های این زبان نخواهد کاست. در عوض، تعداد زیادی از برنامه‌نویسان (کسانی نظیر برنامه‌نویسان گوگل) پایتون را به دلیل سادگی، خوانایی و امکانات فراوان به هر زبان دیگری ترجیح می‌دهند. طرفداران پایتون که غالباً Pythonistas نامیده می‌شوند، معتقدند، این زبان به قدری کامل و کار با آن به قدری لذت‌بخش است که برای تمام نیازهای برنامه‌نویسی می‌توان به آن مراجعه کرد. آن‌ها برای این ادعا دلایل فنی هم ارائه می‌کنند که برخی از آن‌ها به شرح زیر است:

### سادگی

پایتون زبانی بسیار ساده و مینی‌مالیستی است. خواندن یک برنامه خوب پایتون همانند خواندن یک متن انگلیسی است. این طبیعت شبه‌کد (Pseudo-Code) یکی از بزرگ‌ترین نقاط قوت پایتون است و به شما این امکان را می‌دهد که به جای تمرکز روی زبان بر روش حل مسئله تمرکز کنید.

### سهولت یادگیری

سادگی این زبان باعث می‌شود، یادگیری آن از غالب زبان‌های موجود ساده‌تر باشد. برخی عقیده دارند، یک روز تعطیلی آخر هفته برای یادگیری دستور و قواعد زبان پایتون و بخشی از کتابخانه استاندارد آن کافی است.

### آزاد و این‌سورس بودن (FLOSS)

یکی از بارزترین نمونه‌های FLOSS (سرنام Free/Libre and Open Source Software) پایتون است. به عبارت ساده‌تر، شما می‌توانید نسخه‌هایی از این نرم‌افزار را آزادانه توزیع کنید، کدهای منبع آن را بخوانید، تغییر دهید و از آن در برنامه‌هایتان استفاده کنید. اساس FLOSS بر جامعه‌ای بنا شده که افراد آن دانش خود را با یکدیگر به اشتراک می‌گذارند. یکی از دلایل برتری پایتون نیز همین است که به صورت مداوم توسط جامعه‌ای که تنها خواسته‌اش بهتر شدن این زبان است، بررسی شده و بهبود داده می‌شود.

### سطح بالا بودن (High-Level)

هنگام کدنویسی در پایتون لازم نیست نگران جزئیات سطح پایین باشید. مدیریت حافظه، جمع‌آوری پسماندها (Garbage Collection) و... همه توسط خود زبان انجام خواهد شد.

## قابلیت حمل (Portable)

به واسطه طبیعت این‌سورس پایتون، این زبان به بسیاری از پلتفرم‌ها انتقال داده شده است. تمام برنامه‌های شما می‌توانند بدون هیچ تغییری روی تمامی این پلتفرم‌ها اجرا شوند. البته به شرطی که با هوشیاری از قابلیت‌های خاص وابسته به پلتفرم پرهیز کنید. برنامه‌های پایتون شما به سادگی روی لینوکس، ویندوز، FreeBSD، مک، سولاریس، آمیگا، پلی‌استیشن، Pocket PC و... قابل اجرا هستند.

### تفسیری بودن (Interpreted)

برنامه‌ای که با یکی از زبان‌های ترجمه‌ای نظیر C یا #C نوشته می‌شود، توسط کامپایلر و تنظیم تعداد زیادی گزینه و Flag به زبان قابل فهم ماشین تبدیل می‌شود. وقتی برنامه را اجرا می‌کنید، Loader/Linker برنامه، آن را به حافظه کپی کرده و اجرا می‌کند.

برخلاف این زبان‌ها پایتون به مترجم نیازی ندارد و برنامه شما به‌طور مستقیم از روی کد اجرا می‌شود. در واقع در پشت پرده، مفسر پایتون برنامه شما را به حالتی واسطه‌ای به نام ByteCode تبدیل کرده و سپس اجرا می‌کند.

تمام این‌ها باعث می‌شود که استفاده از پایتون ساده‌تر شود. زیرا لازم نیست نگران چرخه کدنویسی، کامپایل، کنترل، اصلاح کد، کامپایل دوباره یا نگران گزینه‌های کامپایلر، در دسترس بودن و لینک شدن کتابخانه‌های مورد نظر و... باشید. این سیستم قابلیت حمل برنامه را نیز افزایش می‌دهد. زیرا کافی است شما کد را به سیستم دیگری کپی کنید و برنامه بدون نیاز به کامپایل دوباره به سادگی اجرا می‌شود.

### شی‌اگرایی (Object Oriented)

پایتون از هر دو شیوه برنامه‌نویسی شی‌اگرا و روال‌گرا به خوبی پشتیبانی می‌کند. در زبان‌های روال‌گرا، برنامه بر مبنای تعدادی روال یا تابع ساخته می‌شود که در واقع چیزی جز قطعات کد با قابلیت استفاده دوباره نیستند. در زبان‌های شی‌اگرا برنامه بر پایه اشیایی ساخته می‌شود که اطلاعات و عملکرد را یکجا نگه‌داری می‌کنند. پایتون در مقایسه با زبان‌هایی نظیر #C یا جاوا روش بسیار ساده‌تری برای برنامه‌نویسی OOP دارد.

### قابلیت توسعه (Extensible)

اگر در برنامه شما قطعه کدی حساس وجود دارد که لازم است با سرعت بالایی اجرا شود یا می‌خواهید الگوریتم به‌کاررفته در آن را مخفی کنید، می‌توانید آن قسمت را به زبان C یا #C بنویسید و آن را از داخل برنامه پایتون اجرا کنید.

## قابلیت جاسازی (Embeddable)

می‌توان پایتون را به صورت توکار (Embedded) در برنامه‌های C یا C# یا زبان‌های دیگر به کار برد و از این طریق به استفاده‌کنندگان برنامه قابلیت اسکریپت‌نویسی را هدیه کرد.

## کتابخانه استاندارد غنی (Extensive Library)

کتابخانه استاندارد پایتون بسیار گسترده و غنی است. این کتابخانه که به همراه نصب استاندارد پایتون در دسترس خواهد بود، می‌تواند گستره عظیمی از امور را نظیر Document Generation, Regular Expression, Database, Threading, Unit Testing, Cryptography, HTML, XML, FTP, Web Browsing, GUI, و Email پوشش دهد.

در کنار کتابخانه استاندارد پایتون، تعداد زیادی کتابخانه خوش‌ساخت دیگر نظیر wxPython (برای ساخت رابط گرافیکی)، Twisted (فریم ورک توسعه اینترنت)، PIL یا Python Imaging Library (برای ویرایش تصاویر) نیز در دسترس شما است. اما در نهایت شاید مهم‌ترین دلیل طرفداران پایتون هیچ یک از موارد فوق نباشد. آنچه آن‌ها را شایسته این زبان می‌کند لذتی است که از کار با چنین زبان ساده و قدرتمندی نصیب برنامه‌نویس خواهد شد.

## چه کسانی از پایتون استفاده می‌کنند؟

به دلیل ماهیت این سورتس و رایگان بودن، آمار دقیقی از استفاده‌کنندگان این زبان در دسترس نیست، اما شاید نام بردن تعدادی از بزرگ‌ترین شرکت‌های مرتبط و غیر مرتبط با IT که از این زبان استفاده می‌کنند تا حدی نشان‌دهنده عمومیت و قدرت این زبان باشد.

مشهورترین کاربر زبان پایتون غول دنیای اینترنت یعنی گوگل است. پایتون به عنوان زبان موردعلاقه برنامه‌نویسان گوگل شناخته می‌شود. موتور پردازش ابری گوگل با نام Google App Engine از پایتون پشتیبانی می‌کند. شما می‌توانید برنامه‌هایی را که به محاسبه‌های سنگین و پیچیده نیاز دارند با پایتون نوشته و از طریق App Engine اجرای آن را به سرورهای قدرتمند گوگل بسپارید. ناسا نیز از دیگر استفاده‌کنندگان پایتون است. به صورت دقیق‌تر شرکت United Space Alliance شرکت اصلی پیمانکار مسئول نگه‌داری شاتل‌ها در ناسا از پایتون برای برنامه‌نویسی‌های علمی، طراحی رابط کاربر و توسعه سریع نرم‌افزار بهره می‌برد. همچنین شرکت اتودسک که از بزرگ‌ترین تولیدکنندگان

نرم‌افزارهای طراحی به کمک کامپیوتر (CAD) و مصورسازی است، از پایتون برای ایجاد قابلیت اسکریپت‌نویسی در محیط مایا استفاده می‌کند. اما تصور نکنید که استفاده از پایتون تنها به محاسبه‌های علمی و کاربردهای مهندسی محدود است؛ جرج لوکاس شرکت ILM (سرنام Industrial Light and Magic) را در سال ۱۹۷۵ برای ایجاد جلوه‌های ویژه فیلم اصلی جنگ ستارگان تأسیس کرد. این شرکت هم‌اکنون برای ایجاد جلوه‌های ویژه، گرافیک کامپیوتری و مدیریت سیستم‌ها از پایتون استفاده می‌کند. و در نهایت می‌توان از شرکت فیلیپس نام برد. این تولیدکننده سرشناس وسایل الکترونیکی و مصرفی برای توسعه سریع نرم‌افزار، کنترل خط تولید و سیستم کنترل کیفیت خود از پایتون بهره می‌برد. برای مشاهده فهرست کاملی از کاربران مشهور پایتون براساس زمینه فعالیت به همراه خلاصه‌ای کوتاه از نحوه استفاده و دلایل به‌کارگیری این زبان می‌توانید به آدرس <http://python.org/about/success> مراجعه کنید.

## کدام پایتون؟ کدام ابزار؟

زبان پایتون در حال حاضر در دو ویرایش متفاوت منتشر می‌شود. ویرایش اول سری 2.x پایتون است که اکنون توزیع استاندارد پایتون به‌شمار می‌آید و به نسخه 2/7 رسیده است. توسعه در این دسته بر مبنای حفظ سازگاری با نسخه‌های قبلی انجام می‌شود. ویرایش دوم سری 3.x پایتون است که از ابتدا بدون سازگاری با نسخه‌های 2.x و با هدف ایجاد قابلیت‌های بیشتر و به‌روزرسانی زبان تولید شده و اکنون به نسخه 3/1 رسیده است. این نسخه هنوز همانند سری دوم عمومیت نیافته و به همین دلیل در این مجموعه مقاله‌ها از سری 2.x استفاده خواهد شد.

به یقین تمام کدهای نوشته شده در این مجموعه با هر نسخه‌ای بالاتر از 2/2 سازگار بوده و قابل اجرا هستند. این سری مقاله‌ها بر مبنای سیستم‌عامل لینوکس (به صورت خاص اوبونتو) نوشته شده‌اند. اما تمام کدها روی تمام سیستم‌ها قابل اجرا هستند. اگر در کدها از توابع و ویژگی‌های خاص وابسته به سیستم‌عامل استفاده شده باشد، در کنار کد به آن اشاره خواهد شد. در صورتی که شما از یکی از توزیع‌های مشهور لینوکس استفاده می‌کنید، به احتمال زیاد پایتون از پیش روی سیستم شما نصب شده است. برای سیستم‌عامل‌های ویندوز

و مک می‌توانید با مراجعه به سایت رسمی پایتون ([www.python.org](http://www.python.org)) نسخه مناسب سیستم‌عامل خود را دانلود کرده و نصب کنید. کار با زبان برنامه‌نویسی پایتون به هیچ ابزار یا امکانات خاصی احتیاج ندارد. تنها با داشتن یک ویرایشگر متن ساده نظیر Gedit در لینوکس یا Notepad در ویندوز می‌توانید شروع به کدنویسی کنید. اما در صورتی که ویرایشگر شما از رنگ‌بندی کد (Syntax Highlight) و سیستم تورفتگی خودکار (Auto Indent) پشتیبانی کند، سرعت کدنویسی و تصحیح شما بسیار بالاتر خواهد رفت. غالب ویرایشگرهای متن لینوکس نظیر Kate (KDE) و Gedit (GNOME) از هر دوی این قابلیت‌ها پشتیبانی می‌کنند. در ویندوز می‌توانید از Notepad++ که نرم‌افزاری آزاد و رایگان است استفاده کنید. همچنین در ویندوز و مک نصب استاندارد پایتون محیط برنامه‌نویسی پیش‌فرض پایتون یا IDLE را نیز به همراه خواهد داشت که بسیار فراتر از یک ویرایشگر متن ساده در نوشتن و تصحیح کد به شما کمک خواهد کرد. در تمام توزیع‌های لینوکس، این برنامه از مخازن سیستم به‌سادگی قابل نصب است. با مراجعه به آدرس <http://wiki.python.org/moin/PythonEditors> می‌توانید فهرست ویرایشگرها و محیط‌های توسعه موجود را برای پایتون مشاهده کنید.

## مثال مقدماتی ۵ دقیقه‌ای

### نمونه کد پایتون

فرض کنید می‌خواهیم حاصل عبارت  $g(x) = \frac{x}{1-x}$  را برای مقادیر  $x$  معادل 0.1, 0.2, 0.3, ..., 0.9 بیابیم. کافی است در یک فایل متنی مثلاً با نام `fme.py` کد موجود در فهرست ۱ را نوشته و در خط فرمان با تایپ دستور زیر آن را اجرا کنید.

```
$ python fme.py
```

خروجی برنامه مانند فهرست ۲ خواهد بود (می‌توانید این عملیات را در IDLE نیز انجام دهید. کافی است پس از بازکردن برنامه از منوی File گزینه New Window را انتخاب کرده، کد موردنظر را تایپ کرده و پس از ذخیره کردن با F5 اجرا کنید).

```
for i in range(10):
    x=0.1*i
    print x
    print x/(1-x*x)
```

فهرست ۱

در این حالت فقط زمانی که متغیر  $i$  یک عدد فرد است ما دستور چاپ را اجرا می‌کنیم. عملکرد  $\%mod$  همانند  $C++/C$  باقی‌مانده تقسیم را باز می‌گرداند. دوباره به علامت دونقطه در انتهای دستور `if` و فاصله‌گذاری ابتدای دستور `print` دقت کنید. همین‌طور دقت داشته باشید که مانند  $C++/C/perl$  احتیاجی به علامت نقطه کاما در انتهای دستورات نیست. هر خط جدید به معنای دستوری جدید خواهد بود. در صورتی که خطی طولانی دارید می‌توانید برای شکستن آن از علامت `"\"` استفاده کنید (فهرست ۶).

```
>>> y=7
>>> z=12.5
>>> x=y+\
      z
>>>
>>> x
19.5
```

فهرست ۶

### پوسته تعاملی (خط فرمان پایتون)

یکی از قابلیت‌های جذاب پایتون خط فرمان تعاملی آن است. شما به صورت معمول زیاد از این قابلیت استفاده نمی‌کنید، اما برای بررسی برخی امکانات و آزمایش سریع دستورات بسیار مفید خواهد بود. هرگاه به عملکرد درست قسمتی شک دارید باید از سیستم آزمایش و خطا استفاده کنید و بهترین و سریع‌ترین روش آن‌موند استفاده از خط فرمان تعاملی پایتون است. به جای نوشتن کد در فایل متنی و اجرای آن ما کد را در پوسته پایتون نوشته و اجرا می‌کنیم. با تایپ دستور Python به‌تنهایی یا اجرای `IDLE` به پوسته پایتون دسترسی خواهیم داشت (شکل ۱).

در اینجا با اجرای پایتون اعلان حالت تعاملی یعنی `>>>` به نمایش درمی‌آید. سپس شروع به تایپ خط به خط می‌کنیم. هر زمان که داخل یک بلوک کد باشیم اعلان ویژه `"..."` به نمایش درمی‌آید. وقتی در انتهای کد یک خط خالی را وارد کنیم، پایتون متوجه اتمام کد شده و آن را اجرا می‌کند. درباره پوسته تعاملی توجه داشته باشید که بین پلتفرم‌های مختلف تفاوت‌های کمی وجود دارد. به عنوان مثال، ممکن است فاصله‌گذاری به صورت خودکار انجام نشود یا در ویندوز شما با تایپ `CTRL+Z` از پوسته خارج می‌شوید، اما در لینوکس باید از `CTRL+D` استفاده کنید. همچنین توجه کنید که

```
>>> x=5
>>> while 1:
      x += 1
      if x == 8:
          print x
          break
8
>>>
```

فهرست ۳

### تعریف بلوک در پایتون

حال به علامت دونقطه انتهای دستور `for` توجه کنید که شروع یک بلوک کد را اعلام می‌کند. برخلاف زبان‌هایی نظیر  $C++/C$  یا حتی Perl، که از `{}` برای تعریف بلوک استفاده می‌کنند، پایتون از ترکیب دونقطه و تورفتگی برای تعریف بلوک‌ها استفاده می‌کند. ما با دونقطه به پایتون می‌گوییم که قصد داریم یک بلوک کد را آغاز کنیم و با تورفتگی ابتدای خطوط دستورات آن بلوک را تعریف می‌کنیم. برای تورفتگی می‌توانیم از ۳ یا ۴ یا ۱۰ فضای خالی استفاده کنیم. میزان این فضای خالی تا زمانی که در تمام کد رعایت شود، اهمیتی ندارد. بنابراین، در صورت وارد کردن کد (فهرست ۴) با خطای دستوری پایتون مواجه می‌شویم، زیرا تنها در صورتی می‌توانیم در یک بلوک از تورفتگی استفاده کنیم که بخواهیم بلوک دیگری در دل آن ایجاد کنیم (فهرست ۵).

```
>>> for i in range (10) :
      print 0.1*i
      print g(0.1*i)
File "<pyshe11#18>", line 3
print g(0.1*i)
IndentationError: unexpected indent
>>>
```

فهرست ۴

```
>>> for i in range (10) :
      if i%2 == 1 :
          print 0.1*i
0.1
0.3
0.5
0.7
0.9
>>>
```

فهرست ۵

```
>>>
0.0
0.0
0.1
0.10101010101
0.2
0.208333333333333
0.3
0.32967032967
0.4
0.47619047619
0.5
0.6666666666667
0.6
0.9375
0.7
1.37254901961
0.8
2.22222222222
0.9
4.73684210526
>>>
```

فهرست ۲

### لیست‌ها در پایتون

این برنامه چگونه کار می‌کند؟ در ابتدا تابع `range()` پایتون نمونه‌ای از استفاده لیست‌ها را در این زبان نشان می‌دهد.

در زبان پایتون لیست چیزی شبیه آرایه است. لیست‌ها یکی از انواع بنیادی پایتون هستند. تابع `range()` در واقع یک لیست را برمی‌گرداند که در اینجا عبارت است از `[0,1,2,3,4,5,6,7,8,9]`. به علامت‌های `[]` توجه کنید.

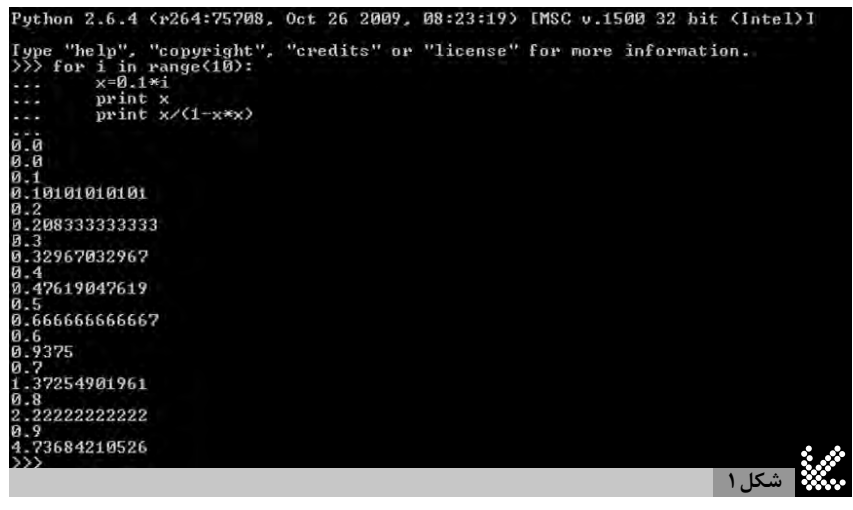
این روش رسمی پایتون برای نشان دادن یک لیست است. لیست در واقع ردیفی از اشیای مختلف (هر تعداد شیء با هر ترتیب و هر نوعی) است که داخل `[]` قرار دارند و با کاما از یکدیگر جدا شده‌اند. پس دستور اول کد ما در واقع معادل عبارت زیر است:

```
for i in [0,1,2,3,4,5,6,7,8,9]:
```

همان‌طور که حدس می‌زنید این کد باعث ده بار گردش حلقه می‌شود که در بار نخست  $i$  برابر صفر و در بار دهم برابر ۹ است. پایتون یک ساختار `while` هم دارد (البته بدون `until`). همچنین مانند  $C++/C$  دستور `break` برای خروج کامل از حلقه در دسترس است (فهرست ۳).



شکل ۲



شکل ۱

شما اکنون تا حدودی با زبان پایتون آشنا شده‌اید و کم‌وبیش توانایی‌ها و قابلیت‌های آن را می‌شناسید. اگر در قسمت‌های بعدی این مجموعه با ما همراه باشید، در هر قسمت با جنبه‌ای تازه و کاربردی از این زبان آشنا خواهید شد.

قسمت‌های بعدی در عین سادگی و کاربردی بودن مستقل از یکدیگر بوده و برای مطالعه و استفاده از آن‌ها به مهارت خاصی نیاز ندارید و حتی لازم نیست به پایتون چندان مسلط باشید، زیرا در هر قسمت مطالب پایه‌ای مرتبط آورده خواهند شد. اما در صورتی که زیبایی پایتون توجه شما را جلب کرده و مشتاق یادگیری بهتر این زبان هستید، به ترتیب دو منبع زیر را به شما پیشنهاد می‌کنیم:

1- A Byte of Python (Swaroop CH)

<http://www.swaroopch.com>

2- Dive into Python (Mark Pilgrim)

<http://www.diveintopython.org>

متأسفانه زبان هر دو منبع انگلیسی است و هنوز مرجعی مناسب و رایگان به زبان فارسی وجود ندارد. در پایان به خاطر داشته باشید که دنیای نرم‌افزار دنیایی است که در آن تنها محدودیت، خلاقیت و دانش شما از برنامه‌نویسی است.



88.88 به علاوه پنج درصد کافی است، عبارت اول را تایپ کنیم (فهرست ۸) یا برای تبدیل اعداد بین مبنای ۱۰ و ۱۶ از دستورات قسمت دوم استفاده کنیم.

```
>>> 1.05*88.88
93.323999999999998
>>>
>>>
>>> 0x12
18
>>> hex(18)
'0x12'
>>>
```

فهرست ۸

اگر به محاسبه‌های ریاضی خاصی احتیاج داشته باشیم باید ابتدا کتابخانه math را از مجموعه کتابخانه‌های استاندارد پایتون import کنیم. دقیقاً همانند سیستمی که در سی یا سی شارپ با دستور #include پیاده می‌شود، سپس می‌توانیم با سیستم نشانه‌گذاری نقطه‌ای از توابع تعریف شده در کتابخانه math استفاده کنیم. در شکل ۲ استفاده از توابع sin() و sqrt() نشان داده شده است. همچنین توجه کنید که در محیط IDLE با تایپ math و علامت نقطه و مکئی کوتاه فهرست توابع موجود کتابخانه ظاهر می‌شود.

در IDLE اعلان "..." به نمایش در نمی‌آید. در حالت تعاملی ارجاع به یک شیء یا ساخت یک شیء یا نوشتن یک عبارت بدون نسبت دادن آن به یک متغیر باعث چاپ شدن مقدار آن می‌شود (فهرست ۷).

```
>>> for i in range(4):
    3*i

0
3
6
9
>>> open("x.txt")
<open file 'x.txt', mode 'r' at 0x01944390>
>>>
```

فهرست ۷

در مثال دوم ما فایل x.txt را باز کرده‌ایم که نتیجه حاصل یک شیء فایل است و چون آن را به متغیری (معمولاً f) نسبت نداده‌ایم، شیء فایل چاپ شده است (فایل x.txt باید در مسیر جست‌وجوی پایتون موجود باشد).

**پایتون به عنوان یک ماشین حساب**  
یکی از قابلیت‌های دیگر پوسته تعاملی امکان استفاده از آن به عنوان یک ماشین حساب سریع است. به عنوان مثال، برای محاسبه مقدار

برای خرید آنلاین کتاب‌های  
انتشارات عرصه شبکه به آدرس  
[www.shabakeh-mag.com](http://www.shabakeh-mag.com)  
مراجعه کنید.

[www.shabakeh-mag.com](http://www.shabakeh-mag.com)



# شکستن مرزهای فروش

کتابی که باید یک میلیون تومان فروخته می‌شود!

نویسنده: ژان بتوسان