

آموزش کاربردی برنامه‌نویسی به زبان

# Python

مهندس علیرضا عظیم زاده میلانی

انتشارات پندار پارس

## انتشارات پندار پارس

### انتشارات پندار پارس



دفتر فروش: انقلاب، ابتدای کارگر جنوبی، کوی رشتچی، شماره ۱۴، واحد ۱۶ [www.pendarepars.com](http://www.pendarepars.com)

تلفن: ۶۶۵۷۲۳۳۵ - تلفکس: ۶۶۹۲۶۵۷۸ همراه: ۰۹۱۲۲۴۵۲۳۴۸ [info@pendarepars.com](mailto:info@pendarepars.com)



نام کتاب : آموزش کاربردی برنامه نویسی به زبان Python

ناشر : انتشارات پندار پارس

ترجمه و تالیف : علیرضا عظیم زاده میلانی

چاپ نخست : دی ماه ۹۴

شمارگان : ۲۰۰۰ نسخه

چاپ، صحافی : روز

قیمت : ۱۹۰۰۰ تومان شابک : ۹۷۸-۶۰۰-۶۵۲۹-۹۶-۷



\*هرگونه کپی برداری، تکثیر و چاپ کاغذی یا الکترونیکی از این کتاب بدون اجازه ناشر تخلف بوده و پیگرد قانونی دارد \*

برای دسترسی به سورس کد مثال‌های کتاب، ثبت بوک کد کتاب در سایت الزامی است.

## فهرست

۵.....	فصل نخست: کاوش در پایتون .....
۵ .....	پایتون چیست .....
۶.....	ویژگی‌های پایتون .....
۶.....	نصب و راه اندازی پایتون .....
۹.....	انتخاب یک IDE (Integrated Development Environment) .....
۱۰.....	نحوهای (Syntax) پایه پایتون .....
۱۲.....	شناسه‌گرها .....
۱۲.....	خط‌ها و تورفتگی‌ها .....
۱۴.....	کوته‌نویسی‌ها .....
۱۵.....	توضیحات .....
۱۵.....	دریافت ورودی .....
۱۶.....	دریافت آرگومان از خط فرمان .....
۱۷.....	انواع متغیرها(داده‌ها) .....
۱۹.....	انواع عملگرها .....
۱۹.....	عملگرهای حسابی و مقایسه‌ای(رابطه‌ای) .....
۲۰.....	عملگرهای انتساب دهی .....
۲۱.....	عملگرهای منطقی و بیتی .....
۲۳.....	عملگرهای عضویت .....
۲۴.....	عملگرهای هویت .....
۲۵.....	توابع Built-in .....
۲۶.....	دستور-تابع Print .....
۲۸.....	(Slicing) Indexing .....
۳۳.....	فصل دوم: ساخت برنامه‌های پایتون .....
۳۴.....	شروط تصمیم‌گیری .....

۳۹	حلقه‌های تکرار
۴۳	دستورات کنترل حلقه
۴۷	<b>فصل سوم: تعریف توابع</b>
۴۹	آرگومان‌های تابع
۵۴	محدوده میدانی متغیرها (Scope of Variables)
۵۵	مدیریت خطاها
۵۶	Exception
۶۲	Assertions
۶۵	<b>فصل چهارم: اعداد، رشته‌ها، لیست‌ها، تاپل‌ها و دیکشنری‌ها</b>
۶۵	بخش نخست، اعداد
۶۷	بخش دوم، رشته‌ها
۶۷	کارکترهای غیر قابل چاپ (Escape Characters)
۶۸	عملگرهای رشته‌ای ویژه
۶۹	عملگر رشته‌ای فرمت‌دهی % (String Formatting Operator)
۷۶	بخش سوم، لیست‌ها
۸۵	بخش چهارم، تاپل‌ها
۸۸	بخش پنجم، دیکشنری‌ها
۹۷	<b>فصل پنجم: فایل‌ها</b>
۱۰۷	ماژول‌ها
۱۱۱	بسته‌ها (Packages)
۱۱۵	<b>فصل ششم: کلاس‌ها</b>
۱۱۶	کار با کلاس‌ها، سازنده‌ها و تخریبگرها
۱۲۳	ارث‌بری در کلاس‌ها
۱۲۹	<b>فصل هفتم: برنامه‌نویسی چندنخی (MULTITHREAD)</b>
۱۳۱	تعریف نخ در ماژول threading

۱۳۹	.....	SELENIUM-WEBDRIVER کار با هشتم: فصل
۱۴۲	.....	find استفاده از متدهای پیدا کردن عناصر با
۱۴۹	.....	Xpath استفاده از پیدا کردن عناصر با
۱۵۴	.....	WebDriver کلاس
۱۵۷	.....	WebElement کلاس
۱۵۸	.....	Alert کلاس
۱۶۱	.....	URLLIB(2) و MECHANIZE، REQUESTS های مازول کار با نهم: فصل
۱۶۱	.....	requests، بخش نخست
۱۶۷	.....	urllib(2)، بخش دوم
۱۷۲	.....	mechanize، بخش سوم
۱۸۵	.....	اطلاعات تکمیلی: پیوست ۱
۱۸۵	.....	ارسال ایمیل، بخش نخست
۱۹۰	.....	سوکت، بخش دوم
۲۰۰	.....	PyCharm و کار با آشنایی، بخش سوم
۲۰۵	.....	مقایسه زبان‌های برنامه‌نویسی، بخش چهارم
۲۱۵	.....	سورس کد: پیوست ۲
۲۱۵	.....	VBulletin در فرم‌های کاربری جدید ساخت حساب‌های کاربری
۲۱۸	.....	VBulletin (کاربران) فرم‌های عضوهای
۲۱۹	.....	VBulletin فرم‌های شخصی به کاربران فرم‌های
۲۲۵	.....	Image File Hacking: برنامه پنجم



## درباره مؤلف

مهندس علیرضا عظیم‌زاده میلانی دانشجوی مقطع کارشناسی ارشد دانشگاه خواجه نصیرالدین طوسی تهران - گرایش شبکه‌های کامپیوتری می‌باشد. ایشان در مهر ۹۴ موفق به کشف چندین آسیب‌پذیری ناشناخته با سطح "ریسک امنیتی بالا" در دو مدل WIMAX شدند و آنها را در سایت‌های معتبر Oday.today و exploit-db به ثبت رساندند. همچنین در پروژه VLAN مدیریت شعب مرکزی بانک تجارت استان زنجان همکاری داشته‌اند و نیز سخنران سمینارهای علمی درباره "راهکارهای نفوذ و مقابله با تهدیدها در شبکه‌های کامپیوتری" در قم و نیشابور بوده‌اند. ایشان مؤلف کتاب "آموزش کاربردی تست نفوذ با کالی لینوکس"، مؤلف و مترجم کتاب‌های الکترونیکی "هک و امنیت با BackTrack" و "مدیریت سرورهای لینوکس با CentOS" نیز می‌باشد.

## مدارک و تخصص‌ها

CCNA, CEH, Security+, LPIC-1, LPIC-2, RHCSA, RHCE, HIDS, WiFu, PWK, ....

## مقدمه مؤلف

خواندن این کتاب به تمام برنامه‌نویسان متعصب به یک زبان برنامه‌نویسی، مدیران و کارشناسان شبکه، امنیت شبکه، هکرها و نفوذگران قانون‌مند، دانش‌آموزان مقطع دبیرستان و دانشجویان علاقه‌مند به برنامه‌نویسی در تمامی رشته‌های دانشگاهی توصیه می‌شود. شاید برایتان عجیب باشد که این زبان برنامه‌نویسی حتی می‌تواند برای پژوهشگران/اساتید و دانشجویان محقق که در حال کار روی پروژه‌های آماری، آنالیز داده‌ها، داده‌کاوی (Data Mining)، سیستم‌های اطلاعات مکانی (GIS)، داده‌های بزرگ (Big Data)، بیولوژی، مدیریت ارتباط با مشتری (CRM)، ساخت قطعات/سیستم‌های هوشمند Embedded، شبیه‌سازی، برنامه‌ریزی منابع سازمانی (ERP) و... هستند نیز مفید باشد؟! اگر تاکنون و به هر دلیلی با هیچ یک از زبان‌های برنامه‌نویسی موجود کار نکرده‌اید یا آشنایی ندارید، اصلاً نگران نباشید. یادگیری این زبان، گامی بزرگ و یک سکوی پرتاب در راستای پیاده‌سازی ایده‌ها و نیازهای کاری‌تان می‌باشد.

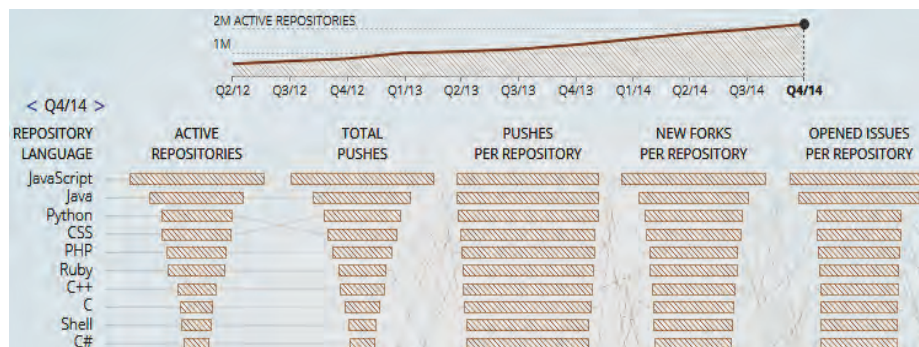
## چرا پایتون؟

اگرچه ممکن است بسیاری از برنامه‌نویسان قدیمی که با زبان‌هایی چون Java و C سروکار داشته‌اند، برنامه‌نویسی با پایتون را چندان جدی نگیرند و حتی آن را کودکانه تصور کنند؛ اما این تفکر از قابلیت‌های این زبان نخواهد کاست. در عوض، تعداد بسیاری از برنامه‌نویسان (همچون برنامه‌نویسان شرکت Google) پایتون را به دلیل سادگی، خوانایی و امکانات فراوانش به هر زبان دیگری ترجیح

می‌دهند. دستداران پایتون که غالباً **Pythonistas** نامیده می‌شوند، معتقدند: این زبان به قدری کامل و کار با آن لذت‌بخش است که برای تمام نیازهای برنامه‌نویسی می‌توان به آن مراجعه کرد. برخی از مهمترین دلایل فنی این دستداران را می‌توانید در فصل نخست کتاب مطالعه نمایید.

در جدول‌های زیر می‌توانید گزارشی از رتبه‌بندی زبان‌های برنامه‌نویسی که توسط سه سایت معتبر در حوزه IT ارائه شده است، مشاهده نمایید.

(۱) منبع سرویس **GitHub** بیش از دو میلیون مخزن کد فعال در سایت GitHub را آنالیز، و گزارش زیر را ارائه کرده است:



(۲) سایت **codeeval** هر سال آماری مبنی بر محبوب‌ترین و رایج‌ترین زبان‌های برنامه‌نویسی را منتشر می‌کند، که این آمار از میان هزاران برنامه‌نویس فعال در صنعت IT جمع‌آوری شده است:

2015 Rank		Change	2014 Rank	Change	2013 Rank	Change	2012 Rank
1	Python	0	1	0	1	0	1
2	Java	0	2	0	2	0	2
3	C++	0	3	0	3	0	3
5	Ruby	-1	4	0	4	0	4
4	C#	2	6	2	8	1	9
7	C	0	7	-1	6	4	10
6	JavaScript	-1	5	2	7	-1	6
8	PHP	0	8	-3	5	0	5

(۳) سرویس **TOIBE-Index** با بررسی تعداد مهندسان این حوزه، دوره‌های آموزشی و رتبه‌بندی موتورهای جستجو، گزارش زیر را ارائه کرده است:

Nov 2015	Nov 2014	Change	Programing Language	Ratings	Change
1	2	↗	Java	20.403%	+6.01%
2	1	↘	C	17.145%	-0.32%



Nov 2015	Nov 2014	Change	Programing Language	Ratings	Change
3	4	▲	C++	6.198%	+0.10%
4	5	▲	C#	4.318%	-0.67%
5	7	▲	Python	3.985%	+1.95%
6	6		PHP	3.213%	+0.15%

اگر قصد مقایسه‌ی کدهای یک برنامه به چهار زبان C، C++، C# و Java با Python را دارید، هم اکنون به بخش چهارم از پیوست-۱ (صفحه‌ی ۲۰۵) رجوع نمایید.

مطالب آموزشی این کتاب به صورت کاربردی و همراه با آخرین تغییرات در نسخه‌ی پایتون 2.x (ابزارها، دستورات، برنامه‌ها، توابع، ماژول‌ها و...) تشریح شده است. در نگارش این کتاب، اطلاعات فنی و تجربی بسیاری که برای یادگیری آنها نیاز به ماه‌ها تجربه می‌باشد، گنجانده شده است. دلیل انتخاب نسخه‌ی ۲ پایتون و نه نسخه‌ی ۳، بخاطر توضیحات بسیار کاملی است که در ویکی (Wiki) سایت پایتون ارائه شده است. و این موضوع بیانگر سنگین‌تر بودن کفهی ترازو به نفع نسخه‌ی ۲ پایتون است (<https://wiki.python.org/moin/Python2orPython3>).

به خوانندگان عزیز پیشنهاد می‌شود، تا حد امکان همه فصل‌های این کتاب را مطالعه نمایند. زیرا مطالب بیان شده در هر فصل، پیش نیاز فصل‌های دیگر می‌باشد.

امید است این کتاب برای شما عزیزان مفید باشد و از خواندن آن لذت ببرید. برای بهبود کمی و کیفی این کتاب از خوانندگان عزیز درخواست می‌شود با مشاهده هرگونه ایراد یا خطای فنی در کتاب، با ایمیل اینجانب ([ali.azimzadeh70@gmail.com](mailto:ali.azimzadeh70@gmail.com)) با موضوع "کتاب پایتون" مکاتبه فرمایند.

در پایان به عرض خوانندگان گرامی می‌رسانم، کتاب دیگری با عنوان "آموزش کاربردی وب هکینگ" توسط اینجانب در حال تألیف و ترجمه می‌باشد که در آینده‌ای نزدیک تقدیم به علاقه‌مندان گرامی می‌گردد.

از حمایت‌های مادی و معنوی شما خوانندگان عزیز که از کپی برداری، اسکن و الکترونیکی کردن کتاب پرهیز می‌نمایید و مؤلف و ناشر را برای تألیف و نشر کتاب‌های بعدی یاری می‌رسانید، بسیار سپاسگزار می‌نمایم.

[ali.azimzadeh70@gmail.com](mailto:ali.azimzadeh70@gmail.com)

[aliazimzadeh.blogfa.com](http://aliazimzadeh.blogfa.com) AND [irsecteam.org](http://irsecteam.org) (Iran Security Team)



# فصل نخست

## کاوش در پایتون

### پایتون چیست

پایتون یک زبان برنامه‌نویسی سطح بالا (high-level)، مفسری (interpreted)، محاوره‌ای (interactive) و شی‌گرای اسکریپتی (object-oriented scripting language) می‌باشد. زبان پایتون بدین خاطر طراحی شده تا قابلیت خوانایی بالاتری نسبت به دیگر زبان‌های برنامه‌نویسی داشته باشد (استفاده از کلید واژه‌های انگلیسی کاملاً معنادار، و در تعدادی کمتر نسبت به دیگر زبان‌ها - کمتر از ۴۰ کلید واژه).

- **مفسر:** مفسری بودن پایتون بدین معناست که برنامه‌های شما در زمان اجرا از نظر قواعد دستوری و ساختاری مورد بررسی قرار می‌گیرند و دیگر نیازی به کامپایل شدن برنامه پیش از هر اجرا نیست (دو زبان برنامه‌نویسی مفسری: Perl و PHP).
- **محاوره‌ای:** پایتون برخلاف دیگر زبان‌های برنامه‌نویسی به شما اجازه می‌دهد به صورت مستقیم با محیط prompt پایتون ارتباط برقرار کنید و دستورهای خود را وارد نمایید و در همان لحظه خروجی دستورات را مشاهده نمایید:

```
>>> 2 + 6
```

```
8
```

- **شی‌گرا:** در برنامه‌نویسی شی‌گرا امکاناتی مانند کپسوله‌سازی، قابلیت استفاده مجدد، حل مسائل پیچیده و... این امکان را به ما می‌دهد تا بتوانیم داده‌های خود را پنهان نماییم، وابستگی به توابع را کمتر کنیم، انعطاف برنامه را افزایش دهیم و... تمام اینها مبنای پایه زبان پایتون است.

## ویژگی‌های پایتون

- **ساده در یادگیری:** پایتون دارای کلید واژه‌های اندک، ساختاری ساده و نحوهای (Syntax) از پیش تعریف شده‌ی واضحی است. این ویژگی به دانشجویان، برنامه‌نویسان و... اجازه می‌دهد تا در کمترین زمان ممکن، کارهای خود را پیاده‌سازی نمایند.
- **ساده در خواندن:** کدهای پایتون وضوح بیشتری برای چشم دارد و آسان‌تر قابل مشاهده است.
- **ساده در نگهداری:** موفقیت پایتون به‌خاطر سادگی در نگهداری و پشتیبانی از سورس-کدش است.
- **قابل انتقال:** برنامه‌هایی که به زبان پایتون نوشته می‌شوند می‌توانند در سایر پلتفرم‌ها نیز اجرا شوند (Windows, Linux/Unix, Macintosh, Android و...).
- **قابل توسعه:** می‌توانید ماژول‌های سطح-پایین نیز به مفسر پایتون خود اضافه نمایید. این ماژول‌ها، برنامه‌نویسان پایتون را توانمند می‌سازد که امکانات بیشتری به ابزارها و ماژول‌های کنونی‌شان اضافه نمایند تا برنامه نهایی از کارایی بیشتری برخوردار باشد.
- **پایگاه‌داده:** این زبان واسط‌های بسیار زیادی برای برقراری ارتباط با دیتابیس‌های رایگان و تجاری در اختیار برنامه‌نویسان می‌گذارد (Microsoft Access, Microsoft SQL Server 2003 and Later, IBM-DB2, SAP-DB و...).
- **برنامه‌نویسی GUI:** زبان پایتون این امکان را می‌دهد تا بتوانید برنامه‌های کاربردی تحت وب یا دستکاپ (حتی با واسط گرافیکی - Graphical User Interface) نیز طراحی نمایید.
- **متن باز.**
- **مُد محاوره‌ای.**

افزون بر ویژگی‌های بالا، پایتون ویژگی‌های خوب دیگری مانند: پشتیبانی از متدهای برنامه‌نویسی ساختاری و تابعی، استفاده کردن از برنامه نوشته شده به‌صورت اسکریپت یا کامپایل کردن آن به ByteCode برای ساخت برنامه‌های کاربردی در مقیاس بزرگ، پشتیبانی از تکنیک "automatic garbage collection"، و یکپارچه شدن با دیگر زبان‌های برنامه‌نویسی (C/++, ActiveX, CORBA و Java)، کدنویسی در قطعات الکترونیکی Embedded (Arduino, Raspberry) و... در خودش دارد.

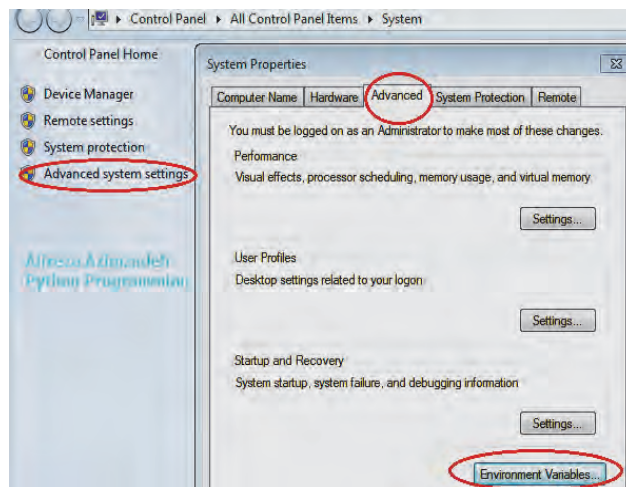
## نصب و راه اندازی پایتون

هر نوع سیستم‌عاملی که داشته باشید، با توجه به پلتفرم خود می‌توانید نسخه‌ی مناسب خود را انتخاب کرده و کار با پایتون را آغاز نمایید. برای دانلود این توزیع می‌توانید به سایت رسمی

www.python.org/downloads مراجعه نمایید. در این لحظه از تألیف کتاب، آخرین نسخه‌های منتشر شده Python-2.7.10 و Python-3.5.1 می‌باشند.

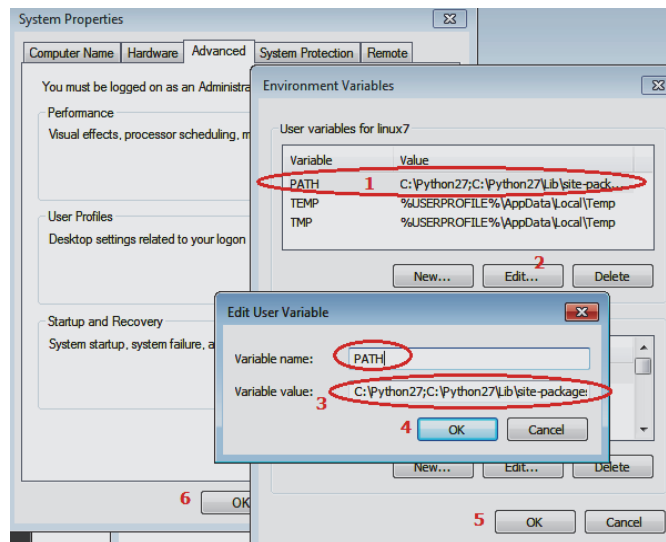
**نکته مهم:** روند مطالب آموزشی این کتاب بر پایه پایتون 2.x.x است (2.6.x, 2.7.x, 2.8.x و...).

- **پایتون در ویندوز:** ابتدا وارد بخش دانلود سایت Python.org شوید و با توجه به معماری سیستم‌عامل‌تان، نسخه‌ی ۳۲ یا ۶۴ بیت را دانلود نمایید. پیشنهاد می‌شود که در هنگام نصب (انتخاب مسیر)، مسیر پیش‌فرض نصب پایتون را تغییر ندهید (C:\Python27). پس از نصب پایتون باید مسیر بسته‌ها و ماژول‌های پایتون را تعیین نماییم؛ برای اینکار بر روی آیکن Computer خود کلیک راست کرده و گزینه Properties را انتخاب نمایید. در ویندوز ۷، از لیست نوشته‌های سمت چپ، گزینه‌ی "Advanced System Settings" را انتخاب نمایید؛ سپس به برگه Advanced بروید و گزینه "Environment Variables..." را انتخاب کنید. همانند شکل زیر:



از بخش User-Variables در کادر داخلی‌اش، متن حاوی PATH را انتخاب کرده و روی گزینه Edit... کلیک کنید، سپس خط زیر را در کادر Variable-value وارد نمایید. همانند شکل زیر:

C:\Python27;C:\Python27\Lib\site-packages\;C:\Python27\Scripts;



در پایان، یک محیط Command-Prompt باز کنید و واژه python را بنویسید تا وارد محیط مفسری پایتون شوید (Start Menu>>Accessories or Start Menu>>Run>>cmd).

```
C:\Users\linux7>
C:\Users\linux7>python
Python 2.7.8 (default, Jun 30 2014, 16:03:49) [MSC v.1500 32 bit (Intel)] on win
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

● **پایتون در لینوکس / یونیکس:** پیش از نصب پایتون در لینوکس، بسته‌های زیر را نصب نمایید:

**Linux:** sudo apt-get install libreadline-gplv2-dev libncursesw5-dev sqlite3 libsdl-dev libgdbm-dev libc6-dev libbz2-dev build-essential libssl-dev tk-dev openssl

**Linux:** wget -c "https://www.python.org/ftp/python/2.7.9/Python-2.7.9.tgz"

**Linux:** tar -xvf Python-2.7.9.tgz

**Linux:** cd Python-2.7.9; sudo ./configure; sudo make; sudo make install

**Linux:** python -V #show the version of python.

**Linux:** python #Run the python interpreter.

**نکته:** دو ابزار کارآمد در زمینه‌ی مدیریت ماژول‌ها و بسته‌ها، pip و easy\_install نام دارند. ابزار pip از نسخه 2.7.9 به بعد به‌طور پیش‌فرض نصب شده و دیگر نیازی به نصب آن ندارید (لینک اصلی اسکریپت pip: <https://bootstrap.pypa.io/get-pip.py>، لینک اصلی اسکریپت easy\_install: <https://bootstrap.pypa.io/get-pip.py>).

و Mechanize ([https://bitbucket.org/pypa/setuptools/raw/bootstrap/ez\\_setup.py](https://bitbucket.org/pypa/setuptools/raw/bootstrap/ez_setup.py)). BeautifulSoup دو مورد از کتابخانه‌های (ماژول) پر کاربرد پایتون هستند که برای آشنایی با ابزارهای pip و easy\_install در این بخش، می‌خواهیم آنها را نصب نماییم. چهار دستور زیر مشابه یکدیگر هستند و نحو (syntax) آنها در همه پلتفرم‌ها (لینوکس، ویندوز،...) یکی است:

**pip install <Name of module(s) with a space>**

**easy\_install <Name of module(s) with a space>**

pip install mechanize beautifulsoup4

python -m pip install mechanize --upgrade

pip install beautifulsoup4 csvkit

easy\_install beautifulsoup4 csvkit

یکی از پارامترهای کاربردی ابزار pip، freeze نام دارد. این پارامتر، ورژن ماژول‌های نصب شده را نشان می‌دهد.

```
C:\Users\linux7>pip freeze
BeautifulSoup==3.2.1  pyparsing==2.0.3
csvkit==0.9.1        pyrtlsdr==0.1.1
dbf==0.94.3          pyserial==2.7
jdcal==1.0           python-dateutil==2
matplotlib==1.3.1   pywin32==219
mechanize==0.2.5     six==1.9.0
numpy==1.8.1         SQLAlchemy==1.0.8
openpyxl==2.2.0b1    vboxapi==1.0
py2exe==0.6.9        virtualenv==13.1.1
PyInstaller==2.1     visvis==1.9.1
pyopenq1==3.1.1a1    wheel==0.24.0
xlrd==0.9.4
```

## انتخاب یک IDE (Integrated Development Environment)

یکی از پیش‌نیازهای حیاتی به هنگام کار و کد زنی به زبان پایتون، داشتن یک محیط ویرایشگر قوی برای عیب‌یابی، کامل کردن کدها، ارایه توضیح (help) درباره توابع-کلاس‌ها-دستورها و... و نحوه‌ی استفاده از آنها است. محیطی که بتواند به کدنویسی توسعه‌دهندگان سرعت بخشد و مشکلات موجود در کدهای نوشته شده را برای یک برنامه‌نویس پر رنگ کند تا زودتر بر طرف شوند، IDE نام دارد. انتخاب یک IDE خوب باعث می‌شود تا بهره‌وری و کسب تجربه‌ی کدزنی، ساده و آسان‌تر شود. ویژگی‌های یک IDE مناسب به شرح زیر است:

- A graphical code editor with code completion and Intelligence.
- A code explorer for functions and classes.
- Syntax highlighting **AND** Code completion.
- Project management.

- Code templates **AND** Code snippets **AND** Code folding.
- Tools for unit testing and debugging.
- Source control support.
- Auto-Indentation and code formatting
- Find Usages **AND** Go to declaration.
- External documentation view

یکی از بهترین IDEها برای کدزنی به زبان پایتون نرم افزار **PyCharm** نام دارد. این نرم افزار توسط شرکت JetBrains توسعه می یابد و در سه نسخه تجاری، رایگان و آموزشی (رایگان) به بازار عرضه می شود. آخرین نسخه منتشر شده PyCharm-5 است. سیستم عامل های تحت پوشش این نرم افزار Windows، Linux و MAX-OS-X هستند. برای دانلود نسخه 4.5.2 تجاری (کرک شده) آن می توانید به سایت [downloadly.ir](http://downloadly.ir) رجوع کنید:

<http://downloadly.ir/software/programming/jetbrains-pycharm>

برای آشنایی (QuickStart) با محیط برنامه PyCharm می توانید به پیوست-۱ کتاب (بخش سوم، آشنایی و کار با PyCharm) مراجعه نمایید.

**پیشنهاد:** یکی دیگر از نرم افزارهای مفید بهنگام برنامه نویسی به زبان پایتون، Aptana-Studio3 نام دارد. این برنامه از زبان های برنامه نویسی همچون Ruby، Rails و PHP نیز پشتیبانی می کند و در پلتفرم های ویندوز، لینوکس و... نیز قابل اجرا است. آخرین نسخه منتشر شده در این لحظه از تألیف کتاب، ورژن 3.6.1 است.

[www.aptana.com/products/studio3/download.html](http://www.aptana.com/products/studio3/download.html)

<http://github.com/aptana/studio3/releases>

## نحوه های پایه پایتون

با نوشتن کلمه python در پوسته لینوکسی یا ویندوز خود می توانید وارد محیط محاوره ای پایتون شوید.

**Linux:** python

Python 2.7.8 (default, Jan 2 2013, 16:53:07 )

[GCC 4.7.2] on linux2

Type "help", "copyright", "credits" or "license" for more information.



&gt;&gt;&gt;

**Windows:** C:\Users\Alimp5-User> python

Python 2.7.8 (default, Jun 30 2014, 16:03:49) [[MSC v.1500 .....

Type "help", "copyright", "credits" or "license" for .....

&gt;&gt;&gt;

**تعریف >>> در پایتون:** بدین معناست که شما در محیط محاوره‌ای پایتون قرار دارید و با وارد کردن دستور(های) خود و فشردن کلید Enter کیبورد، دستور(های) شما در همان لحظه بررسی و اجرا می‌گردد.

تقریباً تمام عبارت‌های پایتون در یک خط تایپ می‌شوند؛ اما با قرار دادن پرانتز( )، براکت [ ] و بریس { } باز و بسته، پیش و پس از عبارات، می‌توانید عبارت‌هایتان را در خط‌های بعدی نیز بنویسید (ایجاد "بلوک کد"). برای اطلاعات بیشتر درباره علائم نشانه‌گذاری می‌توانید به لینک زیر مراجعه نمایید:

<http://en.wikipedia.org/wiki/Punctuation>

```
>>> #correct python expr:
...
>>> (2+
...
... 4)
6
>>>
```

```
>>> 2+
File "<stdin>", line 1
  2+
  ^
SyntaxError: invalid syntax
>>>
>>>
```

**تعریف . . . در پایتون:** بدین معناست که در یک "بلوک کد" قرار دارید. در فصل‌های بعدی درباره توابع صحبت خواهیم کرد؛ اما برای درک بیشتر "بلوک کد" به مثال زیر توجه کنید:

```
C:\Users\linux7>python
Python 2.7.8 (default, Jun 30 2014, 16:03:49) [MSC v.1!
Type "help", "copyright", "credits" or "license" for m
>>>
>>> def azimzadeh( ):
...     print "Ali is writing a book about python.\n"
...     print "Hi World!."
...     w=2+5
...     print w
...
>>> azimzadeh( )
Ali is writing a book about python.
Hi World!.
7
>>> ■
```

در شکل بالا می‌بینید که پس از خط نخست ( def azimzadeh( ) ). چهار خط بعدی، دقیقاً زیر کلید واژه‌ی def نوشته نشده‌اند و تورفتگی به سمت راست دارند؛ در واقع با این کار یک "بلوک کد" تعریف کردیم.

- **شناسه‌گرها:** از شناسه‌ها در پایتون برای تشخیص یک متغیر، تابع، کلاس، ماژول و... استفاده می‌شود. حرف نخست شناسه‌ها باید با حروف A تا Z یا a تا z یا یک آندراسکور ( \_ ) آغاز شود. مفسر پایتون اجازه نمی‌دهد از کارکترهایی مانند @, \$, % در تعریف شناسه‌ها استفاده نمایید.

```
>>> re@za = 2
File "<stdin>", line 1
    re@za = 2
      ^
SyntaxError: invalid syntax
>>> 2var= 'salam'
File "<stdin>", line 1
    2var= 'salam'
      ^
SyntaxError: invalid syntax
>>> var2='salam'
>>> reza_asghar='karaj'
>>>
```

**نکته:** به‌خاطر بسپارید که پایتون یک زبان حساس به حروف کوچک و بزرگ است؛ یعنی دو شناسه **azim** و **Azim** یکسان نیستند و با یکدیگر تفاوت دارند.

**نکته:** کلمات (کلید واژه‌ها) جدول زیر، کلمات رزرو شده زبان پایتون هستند و نباید در تعریف متغیرها و ثابت‌ها از آنها استفاده کنید:

Keywords in Python programming language				
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	

**خط‌ها و تورفتگی‌ها:** یکی از موضوعات جالب این زبان، عدم وجود بریس‌ها ({} ) برای تعیین بلوک‌های-کد است. برای تعریف یک بلوک-کد باید از کلیدهای Space یا Tab کیبورد استفاده کنید. **یادآوری:** اگر به‌خاطر داشته باشید، در زبان C/C++ از بریس‌ها در تعیین بازه بلوک-کدها استفاده می‌شود.

```

>>> x=2
>>> if x==2:
...     print "true"
... else:
...     print "false"
true
>>>
=
>>> x=2
>>> if x==2:
...     print "True"
... else:
...     print "False"
True
>>>

```

بلوک‌ها در شکل بالا به‌درستی تعریف شده‌اند و بدون هیچ خطایی، خروجی‌ها نمایش داده شدند. اما چون در شکل زیر، فرو رفتگی‌های بلوک دوم به‌درستی تعریف نشده‌اند، اینکار باعث شده تا یک پیام خطا تولید و نمایش داده شود:

```

>>>
>>> if True:
...     print "true."
...     print "answer."
... else:
...     print "false."
...     print "answer."
File "<stdin>", line 6
    print "answer."
    ^
IndentationError: unexpected indent
>>>

```

به عنوان یک تمرین ساده، در شکل زیر سعی نمایید بلوک‌های-کد را تشخیص دهید (با توجه به فضاهای خالی).

نکته: به‌خاطر بسپارید که بلوک‌های-کد می‌توانند "تو در تو" نیز باشند. برای نمونه: استفاده از دستور شرطی if در دل دستور while.

```
#!/usr/bin/python
import sys

try:
    # open file stream
    file = open(file_name, "w")
except IOError:
    print "There was an error writing to", file_name
    sys.exit()
print "Enter ", file_name,
print "" When finished"
while file_text != file_finish:
    file_text = raw_input("Enter text: ")
    if file_text == file_finish:
        # close the file
        file.close()
        break
    file.write(file_text)
    file.write("\n")
file.close()
file_name = raw_input("Enter filename: ")
if len(file_name) == 0:
    print "Next time please enter something"
    sys.exit()
try:
    file = open(file_name, "r")
except IOError:
    print "There was an error reading file"
    sys.exit()
file_text = file.read()
file.close()
print file_text
```

- **کوئیشن‌ها:** پایتون از تک-کوئیشن (!)، دابل-کوئیشن (") و سه-کوئیشن (""" یا """) به منظور تفکیک کردن رشته‌ها استفاده می‌کند.

```
>>> word = 'word'
>>> sentence = "This is a sentence"
>>> paragraph = """This is a paragraph. It is
... made up of multiple lines and sentences."""
>>> print paragraph
This is a paragraph. It is
made up of multiple lines and sentences.
```

- **توضیحات:** علامت # (Hash-sign) در زبان پایتون برای نوشتن توضیحات (comments) استفاده می‌شود. هر چیزی که پس از این علامت در برنامه نوشته شود، مفسر پایتون از خواندن آن صرف نظر می‌کند. برای نمونه:

```
>>> # First comment
>>> print "Hello, Python!" ; # second comment
>>> print "Ali.Azimzadeh@gmail.com" ; # third comment
Hello, Python!
Ali.Azimzadeh@gmail.com
>>> name = "python is interesting" #This is again comment
>>> #This is a comment.
>>> #This is a comment, too.
>>> # This is a comment, too.
>>> # i said that already.
```

- **دریافت ورودی:** برای دریافت مقدار اولیه از کاربران در محیط prompt، از دو تابع Built-in پایتون با نام‌های `raw_input()` و `input()` استفاده می‌شود.

```
>>> input ("Enter your Age: ")
>>> age = input ("Enter your Age: "); print age
>>> raw_input ("Enter your Name:")
>>> name = raw_input ("Enter your Name:"); print name
```

در پایتون-۳، تابع `input` حذف شده است. دلیل حذف آن به‌خاطر آسیب‌پذیر بودن به مقادیر ورودی است. چون این تابع مقدار وارد شده در `prompt` را به عنوان یک عبارت پایتون (python expression) در نظر می‌گیرد (`eval(input())`). فرض کنید کاربر عبارت `*rm -rf` را به عنوان ورودی به برنامه‌ی پایتون در حال اجرا در یک سیستم‌عامل لینوکسی بدهد. ❗❗❗

**پیشنهاد:** از توابع `eval` و `exec` استفاده نکنید؛ چون حفره‌های امنیتی در خودشان دارند.

**نکته:** کارکتر سیمی-کالین (؛) در پایتون اجازه می‌دهد تا چندین دستور را در یک خط بنویسید.

```
>>> import sys; x = 'AzimZadeh'; sys.stdout.write(x + '\n')
AzimZadeh
```

- دریافت آرگومان از خط فرمان: ماژول sys (sys.argv), دسترسی به تمام آرگومان‌های وارد شده در خط فرمان را فراهم می‌کند.

```
#!/usr/bin/python
import sys
print 'Number of arguments:', len(sys.argv) , 'arguments.'
print 'Argument List:', str(sys.argv)
```

**Linux:** python test.py arg1 arg2 arg3

Number of arguments: 4 arguments.

Argument List: ['test.py', 'arg1', 'arg2', 'arg3']

**Linux:** python test.py ali reza azimzadeh milani

Number of arguments: 5 arguments.

Argument List: ['test.py', 'ali', 'reza', 'azimzadeh ', 'milani']

همیشه دقت داشته باشید که نخستین آرگومان ارسالی به برنامه، نام اسکریپت شما است و مفسر پایتون آن را به عنوان آرگومان نیز می‌شمارد.

در ادامه قصد داریم با تشریح یک مثال، نحوه استفاده از ماژول پیشرفته getopt را که در تجزیه آرگومان‌ها و پارامترهای ورودی برنامه‌ها، کاربرد زیادی دارند توضیح دهیم:

```
#!/usr/bin/python
import sys, getopt
def main (argv):
    inputfile = ' '
    outputfile = ' '
    try:
        opts,args=getopt.getopt(argv,"hi:o:" ,["help","ifile=","ofile="])
    except getopt.GetoptError:
        print 'In Except: test.py -i <inputfile> -o <outputfile>'
        sys.exit(2)
    for opt, arg in opts:
        if opt in ("-h" , "--help"):
            print 'test.py -i <inputfile> -o <outputfile>'
            sys.exit()
        elif opt in ("-i", "--ifile"):
            inputfile = arg
        elif opt in ("-o", "--ofile"):
            outputfile = arg
    print 'Input file is "', inputfile
    print 'Output file is "', outputfile
if __name__ == "__main__":
    main (sys.argv[1:])
```

pic by Azimzadeh  
alimp5

اما نحوه (syntax) متد getopt در پایتون چگونه است؟: در ادامه نحو ساده متد getopt را می‌بینید:

getopt.getopt (args, options [, long\_options])

args: لیستی از آگومان‌ها است که باید توسط متد getopt تجزیه شود.

options: این گزینه رشته‌ای از حروف است که اسکرپیت قصد تشخیص آنها را دارد. گزینه‌هایی که نیاز به یک آرگومان ورودی دارند، باید جلوی گزینه آن حرف، یک کالن(:) نوشته شود.

long\_options: این پارامتر اختیاری است و اگر تعیین شود، باید یک لیست از رشته‌ها باشد. پارامترها در بخش گزینه‌های طولانی در صورتیکه نیاز به یک آرگومان ورودی داشته باشند، باید جلوی حرفی که در بخش گزینه طولانی است یک کارکتر برابر(=) نوشته شود.

نگران نباشید، شاید درک متن بالا در ابتدا کمی سخت باشد، پس به خط زیر دقت کنید:

opts, args = getopt.getopt(argv, "hs:d", ["help", "sitename="])

فرض کنید می‌خواهیم برنامه‌ای بنویسیم که اگر کاربر -h یا --help را وارد کرد، نحوه‌ی استفاده (Manual guide or Usage) از برنامه نمایش داده شود؛ چون پارامتر -h یا --help هیچگونه آرگومان دیگری دریافت نمی‌کنند، جلوی کارکتر h ("hg:d") علامت کالن(:) را و جلوی حرف help علامت مساوی(=) را نمی‌گذاریم. اگر تمایل داشتید که برنامه تنها پارامتر -h دریافت کند، می‌توانید حرف help را در بخش long\_options (گزینه‌های طولانی) ننویسید.

نکته: دقت داشته باشید که اصلاً نیازی به نوشتن علامت دش(-) در getopt ندارید؛ چون در پشت پرده، این دش‌ها (dash) به‌طور خودکار تشخیص داده می‌شوند.

```
root@kali:~# python tst1.py -h
test.py -i <inputfile> -o <outputfile>
root@kali:~# python tst1.py --help
test.py -i <inputfile> -o <outputfile>
root@kali:~# python tst1.py -i usernames.txt --ofile myResults.txt
Input file is " usernames.txt
Output file is " myResults.txt
root@kali:~#
```

نگران توابع، دستورها و ماژول‌هایی که در شکل صفحه‌ی پیش استفاده شدند نباشید، چون درباره هر یک مفصل صحبت خواهیم کرد. تنها هدف ما در این بخش، درک متد getopt بود که امیدوارم مؤلف توانسته باشد این مطلب به ظاهر پیچیده را به خوبی توضیح دهد.

- **انواع متغیرها (داده‌ها):** برخلاف زبان‌هایی همچون C، C++ یا C#، مفسر پایتون با توجه به نوع-داده‌ی (data type) انتساب داده شده به یک متغیر، قدری حافظه به آن متغیر برای نگهداری مقدارش تخصیص می‌دهد. پایتون با استفاده از تکنیک پویای Garbage-Collection که به‌طور پیش

گزیده فعال است، اقدام به حذف اشیاء (built-in types or class instances) غیرضروری می‌کند تا فضای بیهوده پر شده‌ی حافظه را آزاد نماید (برای کسب اطلاعات بیشتر درباره اشیاء، به فصل ششم مراجعه نمایید). در پایتون برای تعریف نوع-داده‌ها نیازی به استفاده از کلید واژه‌هایی مانند string str1، char op، int w، float x و... نیست.

عملوندی که سمت چپ علامت = قرار دارد نام متغیر است و عملوندی که سمت راست علامت = قرار می‌گیرد، مقداری است که در متغیر ذخیره می‌شود. در زبان پایتون بیان جمله‌ی پیش نادرست است. اما اکنون موقتاً این موضوع را به‌خاطر بسپارید تا با تعریف درست آن در فصل ششم (کلاس‌ها) آشنا شوید.

```
>>> counter = 100 # An integer assignment
>>> miles = 1000.0 # A floating point
>>> name = "John" # A string
>>> print counter
100
>>> print miles
1000.0
>>> print name
John
>>> type(counter); type(miles); type(name)
<type 'int'>
<type 'float'>
<type 'str'>
>>>
```

<pre>&gt;&gt;&gt; x = 2.4 + 3.2 &gt;&gt;&gt; type(x); x &lt;type 'float'&gt; 5.6 &gt;&gt;&gt; w = 2.0 + 5.0 &gt;&gt;&gt; type(w); print w &lt;type 'float'&gt; 7.0 &gt;&gt;&gt; z = 5 / 2 &gt;&gt;&gt; type(z); print z &lt;type 'int'&gt; 2 &gt;&gt;&gt; q = 5.0 / 2 &gt;&gt;&gt; type(q); print q &lt;type 'float'&gt; 2.5 &gt;&gt;&gt;</pre>	<pre>&gt;&gt;&gt; f = 5 / 2.0 &gt;&gt;&gt; type(f); print f &lt;type 'float'&gt; 2.5 &gt;&gt;&gt; 17 &gt; 6 True &gt;&gt;&gt; 15 &lt; 2 False &gt;&gt;&gt; (2/3) &lt; (3/5) False &gt;&gt;&gt; True or False True &gt;&gt;&gt; True and False False &gt;&gt;&gt;</pre>
---	--

**نکته:** برای تشخیص نوع-داده‌ای متغیرها می‌توانید از تابع `type()` استفاده نمایید. روش‌های دیگری برای انتساب دهی همزمان مقادیر به متغیرها نیز وجود دارد؛ اگر قصد اختصاص دادن یک مقدار به چندین متغیر دارید، می‌توانید به‌صورت زیر این کار را انجام دهید:

```
>>> a = b = c = 2
```

همچنین می‌توانید چندین شیء را به چندین متغیر با کمک علامت کاما (,) انتساب دهید:



```
>>> a, b, c = 1, 5, 'azimzadeh'
```

با اجرای دستور بالا در محیط پایتون، عدد 1 به متغیر *a*، عدد 5 به متغیر *b* و رشته *azimzadeh* به متغیر *c* انتساب داده می‌شود.

### انواع نوع-داده‌های استاندارد در پایتون

در زبان پایتون ۵ نوع-داده‌ی استاندارد وجود دارد که در فصل‌های بعدی درباره هر یک از آنها صحبت خواهیم کرد: (۱) Numbers (۲) String (۳) List (۴) Tuple (۵) Dictionary

### • انواع عملگرها: عملگرهایی که توسط پایتون پشتیبانی می‌شوند در ۷ گروه قرار می‌گیرند:

- ◀ عملگرهای حسابی (مانند +, -, \*, /, %, \*\*, //)
  - ◀ عملگرهای مقایسه‌ای-رابطه‌ای (مانند ==, !=, <, >, <=, >=)
  - ◀ عملگرهای انتساب دهی (مانند =, +=, -=, \*=, /=, %=, \*\*=, //=)
  - ◀ عملگرهای منطقی (مانند and, or, not)
  - ◀ عملگرهای بیتی (مانند &, |, ^, ~, <<, >>)
  - ◀ عملگرهای عضویت (مانند in و not in)
  - ◀ عملگرهای هویت (مانند is و is not)
- ◀ **عملگرهای حسابی:** فرض کنید مقدار متغیر *a* برابر ۲ و مقدار متغیر *b* برابر ۶ است.

عملگر	توضیح	نمونه
+	جمع مقادیر دو متغیر	$a + b = 8$
-	تفریق مقادیر دو متغیر از هم	$a - b = -4$
*	ضرب مقادیر دو متغیر در هم	$a * b = 12$
/	تقسیم مقادیر دو متغیر بر دیگری	$b / a = 3$
%	تقسیم مقادیر دو متغیر بر دیگری و بازگرداندن مقدار باقی مانده	$b \% a = 0$
**	به توان رساندن مقدار متغیر <i>a</i>	$a ** b = 64$

عملگرهای مقایسه‌ای: فرض کنید مقدار متغیر  $a$  برابر ۲ و مقدار متغیر  $b$  برابر ۶ است.

عملگر	توضیح	نمونه
==	اگر مقدار متغیرها برابر هم باشند، خروجی True است.	$a == b$
!=	اگر مقدار متغیرها برابر هم نباشند، خروجی True است.	$a != b$
<>	اگر مقدار متغیرها برابر هم نباشند، خروجی True است. (مشابه !=)	$a <> b$
>	اگر مقدار متغیر سمت چپ از مقدار متغیر سمت راست بزرگتر باشد، خروجی True است.	$a > b$
<	اگر مقدار متغیر سمت چپ از مقدار متغیر سمت راست کوچکتر باشد، خروجی True است.	$a < b$
>=	اگر مقدار متغیر سمت چپ از مقدار متغیر سمت راست بزرگتر یا مساوی آن باشد، خروجی True است.	$a >= b$
<=	اگر مقدار متغیر سمت راست از مقدار متغیر سمت چپ بزرگتر یا مساوی آن باشد، خروجی True است.	$a <= b$

عملگرهای انتساب‌دهی: فرض کنید مقدار متغیر  $a$  برابر ۲ و مقدار متغیر  $b$  برابر ۶ است.

عملگر	توضیح	نمونه
=	انتساب مقدار سمت راست عملگر =، به متغیر سمت چپ	$c = a + b$ $c = 8$
+=	انتساب مقدار حاصل از جمع سمت راست عملگر =، به متغیر سمت چپ	$c += a$ $c = c + a$ $c = 10$

عملگر	توضیح	نمونه
-=	انتساب مقدار حاصل از تفریق سمت راست عملگر=، به متغیر سمت چپ	$c -= a$ $c = c - a$ $c = 6$
*=	انتساب مقدار حاصل از ضرب سمت راست عملگر=، به متغیر سمت چپ	$c *= a$ $c = c * a$ $c = 16$
/=	انتساب مقدار حاصل از تقسیم سمت راست عملگر=، به متغیر سمت چپ	$c /= a$ $c = c / a$ $c = 4$
%=	انتساب مقدار حاصل از باقی مانده‌ی تقسیم سمت راست عملگر=، به متغیر سمت چپ	$c \% = a$ $c = c \% a$ $c = 0$
**=	انتساب مقدار حاصل از به توان رسیدن سمت راست عملگر=، به متغیر سمت چپ	$c ** = a$ $c = c ** a$ $c = 64$

عملگرهای منطقی: فرض کنید مقدار متغیر a برابر ۲ و مقدار متغیر b برابر ۶ است.

عملگر	توضیح	نمونه
and	اگر مقدار هر دو متغیر صحیح باشد، شرط true می‌شود	a and b
or	اگر مقدار یکی از عملوندها صحیح باشد، شرط true می‌شود	a or b
not	برای معکوس کردن نتیجه استفاده می‌شود	not (a and b)