مرجع کامل
برنامه‌نویسی پایتون

تألیف:

دکتر جواد وحیدی (عضو هیات علمی دانشگاه علم و صنعت ایران)

دکتر رمضان عباس نژادورزی

فن‌آوری نوین

سرشناسه	وحیدی، جواد، ۱۳٤۹ - Javad ,Vahidi
عنوان و نام پدیدآور	مرجع کامل برنامه‌نویسی پایتون/ تالیف جواد وحیدی، رمضان عباس‌نژادورزی.
مشخصات نشر	بابل: فناوری نوین، ۱۳۹۸.
مشخصات ظاهری	[۷۳۸] ص.: مصور.
شابک	٤-٤٠-٧٢٧٢٤٠-٦٠٠-٩٧٨ : ۱۲۰۰۰۰ ریال
وضعیت فهرست نویسی	فیپا
یادداشت	کتابنامه: ص.[۷۳۸].
موضوع	پایتون (زبان برنامه‌نویسی کامپیوتر)
موضوع	Python (Computer program language)
شناسه افزوده	عباس‌نژاد ورزی، رمضان، ۱۳٤۸ -
رده‌بندی کنگره	QA۷۶/۷۳
رده‌بندی دیویی	۰۰۵/۱۳۱
شماره کتابشناسی ملی	۶۰۷۴۹۰۱

www.fanavarienovin.net

فن‌آوری نوین بابل، کد پستی ٤٧١٦٧-٧٣٤٤٨ تلفن: ۳۲۲۵٦٦۸۷-۰۱۱

مرجع کامل برنامه‌نویسی پایتون

تألیف: جواد وحیدی- رمضان عباس نژادورزی

نوبت چاپ: چاپ اول

سال چاپ: زمستان ۱۳۹۸

شمارگان: ۲۰۰

قیمت: ۱۲۰۰۰۰ تومان

نام چاپخانه و صحافی:

شابک: ٤-٤٠-٧٢٧٢-٦٠٠-٩٧٨

نشانی ناشر: بابل، چهارراه نواب، کاظم بیگی، جنب مسجد منصور کاظم بیگی، طبقه اول

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

تهران، خ اردیبهشت، نبش وحید نظری، پلاک ۱۴۲ تلفکس: ۶۶۴۰۰۲۲۰-۶۶۴۰۰۱۴۴

فهرست مطالب

مقدمه

هرگز به آن‌قدر که می‌پری قانع نباش .
هرگز نگو بیشتر از این ممکن نیست.
دائم از خودت عبور کن!
شاگردی که کمتر از معلمش بداند،
دنیا را به عقب می‌راند،
شاگردی که به‌قدر معلمش بداند دنیا را متوقف می‌کند.
این تویی که پیش می‌رانی...

عصر حاضر ازنظر بسیاری از روشنفکران انقلاب ارتباطات و اطلاعات نام‌گرفته است. درواقع کلیه کارهای روزمره‌ی انسان با سیستم‌های ارتباطی نوین و فناوری اطلاعات گره‌خورده است. لذا اهمیت علم کامپیوتر به‌عنوان نمود اصلی فناوری اطلاعات و ارتباطات بر کسی پوشیده نیست. در دنیای مدرن امروز تسلط و آشنایی با نحوه کار با سیستم‌های کامپیوتری به‌عنوان یک سطح سواد مطلوب در جامعه شناخته می‌شود که اگر کسی آن را نداشته باشد باید بهای عدم آشنایی با دنیای کامپیوتر در زندگی روزمره خویش با مراجعه به کافی‌نت‌ها، درخواست‌های الکترونیکی و موارد دیگر بپردازد.
بدون شک، برنامه‌نویسی یکی از مهم‌ترین مهارت‌هایی است که امروزه نه‌تنها برای فارغ‌التحصیلان و دانشجویان رشته مهندسی کامپیوتر، بلکه برای سایر رشته‌ها و زمینه‌ها نیز به شکل ضروری، موردنیاز است. هرساله، بر تعداد موقعیت‌های شغلی که مرتبط با برنامه‌نویسی هستند و یا پیش‌نیاز اصلی آن‌ها برنامه‌نویسی است، افزوده می‌شود. از طرفی، بخشی جدانشدنی از فرآیند تحقیق و پژوهش در اکثر رشته‌های دانشگاهی امروزی، کار با کامپیوتر و توانایی پیاده‌سازی ایده‌ها و الگوریتم‌ها در قالب برنامه‌های کامپیوتری است. قطعاً در این مسیر، دانشجویان نیازمند یادگیری و کسب مهارت‌های برنامه‌نویسی هستند.

پایتون یک زبان برنامه‌نویسی همه‌منظوره، **سطح بالا**، **شیءگرا** و **مفسری** است که توسط خودو فان روسوم در سال ۱۹۹۱ در کشور هلند طراحی شد. فلسفه ایجاد آن تأکید بر دو هدف

اصلی خوانایی بالای برنامه‌های نوشته‌شده و کوتاهی و بازدهی نسبی بالای آن است. کلمات کلیدی و اصلی این زبان به‌صورت حداقلی تهیه‌شده‌اند و در مقابل کتابخانه‌هایی که در اختیار کاربر است بسیار وسیع هستند.

علی‌رغم استقبال گسترده دانش‌پژوهان عرصه کامپیوتر از زبان پایتون ، در زمینه آموزش این زبان قدرتمند کتاب‌های زیادی به زبان فارسی منتشرنشده است که کتاب حاضر برای پر کردن خلاء موجود در این زمینه نوشته‌شده و توسط انتشارات فن‌آوری نوین به زیور چاپ آراسته‌شده است. امید است اثر حاضر همانند سایر آثار مؤلفین مورد استقبال قرار گیرد.

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.

مؤلفین

fanavarienovin@gmail.com

فصل ۱ آشنایی با زبان پایتون

۱–۱. مقدمه‌ای بر زبان برنامه‌نویسی پایتون

پایتون یکی از معدود زبان‌های برنامه‌نویسی است که می‌توان ادعا کرد ساختاری ساده و قدرتمند دارد، ازاین‌رو، یادگیری این زبان همواره به افراد مبتدی که شاید هیچ تجربه‌ای در برنامه‌نویسی نداشته باشند، توصیه می‌شود و از طرف دیگر، استفاده از این زبان برای حل مسائل مختلف و پیچیده انتخاب اول بسیاری از برنامه نویسان حرفه‌ای بوده است.

بر اساس رتبه‌بندی سایت Tiobe، زبان برنامه‌نویسی Python در سپتامبر سال ۲۰۱۵ با سه پله صعود نسبت به زمان مشابه در سال قبل در جایگاه پنجم قرارگرفته است که نشان‌دهنده‌ی رشد محبوبیت این زبان در میان برنامه نویسان سراسر دنیا است.

همان‌طور که می‌دانید هر زبان برنامه‌نویسی ویژگی‌ها و قابلیت‌های خاص خود را دارد که آن را از سایر زبان‌ها متمایز می‌سازد و علت شکل‌گیری زبان‌های مختلف نیز پاسخگویی به نیازهای متفاوت و متنوع کاربران با استفاده از همین قابلیت‌های متمایز است. به همین دلیل، پیش از شروع به یادگیری هر زبان ابتدا باید نیازها و هدف خود را از یادگیری آن زبان در کنار قابلیت‌هایش قرار دهیم و در صورت تطبیق آن‌ها باهم، قدم درراه یادگیری بگذاریم. بنابراین، برای آشنایی بیش‌تر با زبان پایتون، در ادامه به معرفی برخی از ویژگی‌ها و قابلیت‌های آن می‌پردازیم:

۱. **سادگی و صراحت[1]:** پایتون یک‌زبان ساده و کمینه‌گرا است. وقتی نگاهی به سورس کد یک برنامه ی نوشته‌شده به زبان پایتون بی‌اندازیم، احساس می‌کنیم که با یک متن انگلیسی صریح مواجه هستیم. شاید بتوان گفت این بزرگ‌ترین نقطه‌ی قوت پایتون است که به‌جای درگیر کردن برنامه‌نویس به جزئیات زبان به او اجازه می‌دهد تا روی حل مسئله تمرکز داشته باشد. همین موضوع سرعت کد نویسی و خوانایی این زبان را هم افزایش داده است.

۱. **منحنی یادگیری کم شیب[2]:** قطعاً عامل اصلی این موضوع که یادگیری پایتون به‌عنوان قدم اول به مشتاقان برنامه‌نویسی و حتی کودکان توصیه می‌شود سینتکس فوق‌العاده ساده‌ی آن است. همان‌طور که گفتیم صراحت زبان پایتون نه‌تنها خوانایی آن را افزایش داده است، بلکه با حذف پیچیدگی‌ها سهولت یادگیری آن را نیز بیش‌تر کرده است.

۲. **رایگان و متن‌باز بودن[3]:** توزیع‌های مختلف زبان برنامه‌نویسی پایتون کاملاً رایگان بوده و هر برنامه‌نویس می‌تواند سورس کد آن را بخواند، آن را تغییر دهد، و در برنامه‌های خود از اسکریپت‌های آن استفاده کند.

۳. **سطح بالا بودن[4]:** پایتون ازجمله زبان‌های قدرتمند سطح بالا است که برنامه‌نویس را درگیر جزئیات سطح پایین مثل مدیریت حافظه یا کار با ثبات‌ها (Registers) و غیره نمی‌کند.

---

۱. **Simplicity** ۲. **Low Learning Curve** ۳. Free & Open Source ۴. High-level

٤. **قابل‌حمل بودن:** ماهیت متن‌باز پایتون موجب شده است که این زبان با پلتفرم‌های مختلف سازگار باشد. بنا بر اعلام رسمی سایت پایتون، در حال حاضر این زبان روی ۲۱ پلتفرم ازجمله Windows، GNU/Linux، Macintosh، Solaris، Android، iOS، و ... کار می‌کند و برنامه‌های نوشته‌شده به این زبان بدون نیاز به تغییر یا با تغییرات بسیار جزئی روی تمام پلتفرم‌ها اجرا می‌شوند.

زبانی مفسری: برخلاف زبان‌های کامپایلری مانند C یا جاوا، زبان برنامه‌نویسی پایتون یک‌زبان مفسری است و سورس کد برنامه‌های نوشته‌شده به این زبان با استفاده از یک مفسر اجرا می‌شود که همین موضوع قابل‌حمل بودن آن را افزایش می‌دهد.

شیءگرایی: پایتون در مقایسه با زبان‌هایی مانند جاوا یا C++، روش قدرتمندتر و ساده‌تری را برای اجرا برنامه‌های شیءگرا به کار می‌گیرد.

توسعه‌پذیری: یکی از مشکلات زبان مفسری پایتون سرعت‌پایین اجرا در مقایسه با زبان‌های کامپایلری مانند C یا جاوا است. حال اگر بخواهید قطعه‌ای از کدها سریع‌تر اجرا شود یا اگر بخواهید بخشی از الگوریتم برنامه‌ی خود را پنهان کنید می‌توانید آن بخش را به زبان C، C++ یا جاوا بنویسید و آن را در میان کدهای پایتون برنامه‌ی خود قرار دهید.

تعبیه پذیری: علاوه بر این‌که می‌توان کدهای زبان‌های دیگر را در برنامه‌های نوشته‌شده به زبان پایتون قرار داد، می‌توان قطعه کدهایی را به زبان پایتون نوشت و در سورس کد برنامه‌های C، C++ یا جاوا نشاند و به این ترتیب قابلیت‌های اسکریپتی به سورس کد مدنظر اضافه نمود.

کتابخانه‌ی گسترده: پایتون از یک کتابخانه‌ی استاندارد غنی بهره می‌برد و در کنار این کتابخانه‌ی وسیع، کتابخانه‌های سایر توسعه‌دهندگان نیز به‌سرعت درحال‌توسعه می‌باشند که درمجموع ابزارهای مناسبی را برای ایجاد اسناد، رابط‌های گرافیکی کاربر (GUI)، مرورگرهای وب، رمزنگاری، هوش مصنوعی، پست الکترونیکی، بازی‌سازی، داده‌کاوی، ایجاد و مدیریت وب‌سایت، و بسیاری کاربردهای دیگر در اختیار برنامه نویسان قرار می‌دهد.

۲. **همه‌منظوره بودن:** پایتون یک‌زبان برنامه‌نویسی با طیف گسترده‌ای از کاربردها است که در حوزه‌های مختلف و متنوع کاربرد داشته است که ازجمله مهم‌ترین کاربردهای آن در طی سالیان گذشته می توان به موارد زیر اشاره کرد:

۱. موتور جستجوگر گوگل و موتور گرافیکی یوتیوب
۲. ساخت برنامه‌های کاربردی علمی در سازمان فضایی ناسا، Fermilab
۳. بخشی از سرویس ایمیل یاهو
۴. تست سخت‌افزار در IBM، Intel و Cisco
۵. ابزارهای نصب لینوکس در نسخه‌ی Redhat
۶. سرویس ابری Dropbox

۷. و بسیاری کاربردهای دیگر نظیر طراحی سایت‌های دینامیک، تولید نرم‌افزارهای دسکتاپ، انیمیشن‌سازی، بازی‌سازی، شبکه، امنیت، پایگاه داده، داده‌کاوی، ساخت برنامه‌های محاسباتی و کاربردی در رشته‌های مختلف نظیر ریاضی، فیزیک، آمار، زیست و غیره.

درنهایت می‌توان گفت که پایتون ابزاری مهیج و قدرتمند در اختیار برنامه نویسان است که کار با آن ساده و سرگرم‌کننده می‌باشد و تسلط بر آن کاربران را وارد دنیایی شگفت‌انگیز و بی‌نهایت می‌کند که هرکس می تواند متناسب با توانایی‌هایش از امکانات آن برای حل مسائل خود بهره‌مند شود.

۲–۱. مقایسه‌ی زبان برنامه‌نویسی پایتون با سایر زبان‌های برنامه‌نویسی

با دانستن این‌که مقایسه زبان‌های برنامه‌نویسی با یکدیگر اصلاً کار درستی نیست - چراکه هر زبانی را بهر کاری ساخته‌اند و هر زبان دارای نقاط قوت و ضعف خاص خود است- بااین‌حال، برخی از کاربران همواره دوست دارند تا بدانند زبانی که قرار است فراگیرند در مقایسه با سایر زبان‌های برنامه‌نویسی هم‌رده‌اش، در چه جایگاهی قرار دارد. بنابراین، در ادامه به مقایسه‌ای کوتاه از زبان پایتون با سایر زبان‌های برنامه‌نویسی مطرح دنیا خواهیم پرداخت:

۱- **مزیت‌های زبان پایتون نسبت به زبان سی شارپ:** بسیاری از کارشناسان بر این باورند که شرکت بزرگ مایکروسافت صرفاً زبان برنامه‌نویسی جاوا را کپی کرده و زبانی تحت عنوان سی شارپ را خلق کرده است (مقایسه این دو زبان با یکدیگر خارج از حوزه‌ی این قسمت از آموزش است، اما به‌هرحال هرکدام از این دو زبان دارای نقاط قوت و ضعفی هستند). زبان برنامه‌نویسی پایتون در مقایسه با سی شارپ، از نقاط قوت زیر برخوردار است:

۱. یادگیری آسان‌تر
۲. کد نویسی کم‌تر
۳. متن‌باز و جامعه‌ی توسعه‌ی گسترده
۴. پشتیبانی چندمنظوره بهتر (Multiplatform)
۵. امکان استفاده‌ی راحت از چندین محیط توسعه‌ی نرم‌افزار مختلف
۶. قابلیت توسعه‌ی راحت‌تر با استفاده از زبان‌های C، C++ یا جاوا
۷. پشتیبانی بیش‌تر عملی/ مهندسی

۲- **مزیت‌های زبان پایتون نسبت به زبان جاوا:** سالیان درازی را برنامه نویسان سراسر دنیا منتظر ماندند تا به زبانی دست یابند که یک‌بار کد نویسی کنند و هرکجا که خواستند آن را اجرا کنند تا این‌که زبان جاوا این رؤیای ایشان را به واقعیت مبدل ساخت. جالب است بدانید که در حال حاضر زبان برنامه‌نویسی جاوا به‌عنوان یکی از محبوب‌ترین زبان‌های برنامه‌نویسی دنیا است (حتی محبوب‌تر از پایتون!). به‌هرحال، زبان پایتون دارای یکسری مزیت‌ها نسبت به این زبان است که عبارت‌اند از:

۱. یادگیری به‌مراتب راحت‌تر
۲. کد نویسی به‌مراتب کم‌تر
۳. متغیرهایی باقابلیت ذخیره‌سازی انواع داده‌ها

۴. سرعت توسعه‌ی اپلیکیشن به‌مراتب بیش‌تر از جاوا

۳- **مزیت‌های زبان پایتون نسبت به زبان پرل:** زبان برنامه‌نویسی پرل به‌عنوان زبانی در میان برنامه نویسان شناخته‌شده است که به‌خوبی با پایگاه داده کار می‌کند و داده‌ها را از آن فراخوانی می‌کند، اما درعین‌حال، از این زبان برای ساخت انواع اپلیکیشن‌ها نیز استفاده می‌شود. زبان پایتون در مقایسه با پرل، از نقاط قوت زیر برخوردار است:

۱. یادگیری سریع‌تر
۲. خوانایی بیش‌تر
۳. تعامل بهتر با زبان جاوا
۴. سازگاری بهتر و بیش‌تر با پلتفرم‌های مختلف
۵. امنیت بیش‌تر داده‌ها

اگرچه که در مقایسه‌ی بالا، تقریباً می‌شود گفت که زبان برنامه‌نویسی Python نسبت به زبان‌های Java, Perl و C# از نقاط قوت قابل‌توجهی برخوردار است، اما توجه داشته باشیم که این نیازهای کاری‌تان است که مشخص می‌کند کدام زبان را می‌بایست انتخاب کنید.

چرا زبان برنامه‌نویسی پایتون را انتخاب کنیم؟ زبان‌های برنامه‌نویسی زیادی در حال حاضر وجود دارند که یک برنامه‌نویس مبتدی می‌تواند یکی از آن‌ها را برای شروع انتخاب کند و این در حالی است که هر یک از زبان‌های برنامه‌نویسی دارای نقاط ضعف و قوت خاص خودشان هستند و با آگاهی از همین نقاط ضعف و قوت است که به‌عنوان یک برنامه‌نویس مبتدی می‌توانیم بسته به نیازی که برای یادگیری برنامه‌نویسی داریم دست به انتخاب درستی بزنیم.

معمولاً برنامه نویسان حرفه‌ای سعی می‌کنند که به بیش از یک‌زبان برنامه‌نویسی تسلط پیدا کنند تا متناسب با نقاط قوتی که هر زبان برنامه‌نویسی دارد - مثلاً یک‌زبان برای کار با پایگاه داده سرعت بالایی دارا است و زبان دیگر در تحلیل داده‌ها و غیره خوب است- بتوانند یک اپلیکیشن حرفه‌ای بنویسند.

آگاهی از نقاط ضعف و قوت زبان‌های برنامه‌نویسی - به‌خصوص زبان برنامه‌نویسی پایتون که در این دوره ی آموزشی مدنظر است- به برنامه نویسان کمک می‌کند تا بادید بازتری اقدام به استفاده از آن زبان برنامه‌نویسی نمایند. در ادامه قصد داریم به نکاتی بپردازیم که زبان برنامه‌نویسی پایتون را از سایر زبان‌های برنامه‌نویسی متمایز می‌سازد و این زبان قدرتمند را به گزینه‌ی مناسبی برای طراحی و ساخت اپلیکیشن‌های حرفه‌ای مبدل می‌سازد.

هر زبان برنامه‌نویسی با یک هدف خاص در ذهن توسعه‌دهندگان آن زبان طراحی و توسعه داده‌شده است تا دردی از دردهای سایر برنامه نویسان را دوا کند. در ارتباط با زبان برنامه‌نویسی پایتون، بایستی گفت که هدف اصلی آقای روسوم - خالق زبان برنامه‌نویسی پایتون- این بود که زبانی به دنیا عرضه کند که در یک کلام ساده و کاربردی باشد.

توجه داشته باشیم که اگر با توجه به نیازهای خود در توسعه‌ی اپلیکیشن اقدام به انتخاب زبان نامناسبی کنیم، این نوع انتخاب در آینده می‌تواند بهای گزافی در برداشته باشد که از آن جمله می‌توان به‌صرف وقت زیاد، سرعت توسعه‌ی کم، راندمان اندک و بسیاری مشکلات دیگر اشاره کرد.

۵. نیاز به تعداد خطوط کد کمتر: سورس کد برنامه‌های نوشته‌شده با پایتون در مقایسه با سایر رقبا - مثل زبان‌های C، جاوا و C++- چیزی در حدود ۲ تا ۱۰۰ برابر کمتر است. لذا، این نوید داده می‌شود که زمان کمتری برای نوشتن یک اپلیکیشن با این زبان برنامه‌نویسی قدرتمند نیاز دارید.

۶. خوانایی زیاد: همان‌طور که درک زبانی همچون زبان چینی در مقایسه با مثلاً زبان انگلیسی بسیار دشوارتر است، در مورد زبان‌های برنامه‌نویسی هم دقیقاً قضیه به همین شکل است. به‌عبارت‌دیگر، برخی از زبان‌های برنامه‌نویسی هستند که مطالعه‌ی سورس کد آن‌ها به‌مراتب دشوارتر از سایر زبان‌ها است و خبر خوشحال‌کننده این‌که املای زبان برنامه‌نویسی پایتون - اگر نگوییم راحت‌ترین- یکی از راحت‌ترین املاها در میان زبان‌های برنامه‌نویسی است، چراکه تا حد بسیار زیادی شبیه به زبان انگلیسی است!

۷. یادگیری سریع: پیش‌ازاین هم گفتیم که منحنی یادگیری کم شیب پایتون، آن را به گزینه‌ی خوبی برای مبتدیان مبدل ساخته است. توسعه‌دهندگان اصلی زبان برنامه‌نویسی پایتون همواره این دغدغه را داشته‌اند تا زبانی طراحی کنند که خیلی قوانین عجیب‌وغریب نداشته و استثناءهای آن منجر به یادگیری دشوار این زبان نگردد و همین رویکرد در طراحی زبان برنامه‌نویسی پایتون منجر گردیده تا این زبان جزو زبان‌هایی گردد که یادگیری آن خیلی سریع اتفاق می‌افتد، حتی برای کسانی که هیچ آشنایی با دنیای برنامه‌نویسی ندارند.

۳-۱. از پایتون برای چه اپلیکیشن‌هایی می‌توان استفاده کرد؟

با توضیحات فوق، حال قصد داریم ببینیم که از زبان برنامه‌نویسی پایتون برای چه نوع اپلیکیشن‌هایی می‌توانیم استفاده کنیم که در ادامه، برخی از شاخص‌ترین کاربردهای پایتون را برمی‌شماریم:

۸. برای نمونه‌سازی: گاهی اوقات تیم‌های توسعه‌ی نرم‌افزاری در سراسر دنیا نیاز دارند تا یک نمونه‌ی اولیه از ایده‌ی خاصی که در ذهن دارند ایجاد کنند تا با دید بهتری اقدام به سیاست‌گذاری توسعه‌ی اپلیکیشن خود کنند. در چنین مواقعی، زبان برنامه‌نویسی پایتون به‌منزله‌ی یکی از بهترین گزینه‌ها است چراکه سرعت توسعه ی اپلیکیشن با این زبان نسبت به سایر رقبا به‌مراتب بیش‌تر است و توسعه‌دهندگان زودتر می‌توانند خروجی کدهای خود را ببینند.

۹. ساخت وب اپلیکیشن: اگرچه زبان جاوا اسکریپت به‌عنوان معروف‌ترین زبان برنامه‌نویسی وب اپلیکیشن-ها محسوب می‌گردد، اما در این حوزه زبان پایتون هم حرف‌هایی برای گفتن دارا است و در حال حاضر در جایگاه دوم قرار دارد. واقعیت امر آن است که پایتون قابلیت‌هایی را در اختیار توسعه‌دهندگان قرار می‌دهد که جاوا اسکریپت از برآورده کردن آن‌ها ناتوان است و در کل پایتون این امکان را در اختیار توسعه‌دهندگان قرار می‌دهد تا وب اپلیکیشن‌ها سریع‌تر را طراحی کنند.

۱۰. طراحی اپلیکیشن‌های محاسباتی، علمی و مهندسی: کتابخانه‌های طراحی‌شده برای پایتون این امکان را به توسعه‌دهندگان می‌دهد تا به‌راحتی و به‌سرعت بتوانند اپلیکیشن‌های محاسباتی، علمی و مهندسی طراحی

کنند که ازجمله‌ی مهم‌ترین این کتابخانه‌ها می‌توان به NumPy و SciPy اشاره کرد. با بسته NumPy در فصل ۴ آشنا می‌شویم.

۱۱. کار با XML: زبان XML یکی از زبان‌هایی است که برای ذخیره‌سازی داده‌های تحت وب مورداستفاده قرار می‌گیرد و پایتون هم ارتباط بسیار خوبی با این زبان دارا است. به‌طور مثال، اگر در پروژه‌ای بخواهید از وب‌سرویس استفاده کنید، زبان پایتون گزینه‌ی بسیار مناسبی است.

۱۲. ارتباط با پایگاه داده: امروزه کسب‌وکارهای بسیاری هستند که مبتنی بر داده‌ها هستند و مسلماً نیاز دارند تا داده‌های خود را در بانک اطلاعاتی ذخیره سازند. پایتون به‌سادگی می‌تواند با پایگاه داده‌های مختلف ارتباط برقرار ساخته و به تبادل داده با پایگاه داده - خواندن داده‌ها، ثبت داده‌ها، به‌روزرسانی داده‌ها و حذف آن‌ها بپردازد.

۱۳. طراحی رابط کاربری: پایتون همچون زبان سی شارپ نیست که توسعه‌دهنده با استفاده از نرم‌افزار ویژوال استودیو به‌راحتی بتواند با کشیدن و رها کردن اقدام به طراحی رابط کاربری کند، اما درعین‌حال فریم‌ورک‌های بسیاری برای این زبان طراحی‌شده‌اند که طراحان با استفاده از آن‌ها می‌توانند اقدام به طراحی UI اپلیکیشن‌های خود کنند.

۴-۱. آموزش زبان‌های برنامه‌نویسی

آموزش زبان‌های برنامه‌نویسی مانند زبان‌های طبیعی و زنده دنیا است. یعنی، مانند زبان‌های طبیعی برای آموزش زبان‌های برنامه‌نویسی باید مراحل زیر را انجام داد.

- مانند هر زبان طبیعی ابتدا باید علائم تشکیل‌دهنده زبان را آموخت. به‌عنوان‌مثال، زبان فارسی از علائم الف تا ی، ارقام ۰ تا ۹ و علائم خاص مانند !، ؟ و غیره تشکیل‌شده است. هرکدام از این علائم (نمادها) مفهوم خاصی دارند. زبان پایتون، نیز از علائم a تا z، A تا Z، 0 تا 9، علائم ویژه نظیر {، }، [،]، / تشکیل‌شده است. ابتدا، باید مفاهیم هر یک از علائم را در زبان پایتون آموخت.
- همان‌طور که می‌دانید از ترکیب علائم هر زبان کلمات به وجود می‌آیند. برخی از کلمات دارای معنی و مفهوم هستند و برخی دیگر معنی و مفهوم خاصی ندارند. به‌عنوان‌مثال، کلمات "بابا"، "آب" و "داد"، در زبان فارسی مفهوم خاصی دارند. ولی کلمات "تپتانم" و "بکیاپ" مفهوم خاصی ندارند. به کلماتی که در زبان دارای مفهوم خاص هستند، کلمات کلیدی می‌گویند. در زبان پایتون کلمات کلیدی نظیر for، if، else، while، وجود دارند. در آموزش یک زبان ابتدا باید کلمات کلیدی آن را شناخت و معنی و کاربرد هرکدام از آن‌ها را آموخت.
- با ترکیب کلمات کلیدی به همراه قواعد خاص در هر زبان طبیعی، جمله ایجاد می‌شود (مانند بابا آب داد)، همان‌طور که می‌دانید در زبان فارسی ابتدا فاعل، سپس مفعول و در پایان فعل قرار می‌گیرند. در زبان پایتون نیز برای ایجاد جملات (دستورات) قواعد خاصی وجود دارد. به‌عنوان‌مثال، print برای چاپ اطلاعات به کار می‌رود که به‌صورت زیر استفاده می‌گردد:

```
print (اطلاعات)
```

همان‌طور که می‌دانید، در زبان‌های طبیعی با کنار هم قرار دادن جملات مرتبط به هم پاراگراف ایجاد می‌شود. در زبان‌های برنامه‌نویسی نیز با کنار هم قرار دادن دستورات مرتبط به هم، بلاک ایجاد می‌شود. در پایتون برای ایجاد بلاک باید از تورفتگی استفاده کرد که در ادامه می‌آموزیم.

با کنار هم قرار گرفتن پاراگراف‌ها، صفحات و فصول ایجاد خواهند شد و این روند ادامه می‌یابد تا یک کتاب نوشته شود. در زبان‌های برنامه‌نویسی نیز نوشتن برنامه‌ها هم همین روند را دارد. با کنار هم قرار دادن بلاک‌ها، فایل، و کنار هم قرار گرفتن فایل‌های مرتبط به هم، برنامه را ایجاد می‌شود. بنابراین، در ادامه کتاب به آموزش زبان پایتون با این روش خواهیم پرداخت.

جدول ۱-۱ کلمات کلیدی پایتون.

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	

۱-۴-۱. سطرها

مفسر پایتون و همچنین کاربران، کدهای درون هر ماژول را به‌صورت تعدادی سطر مشاهده می‌کنند. در پایتون دو نوع سطر وجود دارند. ۱. **سطرهای فیزیکی**[1]، سطرهایی هستند که توسط ویرایشگرهای متن شماره‌گذاری می‌شوند و به‌سادگی توسط کاربر قابل‌تشخیص می‌باشند. ۲. **سطرهای منطقی**[2]، برداشت مفسر از اجرای برنامه است. هر سطر بیان‌گر یک دستور پایتون است. به‌عنوان‌مثال، دستورات زیر را در نظر بگیرید:

```
>>> name = "Fanavarienovin.net"
>>> print(name)
```

دستور اول رشته fanavarienovin.net را به متغیر name نسبت می‌دهد و دستور دوم، عبارت fanavarienovin.net را نمایش می‌دهد. در این دستورات، هر سطر منطقی یک سطر فیزیکی در نظر گرفته‌شده است. با اجرای این دستورات خروجی زیر نمایش داده می‌شود:

```
Fanavarienovin.net
```

گاهی اوقات هر سطر فیزیکی می‌تواند شامل چند سطر منطقی باشد. در این حالت، باید بین سطرها، کاراکتر ";" قرار داد. به‌عنوان‌مثال، دستورات زیر را ببینید:

---

۱. Physical Lines ۲. Logical Lines

```
>>> name = "Fanavarienovin.net"; print(name)
```

با اجرای این دستورات نیز خروجی زیر نمایش داده می‌شود:

```
Fanavarienovin.net
```

گاهی اوقات برای خوانایی بیش‌تر بهتر است دستورات یک سطر منطقی در چند سطر فیزیکی تایپ شود؛ به‌عنوان‌مثال، دستورات زیر را مشاهده کنید:

```
>>> message ="Python is a \
good programing language"
>>> print(message)
```

در این مثال، خطوط اول و دوم یک دستور منطقی هستند که در دو سطر آمده‌اند. برای توسعه یک دستور در چند سطر فیزیکی از کاراکتر "\" در انتهای سطر استفاده می‌شود. با اجرای این دستورات خروجی زیر نمایش داده می‌شود:

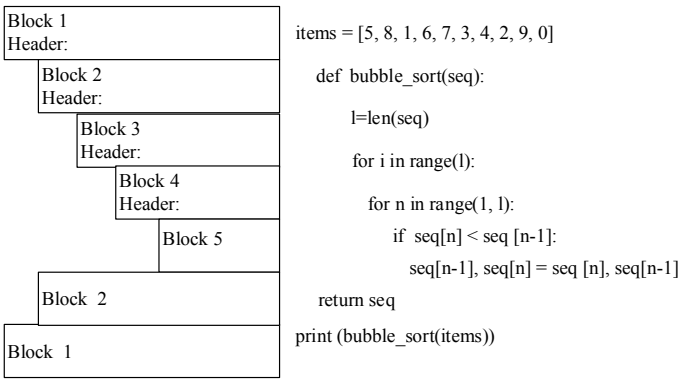
```
Python is a good programing language
```

تعداد کاراکترهای هر سطر فیزیکی نباید از **۷۹** کاراکتر بیش‌تر شود.

سطرهای خالی[1]، برای افزایش خوانایی برنامه به کار می‌روند که شامل فضای خالی (Space یا Tab) هستند و توسط مفسر نادیده گرفته می‌شوند و به بایت کد ترجمه نمی‌گردند.

۲-۴-۱. بلاک بندی

بلاک بندی، یکی از امکاناتی است که برای افزایش خوانایی کد پایتون به کار می‌رود. در زبان پایتون برای ایجاد بلاک از **تورفتگی**[2] سطرها استفاده می‌شود. درواقع، تورفتگی میزان فضای خالی (Space یا Tab) است که در ابتدای هر سطر فیزیکی قرار می‌گیرد. تمام دستورات موجود در یک بلاک باید به یک میزان نسبت به سرآیند خود تورفتگی داشته باشند. یعنی، تعداد فضای خالی تمام دستورات آن بلاک نسبت به سرآیند یکی باشد. شکل ۱ – ۱ نمونه‌ای از این بلاک بندی را نشان می‌دهد.



شکل ۱ – ۱ بلاک بندی در پایتون.

برای ایجاد هر تورفتگی از چهار جای خالی(کلید Space) استفاده کنید.
هرگز برای تورفتگی از کلیدهای Space و Tab باهم استفاده نکنید

---

[1]. Blank Lines [2]. Indentation

۳-۴-۱. دستورات

دستورات، کدهای برنامه هستند که شامل کلمات کلیدی بوده و کار خاصی را اجرا می‌کنند. پایتون دو نوع دستور دارد:

۱. **دستورات ساده**[۱]؛ دستوراتی هستند که فقط در یک سطح منطقی پیاده‌سازی می‌شوند، مانند دستورات import و دستورات فراخوانی توابع.

۲. **دستورات مرکب**[۲]؛ گروهی از دستورات مرتبط به هم هستند که می‌توانند دارای یک بخش (مانند دستور def - برای تعریف تابع به کار می‌رود-) و یا چند بخش(مانند دستورات شرطی if/elif/else) باشند. در دستورات چندبخشی هر بخش شامل یک **سرآیند**[۳] و یک **بدنه** است. هر سرآیند با یک کلمه کلیدی شروع‌شده و با ":" خاتمه می‌یابد. همان‌طوری که قبلاً بیان گردید، بدنه باید پس از سرآیند و با رعایت سطح تورفتگی بیش‌تر تایپ شود تا علاوه بر افزایش خوانایی برنامه، بلاک‌های هر بخش برنامه را نیز مشخص کنند.

۴-۴-۱. شناسه‌ها

همان‌طوری که بیان گردید، برای برنامه‌نویسی در پایتون علاوه بر کلمات کلیدی به متغیرها، توابع، کلاس‌ها، ماژول‌ها و دیگر اشیاء نیاز است. برای شناسایی این اشیاء ابتدا باید آن‌ها را نام‌گذاری کرد. برای نام‌گذاری شناسه باید یکسری قوانین را رعایت نمود که عبارت‌اند از:

- کاراکتر اول نام شناسه می‌تواند یکی از حروف انگلیسی ("A"…"Z" یا "a"…"z") یا کاراکتر _ باشد.
- کاراکترهای بعدی نام شناسه می‌توانند یکی از حروف انگلیسی ("A"…"Z" یا "a"…"z")، کاراکتر _ و اعداد ("0"..."9") باشند.
- از کاراکترهای خاص نظیر " !"، "@"، "$"، "؟"، ; و غیره که در زبان پایتون کاربرد خاصی دارند، نمی‌توان در نام‌گذاری شناسه‌ها استفاده کرد.
- از کاراکتر فضای خالی (Space) نمی‌توان در نام شناسه‌ها استفاده کرد.
- خط تیره '_' برای نام‌گذاری ماژول‌ها مجاز است، ولی بهتر است استفاده نشود.
- زبان پایتون نسبت به حروف بزرگ و کوچک حساس است. یعنی مفسر بین حروف بزرگ و کوچک فرق قائل می‌شود. بنابراین، نام‌های fraction، Fraction، FRaction، FractIon، متفاوت در نظر گرفته می‌شوند.
- در تعداد کاراکترهای نام شناسه محدودیتی وجود ندارد.

در جدول ۲-۱ برخی از نام‌های غیرمجاز و دلیل غیرمجاز بودن آن‌ها را می‌بینید.

شناسه‌هایی که برنامه‌نویس تعریف می‌کند می‌توانند شناسه‌های **خصوصی**[۴] **ماژول** یا **شناسه‌های خصوصی کلاس** باشند. شناسه‌های خصوصی ماژول با یک کاراکتر _ شروع می‌شوند (مانند شناسه‌های name_، x_،

---

۱. Simple Statements ۲. Compound Statements ۳. Header ۴. private

y_). اما شناسه‌های خصوصی کلاس با دو کاراکتر __ شروع می‌شوند (نظیر شناسه‌های name__، type__، و غیره). در ادامه به این شناسه‌ها بیش‌تر می‌پردازیم.

جدول ۲ -۱ برخی از نام‌های غیرمجاز.

نام	دلیل
9nine	استفاده از رقم 9 در ابتدای نام غیرمجاز است.
get.length	استفاده از کاراکتر "." در نام شناسه غیرمجاز است.
Var 2	استفاده از کاراکتر فضای خالی در نام شناسه مجاز نیست.
fan@gmail	استفاده از کاراکتر "@" در نام شناسه غیرمجاز است.
$money	استفاده از کاراکتر "$" در نام شناسه مجاز نیست.

۵-۴-۱. متغیرها

برای نگه‌داری هر ماده‌ی لازم است که یک ظرف استفاده شود. به‌عنوان‌مثال، در خانه برای نگه‌داری مواد غذایی، ظروف مختلفی وجود دارند که هرکدام برای نگه‌داری مواد خاصی به کار می‌روند. ظرف نگه‌داری داده در برنامه‌نویسی **متغیر** نام دارد. متغیر، نامی است که برای یک مکان از حافظه که ممکن است در طول اجرای برنامه مقدار آن تغییر کند، ولی در یک‌لحظه خاص فقط یک مقدار را دارد. برای استفاده از متغیرها باید دو کار انجام شود.

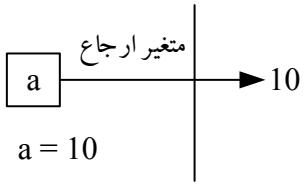
- **نام‌گذاری متغیرها**، برای نام‌گذاری متغیر از قوانین نام‌گذاری شناسه‌ها استفاده می‌کنیم.
- **مقداردهی متغیرها**، پس‌ازاین که نام متغیر را تعیین کردید باید به آن مقدار تخصیص دهید. تخصیص مقدار به متغیرها در پایتون با دستورات انتساب یا خواندن انجام می‌شود. چون، زبان پایتون شیءگرا است، یک متغیر چیزی نیست **به‌جز یک نام** به یک **شی**[۱] **مشخص در حافظه اشاره** می‌کند.

برخلاف زبان‌های C++، C و دیگر زبان‌ها، در پایتون نیازی نیست که یک متغیر تعیین نوع شود، بلکه با تخصیص مقدار، نوع آن تعیین می‌گردد. در هنگام استفاده از متغیر با تخصیص مقدار جدید، نوع متغیر نیز تغییر می‌یابد.

به‌عنوان‌مثال، دستورات زیر را ببینید:

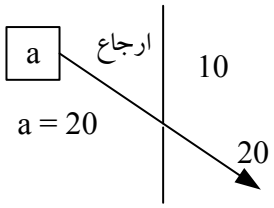
```
>>> a = 10
>>> a = 20
>>> b = a
```

دستور اول، یک شیء از نوع عدد صحیح با مقدار ۱۰را در مکانی از حافظه ایجاد می‌کند و متغیر a را در جای دیگری از حافظه تعریف می‌کند و یک پیوند از متغیر a به ۱۰ برقرار می‌کند(مانند شکل زیر):

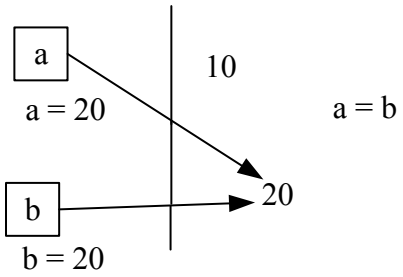


۱. object

دستور دوم، شیء ۲۰ را در یک مکان حافظه ایجاد خواهد کرد و یک پیوند از متغیر قبلی a به آن برقرار می‌کند(مانند شکل زیر):



دستور سوم، متغیر b را به‌عنوان یک شیء دیگر در مکان جدید تعریف می‌کند و یک پیوند بین شی‌ای که a به آن اشاره می‌کند، برای شیء b برقرار می‌کند(مانند شکل زیر):

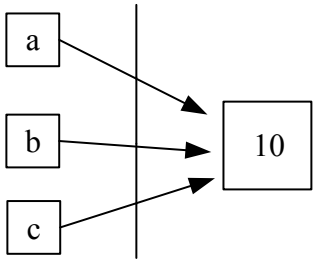


۶-۴-۱. انتساب چندگانه

در زبان پایتون می‌توان چند متغیر را به‌طور هم‌زمان با انتساب چندگانه مقداردهی نمود. یعنی، می‌توان با یک دستور چند متغیر را ایجاد کرد که به یک شیء ارجاع می‌دهند. به‌عنوان‌مثال، دستور زیر را ببینید.

```
>>> a = b = c = 10
```

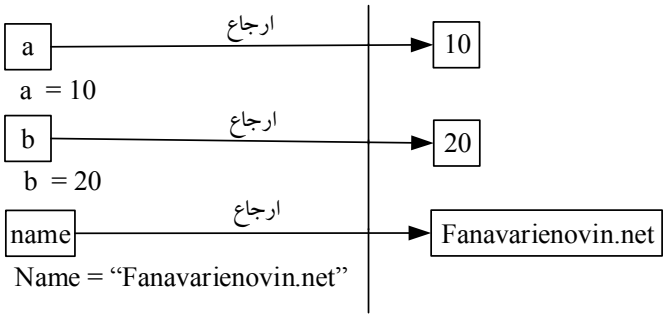
این دستور موجب می‌شود تا شیء ۱۰ ایجادشده، سپس سه شیء به نام‌های a، b و c ایجاد شوند و ارتباط همه این اشیاء را با شیء ۱۰ برقرار می‌کند.



همان‌طور که در این دستور مشاهده کردید، تمام این اشیاء (a، b و c) به شیء ۱۰ اشاره می‌کنند، امّا، گاهی اوقات نیاز است، با یک دستور چند متغیر تعریف کنیم که به چند شیء متفاوت اشاره کنند. برای این منظور، تنها از یک عملگر "=" استفاده می‌شود و قبل از عملگر "="، بین متغیرها "," قرار می‌گیرد. بعد از علامت "=" بین اشیاء (مقادیر) نیز , (کاما) قرار می‌گیرد. به‌عنوان‌مثال، دستور زیر را ببینید:

```
>>> a , b, name = 10, 20, "Fanavarienovin.net"
```

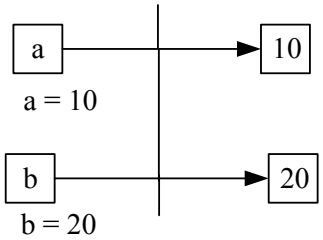
این دستور ابتدا سه شیء به نام‌های ۱۰، ۲۰ و"Fanavarienovin.net" ایجاد می‌کند، سپس سه شیء دیگر به نام‌های a، b و name ایجاد می‌نماید و a را به ۱۰، b را به ۲۰ و name را به"Fanavarienovin.net"پیوند می‌دهد (شکل زیر):



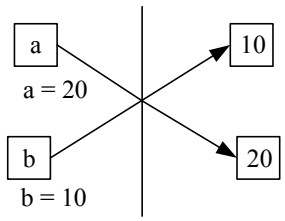
یکی از کاربردهای بسیار مهم انتساب چندگانه، تعویض محتوی دو متغیر است. به‌عنوان‌مثال، دستورات زیر را ببینید:

```
>>> a = 10
>>> b = 20
>>> a, b = b, a
```

دستور اول، شیء ۱۰ و متغیر (شیء) a را ایجاد کرده، a را به شیء ۱۰ پیوند می‌دهد، دستور دوم، شیء ۲۰ و شیء b را ایجاد کرده، پیوند بین b و ۲۰ را برقرار می‌کند، شکل زیر:



دستور سوم، مکانی که b به آن اشاره می‌کند را به a و مکانی که a به آن اشاره می‌کند را به b تخصیص می‌دهد(شکل زیر):



۵-۱. عملگرها

عملگرها [۱]، نمادهایی هستند که اعمال خاصی را انجام می‌دهند. عملگرها انواع مختلف دارند که برخی از آن‌ها عبارت‌اند از:

---

[۱] .operators

۱. عملگرهای محاسباتی ۲. عملگرهای رابطه‌ای ۳. عملگرهای ترکیبی
۴. عملگرهای منطقی ۵. عملگرهای بیتی

۱-۵-۱. عملگرهای محاسباتی

این عملگرها برای انجام محاسبات بر روی‌داده‌های عددی به کار می‌روند (جدول ۳-۱). ازجمله این عملگرها می‌توان عملگرهای + (جمع)، - (تفریق)، * (ضرب)، / (تقسیم)، ٪ (باقی‌مانده تقسیم صحیح) ، // (تقسیم صحیح)، ** (توان) را نام برد. با عملگرهای +، -، *، / از قبل آشنا هستید.

جدول ۳ - ۱ عملگرهای محاسباتی.

عملگر	نام عملگر	مثال	نتیجه	توضیحات
+	جمع	۳ + ۱۲	۱۵	عملوند اول را با عملوند دوم جمع می‌کند.
-	تفریق	۳ – ۱۳,۵	۱۰,۵	عملوند دوم را از عملوند اول کم می‌کند.
*	ضرب	۱۲* ۲,۵	۳۰	عملوند اول را در عملوند دوم ضرب می‌کند.
/	تقسیم	۱۳ / ۲	۶,۵	عملوند اول را بر عملوند دوم تقسیم می‌کند.
%	باقی‌مانده تقسیم صحیح	۱۳ ٪ ۵	۳	باقی‌مانده تقسیم صحیح عملوند اول بر عملوند دوم را محاسبه می‌کند.
//	تقسیم صحیح	۱۰//۳	۳	تقسیم صحیح عملوند اول بر عملوند دوم را محاسبه می‌کند.
**	توان	۲ ** ۹	۵۱۲	عملوند اول را به توان عملوند دوم می‌رساند.

عملگر ٪، برای محاسبه باقی‌مانده تقسیم صحیح به کار می‌رود. به‌عنوان‌مثال، دستورات زیر را ببینید:

```
>>> x = 10; y = 3
>>> z = x % y
```

در دستور دوم، مقدار متغیرهای x و y دست‌نخورده باقی می‌مانند. یعنی، مقدار x و y تغییر نمی‌کند.

عملگر **، برای محاسبه توان به کار می‌رود. به‌عنوان‌مثال، دستورات زیر را در نظر بگیرید:

```
>>> x, y = 2, 4
>>> z = x ** y
```

این دستورات، ۲ به توان ۴ را محاسبه کرده، ۱۶ را در z قرار می‌دهند.

عملگر // ، برای محاسبه تقسیم گرد شده به کار می‌رود. به‌عنوان‌مثال، دستورات زیر را ببینید:

```
>>> x, y = 23, 4
>>> z = x // y
```

این دستورات، ۲۳ تقسیم‌بر ۴ را به‌صورت گرد شده انجام داده، در z (۵) قرار می‌دهند.

۲-۵-۱. عملگرهای رابطه‌ای (مقایسه‌ای)

این عملگرها برای مقایسه دو عملوند به کار می‌روند و نتیجه درست یا نادرست را برمی‌گردانند. عملگرهای رابطه‌ای (مقایسه‌ای) در جدول ۴ – ۱ آمده‌اند. دقت کنید عملگر == تساوی (مساوی بودن) می‌باشد و عملگرهای <> و =! مخالف یا نامساوی هستند.

۳-۵-۱. عملگرهای ترکیبی

این عملگرها، ترکیبی از عملگرهای محاسباتی و = هستند. عملکرد این عملگرها را در جدول ۵-۱ می‌بینید.

۴-۵-۱. عملگرهای منطقی

عملگرهای منطقی، بر روی عبارات منطقی درست یا نادرست عمل می‌کنند. عملکرد عملگرهای منطقی در جدول ۶ -۱ آمده است. همان‌طور که در جدول ۶ -۱ می‌بینید، زمانی نتیجه عملگر and (و منطقی) درست است که هر دو عملوند مقدار درست داشته باشند. اما نتیجه عملگر or (یا منطقی) هنگامی نادرست است که هردو عملوند نادرست باشند. عملگر not، نتیجه درست را نادرست و نتیجه نادرست را به درست تبدیل می‌کند.

اکنون دستورات زیر را ببینید.

```
>>> x = true
>>> y = false
>>> z1 = x and y
>>> z2 = x or y
>>> z3 = not y
```

دستور اول شیء true و x را تعریف کرده، true را به x پیوند می‌دهد، دستور دوم، اشیاء y و false را ایجاد کرده، بین y و false پیوند برقرار می‌کند، دستور سوم، نتیجه x and y (یعنی false) را در شیء ایجادشده z1 قرار می‌دهد، دستور چهارم، نتیجه x or y (یعنی true) را در z2 قرار می‌دهد و دستور پنجم، not y (یعنی true) را در z3 قرار می‌دهد.

جدول ٤ – ١ عملگرهای رابطه‌ای (مقایسه‌ای).

عملگر	نام	مثال	نتیجه	توضیحات
>	بزرگ‌تر	۲ > ۳	False	اگر عملوند اول بزرگ‌تر از عملوند دوم باشد، نتیجه درست است، وگرنه نتیجه نادرست می‌باشد.
>=	بزرگ‌تر یا مساوی	۵ >= ۳	True	اگر عملوند اول بزرگ‌تر یا مساوی عملوند دوم باشد، نتیجه درست است، وگرنه، نتیجه نادرست می‌باشد.
<	کوچک‌تر	۵ < ۷	True	اگر عملوند اول کوچک‌تر از عملوند دوم باشد، نتیجه درست است، وگرنه نتیجه نادرست است.
<=	کوچک‌تر یا مساوی	۵ <= ۳	false	اگر عملوند اول کوچک‌تر یا مساوی عملوند دوم باشد، نتیجه درست است، وگرنه نتیجه نادرست خواهد شد.
<> یا =!	نامساوی	۲ =! ۵	true	اگر عملوند اول مخالف عملوند دوم باشد، نتیجه درست است، وگرنه، نتیجه نادرست خواهد بود.
==	تساوی	۲ == ۳	false	اگر عملوند اول مساوی عملوند دوم باشد، نتیجه درست است، وگرنه نتیجه نادرست خواهد شد.

جدول۵-۱ عملگرهای ترکیبی.				
عملگر	روش استفاده	مثال	نتیجه	عملکرد
+=	x += y	x = 3; x += 5	۸	x =x + y
-=	x -= y	x = 7; x -= 3	۴	x = x - y
*=	x *= y	x = 3; x *= 5	۱۵	x = x * y
/=	x /= y	x = 17; x /=5	۳,۴	x = x / y
%=	x %= y	x = 17; x %=5	۲	x = x % y
**=	x **= y	x=3; x **=2	۹	x = x**y
//=	x //= y	x=17; x//=3	۵,۰	x = x//y

جدول ٦-۱ عملکرد عملگرهای منطقی.					
not y	not x	x or y	x and y	x	Y
true	true	false	false	false	False
false	true	true	false	false	True
true	false	true	false	true	false
false	false	true	true	true	true

۵-۵-۱. عملگرهای بیتی

عملگرهای بیتی، عملگرهایی که بر روی بیت‌های داده کار می‌کنند و می‌توانند آن‌ها را دست‌کاری کنند، برخی از این عملگرها عبارت‌اند از:

۱. **عملگر &، "و" بیتی** را انجام می‌دهد. این عملگر، دو عملوند را بیت به بیت باهم "و" بیتی می‌نماید (نتیجه و بیتی زمانی یک است که هر دو بیت ۱ باشند). به‌عنوان‌مثال، دستورات زیر را ببینید:

```
>>> a, b = 3, 2
>>> z = a & b
```

a= 00000011
b=00000010
==========
z = 00000010

پس z برابر با 2 می‌شود.

۲. **عملگر|، "یا" بیتی** را انجام می‌دهد. این عملگر، دو عملوند را بیت به بیت باهم "یا بیتی" نموده (نتیجه یا بیتی زمانی صفر است که هر دو بیت ۰ باشند). به‌عنوان‌مثال، دستورات زیر را ببینید:

```
>>> a, b = 3, 2
>>> z = a | b
```

a= 00000011
b=00000010
==========
z = 00000011

پس z برابر 3 خواهد شد.

۳. **عملگر ^، xor (یا انحصاری) بیتی** را انجام می‌دهد. این عملگر دو عملوند را بیت به بیت (بیت‌های متناظر) باهم یا انحصاری می‌کند (نتیجه یا انحصاری زمانی یک است که دو بیت مخالف یکدیگر باشند). به‌عنوان‌مثال، دستورات زیر را ببینید:

```
>>> a, b = 7, 2
>>> z = a ^ b
```

a= 00000111
b=00000010
==========
z = 00000101

۴. **عملگر ~، نقیض بیتی** است. این عملگر قبل از یک عملوند قرارگرفته، تمام بیت‌های 1 آن را به 0 و تمام بیت‌های 0 را به 1 تبدیل می‌کند. به‌عنوان‌مثال، دستورات زیر را مشاهده کنید:

```
>>> a = 10
>>> b = ~ a
```

a= 00001010
b=11110101

۵. **عملگر >>، شیفت به چپ** را انجام می‌دهد.این عملگر بین دو عملوند قرارگرفته و مقدار عملوند سمت چپ را به تعداد عملوند سمت راست به سمت چپ شیفت می‌دهد. به‌عنوان‌مثال، دستورات زیر را ببینید:

```
>>> a, b = 2, 3
>>>z = a << b
```

a= 00000010
b=3
---
z = 00010000

همان‌طور که مشاهده می‌شود، z برابر با ۱۶ است. یعنی، با هر شیفت به چپ، مقدار a در ۲ ضرب می‌شود و در z قرار می‌گیرد. پس مقدار a در ۸ ضرب شده (۲۳ * ۲) تا ۱۶ به‌دست‌آمده، در z قرار می‌گیرد.

۶. **عملگر <<، شیفت به سمت راست** را انجام می‌دهد. این عملگر بین دو عملوند قرارگرفته و مقدار عملوند اول را به تعداد عملوند دوم به سمت راست شیفت می‌دهد. به‌عنوان‌مثال، دستورات را مشاهده کنید:

```
>>> a =12
>>> z = a >> 2
```

a = 00001100
z = 00000011

همان‌طور که در این دستورت مشاهده کردید، با هر شیفت به چپ عدد تقسیم‌بر ۲ می‌شود، مقدار ۱۲ تقسیم‌بر ۴ شده و مقدار 3 (یعنی، 00000011) به‌دست‌آمده است.

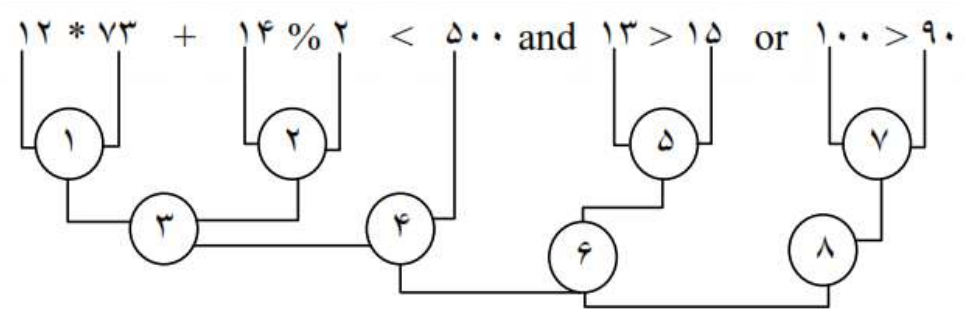
۶-۵-۱. اولویت عملگرها

فرض کنید عبارت مقابل را داشته باشید: **۲ + ۳ * ۵**

نتیجه این عبارت می‌تواند یکی از مقادیر ۲۵ یا ۱۷ باشد. چون اگر عملگر جمع اول انجام شود، نتیجه عبارت برابر ۳ + ۲ یعنی ۵ و سپس ۵ * ۵ (برابر با ۲۵) خواهد بود. اما، اگر عملگر ضرب اول انجام شود. یعنی ۵ * ۳ ابتدا انجام شود، نتیجه برابر با ۱۵ + ۲ (۱۷) خواهد بود. چون، اولویت عملگر ضرب بیش‌تر از عملگر جمع می‌باشد، پس نتیجه ۱۷ خواهد شد. اگر عبارت از ترکیبی از عملگرهای مختلف تشکیل‌شده باشد اولویت

عملگرها را طبق جدول ۷ – ۱ می‌توان تعیین کرد. یعنی، بالاترین اولویت را عملگر () و پایین‌ترین اولویت را عملگرهای and، or و not دارند.

تقدم انجام عملگرها در عبارت زیر را تعیین کنید؟



جدول ۷ – ۱ اولویت عملگرها

ردیف	اولویت
1	( )
2	**
3	~
4	+ و – (علامت مثبت و منفی)
5	*، /، %، //
6	+ و – (به‌علاوه و منها) جمع و تفریق
7	>> و <<
8	|، ^، &
9	<>، =!، ==
10	<، =<، >، =<
11	=، =**، =/، =*، =//، =%، =-، =+
12	is not , is
13	in not , in
14	not , and ،or

فرآیند انجام این عبارت در جدول ۸ – ۱ آمده است.

۶–۱. انواع داده‌ها(اشیای آماده)

پایتون هر نوع داده را توسط یک کلاس ارائه می‌کند. بنابراین، هر داده **نمونه‌ای**[1] یا یک **شیء**[2] از کلاس مشخص است. علاوه بر کلاس‌های آماده، برنامه‌نویس می‌تواند کلاس‌های جدیدی تعریف کند که در فصل‌های بعدی خواهیم دید. در پایتون انواع داده‌های مختلفی وجود دارند که عبارت‌اند از:

۱. داده‌های عددی ۲. داده‌های رشته‌ای ۳. لیست‌ها

۴. مجموعه‌ها ۵. فایل‌ها ۶. دیکشنری‌ها

---

1. Instance 2. Object

جدول ۸– ۱ فرآیند انجام عبارت نمونه.			
مراحل	انجام عملیات	عملگر	نتیجه
1	12 * 73	*	876
2	14 % 2	%	0
3	876 + 0	+	876
4	876 > 500	>	True
5	13 < 15	<	True
6	True and True	and	True
7	100 < 90	<	False
8	True or False	or	True

در این فصل به داده‌های عددی می‌پردازیم و در فصول بعدی رشته‌ها، لیست‌ها، مجموعه‌ها، فایل‌ها و دیکشنری را خواهیم آموخت.

۱–۶–۱. انواع داده‌های عددی

در پایتون گروهی از انواع اشیاء وجود دارند که برای کار با اعداد به کار می‌روند. این انواع اشیاء عبارت‌اند از:

۱. داده‌های صحیح (Integer)
۲. داده‌های ممیز شناور (Float)
۳. داده‌های مختلط (Complex)
۴. داده‌های ده‌دهی (Decimal)
۵. داده‌های کسری (Fraction)
۶. داده‌های منطقی (Boolean)

داده‌های صحیح

این نوع داده‌ها برای معرفی اعداد صحیح مثبت و منفی (بدون ممیز اعشار) نظیر 1785، 0، 900- و غیره به کار می‌روند. در پایتون نسخه ۲ دو نوع داده صحیح وجود دارد که عبارت‌اند از:

۱. داده‌های صحیح با محدودیت اندازه که int نامیده می‌شوند.

۲. داده‌های صحیح بدون محدودیت اندازه که long نامیده می‌شوند. در پایتون نسخه ۲ برای تعیین داده‌های صحیح با نوع long ، انتهای داده کاراکتر L یا l قرار می‌گیرد.

چنانچه در نسخه ۲ پایتون داده‌ای را با نوع int در نظر بگیرید، سرریز[1] اتفاق افتد(یعنی، داده‌ای را در آن متغیر قرار دهید که در متغیر جا نشود)، خطایی رخ نخواهد داد و پایتون به‌صورت خودکار نوع int را به شیء با نوع long تبدیل خواهد کرد.

دقت کنید که بیش‌ترین مقدار و کم‌ترین مقدار یک شیء نوع int را می‌توانید با sys.maxint - 1 و sys.maxint ببینید. برای این منظور می‌توانید دستورات زیر را اجرا کنید:

```
>> import sys
>>> print sys.maxint , sys.maxint-1
```

امّا در نسخه ۳ پایتون اعداد صحیح با یک نوع int ارائه می‌گردند که ازلحاظ اندازه محدودیتی ندارند. لذا، استفاده از کاراکترهای L و l در پایان این اعداد مجاز نمی‌باشد. چون در این نسخه محدودیت نوع int

---

[1]. Overflow [2]. Binary [3]. Octal [4]. Hexadecimal

حذف‌شده است، لذا، sys.maxint حذف‌شده است. امّا، می‌توان به‌جای آن از دستور sys.maxsize استفاده کرد.

اعداد صحیح را می‌توان در **مبنای دو**[2]، **مبنای هشت**[3] و **مبنای شانزده**[4] بیان کرد. اعداد مبنای ۲ را باید با 0b یا 0B شروع نمود. به‌عنوان‌مثال، عدد زیر در مبنای ۲ است:

```
>>> a = 0b1101
```

امّا، اعداد مبنای ۸ را می‌توان با 0O یا 0o شروع کرد. به‌عنوان‌مثال، عدد زیر در مبنای ۸ است:

```
>>> a = 0o743
```

ولی، اعداد مبنای ۱۶ را باید با 0x یا 0X آغاز نمود. به‌عنوان‌مثال، عدد زیر در مبنای ۱۶ است:

```
>>> a = 0xb7D
```

در پایتون توابعی برای تبدیل یک عدد از مبنای ۱۰ به مبنای ۲، ۸ و ۱۶ وجود دارند که عبارت‌اند از:

تابع bin()، یک عدد مبنای ۱۰ را به عدد مبنای ۲ تبدیل می‌کند. به‌عنوان‌مثال، دستورات زیر خروجی 0b101 را نمایش می‌دهند:

```
>>> a = 5
>>> print(bin(a))
```

تابع oct()، برای تبدیل عدد مبنای ۱۰ به مبنای ۸ به کار می‌رود. به‌عنوان‌مثال، دستورات زیر خروجی 0o22 را نمایش می‌دهند:

```
>>> a = 18
>>> print(oct(a))
```

تابع hex()، برای تبدیل عدد مبنای ۱۰ به مبنای ۱۶ به کار می‌رود. به‌عنوان‌مثال، دستورات زیر خروجی 0x14 را نمایش می‌دهند:

```
>>> a = 20
>>> print(hex(a))
```

تابع int()، برای تبدیل یک عدد از یک مبنا به مبنای ۱۰ به کار می‌رود. به‌عنوان‌مثال، دستورات زیر 20 را نمایش می‌دهند:

```
>>> a = 0x14
>>> print(int(a))
```

اعداد اعشاری

اعداد می‌توانند اعشاری باشند. پایتون برای نگه‌داری اعداد اعشاری (نظیر 3.1415، 0.5 و...) از اشیایی با نوع float استفاده می‌کند. علاوه بر نمایش اعداد اعشاری به‌صورت ممیز شناور می‌توان اعداد اعشاری را با نماد علمی[1] نمایش داد که در پایتون برای نمایش اعداد اعشاری با نماد علمی از حرف E یا e استفاده می‌شود. به‌عنوان‌مثال، در پایتون اعداد 5×10^{7} و 6×10^{-10} به ترتیب به صورت‌های 5E7 (5e7) و 6E-10 (6e-10) نمایش داده می‌شوند.

اعداد مختلط

[1]. Scientific Notation [2]. Complex Number [3]. Real [4]. Imaginary

همان‌طور که در ریاضی دیدیم، هر عدد مختلط[۲] از دو بخش حقیقی[۳] و موهومی[۴] تشکیل‌شده است. اعداد مختلط در پایتون با نوع شیء complex تعریف می‌شوند. عدد مختلط در پایتون به‌صورت x + yj نمایش داده می‌شود که x نشان‌دهنده، بخش حقیقی و y نشان‌دهنده بخش موهومی است. به‌عنوان‌مثال، عدد 3 + 5j یک عدد مختلط است که بخش حقیقی آن 3 و بخش موهومی آن 5 می‌باشد.

از کلاس complex می‌توان برای تعریف اعداد مختلط استفاده نمود که این کلاس به‌صورت زیر به کار می‌رود:

```
complex (real, imag)
```

که real بخش حقیقی و imag بخش موهومی عدد مختلط را مشخص می‌کند. چنانچه هریک از این بخش‌ها به‌عنوان آرگومان ارسال نشوند، به‌صورت پیش‌فرض صفر در نظر گرفته می‌شوند، به‌عنوان‌مثال، دستورات زیر را ببینید:

```
>>> a = 5; b = -3
>>> complex(a, b)
```

دستور اول، مقادیر 5 و 3- را به ترتیب به اشیاء a و b تخصیص می‌دهد و دستور دوم، یک شیء complex با مقدار حقیقی 5 و مقدار موهومی 3- ایجاد می‌نماید(خروجی (5-3j) نمایش داده می‌شود).

با دو صفت real و imag می‌توان بخش‌های حقیقی و موهومی یک عدد مختلط را به‌دست آورد. به‌عنوان‌مثال، دستورات زیر را ببینید:

```
>>> a = 4.5
>>> b = complex(a)
>>> b.real
>>> b.imag
```

دستور اول، متغیر a با مقدار ۴٫۵ را ایجاد کرده، دستور دوم، شیء 4.5+0.0j را ایجاد می‌نماید و دستور سوم، بخش حقیقی شیء مختلط b (یعنی ۴٫۵) را نمایش می‌دهد و دستور چهارم، بخش موهومی شیء b (یعنی 0.0) را نمایش می‌دهد.

اعداد دسیمال(ده‌دهی)

همان‌طور که بیان گردید، در پایتون اعداد اعشاری به‌صورت شیء با نوع float معرفی می‌گردند. مفسر پایتون برای ارائه نوع ممیز شناور از کدگذاری Binary-Floating Point استفاده می‌نماید که برای محاسبات حسابداری مناسب نمی‌باشد. چون، پایتون اعدادی از قبیل 0.1، 0.2 و 0.3 را به‌صورت 0.1000000000000001، 0.2000000000000001 و 0.3000000000000001 نشان می‌دهد که دقیقاً برابر 0.1، 0.2 و 0.3 نمی‌باشند. این موضوع در برخی از اوقات موجب خطای منطقی خواهد شد. به‌عنوان‌مثال، دستورات زیر را ببینید:

```
>>> a =  0.2 + 0.2 + 0.2
>>> a == 0.6
```

با اجرای این دستورات False نشان داده می‌شود. یعنی، 0.2 + 0.2 + 0.2 برابر 0.6 نخواهد شد و این موضوع یک خطای منطقی برنامه است. اکنون اگر a را نشان دهیم، یعنی، دستور زیر را تایپ کنیم، خروجی 0.6000000000000001 نشان داده می‌شود:

```
>>>  a
```

و دستورات زیر مقدار 0.30000000000000004 را نشان می‌دهند:

```
>>> a = 0.1 + 0.1 + 0.1
>>> a
```

دستور اول، سه بار 0.1 را جمع کرده و در a قرار می‌دهد که انتظار می‌رود، نتیجه 0.3 شود، امّا، با نمایش a می‌بینیم که a برابر با 0.30000000000000004 می‌باشد که این انحراف ناشی از نحوه کدگذاری اعداد اعشاری می‌باشد. به همین دلیل، در پایتون یک نوع شیء جدید به نام Decimal طراحی‌شده است. این نوع در ماژول decimal قرار دارد. برای استفاده از نوع Decimal ابتدا باید با دستور زیر این ماژول را به برنامه اضافه کنید:

```
>>> import decimal
```

دستورات زیر True را نشان می‌دهند:

```
>>> import decimal
>>> a = decimal.Decimal("0.6")
>>> b = decimal.Decimal("0.2")
>>> a == b + b + b
```

دستور اول، ماژول decimal را اضافه می‌کند، دستور دوم، مقدار دهدهی 0.6 را در شیء a قرار می‌دهد، دستور سوم، مقدار دهدهی 0.2 را در شیء b قرار می‌دهد، دستور چهارم، a (0.6) را با حاصل جمع b + b + b (0.2+0.2+0.2) مقایسه کرده و نتیجه True را برمی‌گرداند.

اعداد کسری

نوع دیگری که پایتون پشتیبانی می‌کند، اشیاء کسری(اعداد گویا) می‌باشد. برای ایجاد اعداد گویا از نوع شیء Fraction استفاده می‌شود. این شیء در ماژول fractions قرار دارد. برای استفاده از این نوع شیء باید کلاس fractions را به برنامه import کرد. به‌عنوان‌مثال، دستورات زیر را ببینید:

```
>>> import fractions
>>> a , b = 2, 3
>>> f1 = fractions.Fraction(a, b)
>>> f1
>>> print(f1)
```

دستور اول، ماژول fractions را به برنامه اضافه می‌کند، دستور دوم، دو شیء به نام a و b ایجاد کرده، اشیاء ایجادشده ۲ و ۳ را به ترتیب به a و b نسبت می‌دهد، دستور سوم، شیء f1 را از نوع Fraction تعریف کرده، مقدار Fraction(2, 3) را به آن تخصیص می‌دهد، دستور چهارم، مقدار Fraction(2, 3) را نمایش می‌دهد، دستور پنجم، مقدار عدد کسری ۲/۳ را نمایش می‌دهد.

نوع منطقی

نوع منطقی یکی از دو مقدار True یا False را دارد. در این کلاس True و False به ترتیب معادل اعداد ۱ و صفر(0) می‌باشند. برای نمایش نوع منطقی در پایتون کلاس bool وجود دارد که مشتق کلاس اعداد صحیح (int) است. به‌عنوان‌مثال، دستورات زیر را ببینید:

```
>>> a = True
>>> a
```

دستور اول، شیء a را ایجاد کرده و شیء True را به آن نسبت می‌دهد، دستور دوم، مقدار شیء a (یعنی True) را نمایش می‌دهد.

اکنون دستورات زیر را مشاهده کنید:

```
>>> a = True; b = False
>>> int(a)
>>> int(b)
>>> complex(a, b)
```

دستور اول، اشیایی به نام‌های a و b تعریف می‌کند و به ترتیب مقادیر True و False را به آن‌ها تخصیص می‌دهد، دستور دوم، int(a) یعنی (۱) را نمایش می‌دهد، دستور سوم، int(b) یعنی، مقدار صفر را نمایش می‌دهد و دستور چهارم، complex(a,b) یعنی 1+0j را نمایش می‌دهد.

۲-۶-۱. رشته

در پایتون رشته[1]، مجموعه‌ای از کاراکترهای پشت سر هم است که در بین جفت کتیشن(" ") یا تک‌کتیشن(' ') قرار می‌گیرند. به‌عنوان‌مثال، دستورات زیر را ببینید:

```
>>> a = "Python language"
>>> a
>>> print(a)
```

دستور اول، شیءایی به نام a با نوع رشته‌ای تعریف می‌کند و شیء رشته‌ای 'Python language' را به آن تخصیص می‌دهد، دستور دوم، مقدار a (یعنی، 'Python language') را نمایش می‌دهد و دستور سوم نیز مقدار a (یعنی، Python language) را نمایش می‌دهد.

در پایتون برخلاف برخی از زبان‌های برنامه‌نویسی دیگر **نوع کاراکتری**[2] وجود ندارد. یعنی، در زبان پایتون کاراکتر، رشته‌ای با طول یک است.

در پایتون می‌توان از کاراکترهای کتیشن در داخل یکدیگر استفاده کرد. در این حالت فقط باید نوع کتیشن داخلی با بیرونی متفاوت باشد. امّا، اگر بخواهید از کاراکتر کتیشن یکسان استفاده کنید، باید از کاراکتر \ ۳ قبل از کتیشن استفاده کنید. به‌عنوان‌مثال، دستورات زیر را مشاهده کنید:

```
>>> "Python 'language'"
>>> 'I\'m a student'
```

دستور اول، کاراکتر تک‌کتیشن را در داخل جفت‌کتیشن استفاده می‌کند(خروجی را به‌صورت "Python 'language'" نمایش می‌دهد) و دستور دوم، کاراکتر تک‌کتیشن را در داخل کاراکتر تک‌کتیشن دیگر استفاده می‌کند. برای این منظور، از کاراکتر \ قبل از کاراکتر تک‌کتیشن داخلی استفاده می‌نماید(عبارت "I'm a student" را نمایش خواهد داد).

عملگرهای رشته

۱. **عملگر +**، این عملگر برای اتصال (الحاق) دو رشته به کار می‌رود. به‌طوری‌که رشته سمت راست را به انتهای رشته سمت چپ اضافه می‌کند. به‌عنوان‌مثال، دستورات زیر را ببینید:

```
>>> s1 = "Fanavarienovin"
>>> s2 = ".net"
>>> s1 + s2
```

---

[1]. String [2]. Char [3]. Escape

دستور اول، رشته s1 را ایجاد کرده، شیء Fanavarienovin را به آن تخصیص می‌دهد، دستور دوم شیء s2 را ایجاد نموده، رشته net. را در آن قرار می‌دهد و دستور سوم، رشته s2 را به انتهای رشته s1 می‌چسباند. یعنی 'Fanavarienovin.net' را نمایش می‌دهد.

۲. **عملگر***، برای تکرار یک رشته به کار می‌رود. این عملگر دو عملوند یکی از نوع رشته‌ای و دیگری از نوع عدد صحیح را دریافت کرده، رشته را به تعداد عدد دریافت شده تکرار می‌کند و برمی‌گرداند. به‌عنوان‌مثال، دستور زیر عبارت 'Fanavarienovin Fanavarienovin Fanavarienovin ' را نمایش می‌دهد:

```
>>> "Fanavarienovin " * 3
```

۷-۱. تبدیل نوع

در زبان پایتون مانند دیگر زبان‌ها می‌توان یک نوع شیء داده‌ای را به شیء داده‌ای با نوع دیگر تبدیل کرد. برای این منظور، توابعی وجود دارند که برخی از آن‌ها عبارت‌اند از:

۳. **تابع int()**، برای تبدیل هر نوع داده به نوع صحیح به کار می‌رود. این تابع به‌صورت زیر استفاده می‌شود:

```
int (pars1, pars2)
```

pars1، مقدار داده‌ایی است که باید تبدیل شود و pars2 مبنای اولیه تبدیل را تعیین می‌کند که اگر ذکر نشود، مبنای ۱۰ در نظر گرفته می‌شود:

```
>>> int(0x41a)
>>> int(0b10101)
>>> int("21", 16)
>>> int("57", 10)
>>> int()
>>> int(4.9)
```

دستور اول، مقدار 0x41a مبنای ۱۶ را به مبنای ۱۰ (یعنی 1050) تبدیل می‌کند، دستور دوم، مقدار 0b10101 را به مبنای ۱۰(یعنی ۲۱) تبدیل می‌کند، دستور سوم، مقدار ۲۱ مبنای ۱۶ را به مبنای ۱۰ (یعنی ۳۳) تبدیل می‌کند، دستور چهارم، مقدار رشته‌ای"57" در مبنای ۱۰ را به عدد صحیح مبنای ۱۰ (یعنی ۵۷) تبدیل می‌کند، دستور پنجم، مقدار صفر را برمی‌گرداند، چون اگر پارامترهای تابع int() ذکر نشوند، مقدار صفر را برمی‌گرداند و دستور ششم نوع اعشاری با مقدار 4.9 را به نوع صحیح با مقدار ۴ تبدیل می‌کند.

۴. **تابع float()**، برای تبدیل یک مقدار به نوع اعشاری ممیز شناور به کار می‌رود و به‌صورت زیر استفاده می‌شود:

```
float (value)
```

value، مقداری است که باید به نوع float تبدیل شود و اگر چیزی ذکر نشود، 0.0 را برمی‌گرداند. اکنون دستورات زیر را ببینید:

```
>>> float("21")
>>> float()
>>> float("4e+4")
```

دستور اول، مقدار رشته‌ای"21" را به نوع اعشاری با مقدار 21.0 تبدیل می‌کند، دستور دوم، مقدار 0.0 را برمی‌گرداند، دستور سوم، مقدار رشته‌ای "4e+4" را به نوع اعشاری با مقدار 40000.0 تبدیل می‌کند.

۸–۱. تابع print()

همان‌طور که قبلاً بیان گردید، زمانی که یک عبارت را در مفسر تایپ کرده باشید و کلید Enter را بزنید، عبارت فوراً ارزیابی‌شده، نتیجه ارزیابی عبارت نمایش داده می‌شود. به‌عنوان‌مثال، دستور زیر را تایپ کرده تا نتیجه ارزیابی عبارت (یعنی، 57.125) را ببینید:

```
>>>  8*(5+3)-110/6
```

این ویژگی برای زمانی به کار می‌رود که بخواهید نتیجه یک دستور محاسباتی را حساب کرده یا بخواهید املائی عبارت را ارزیابی کنید.

حال، اگر این دستورات را دریک ماژول تایپ کنید، با اجرای این دستورات خروجی آن‌ها نمایش داده نمی‌شود. برای نمایش اطلاعات در ماژول می‌توانید از تابع print() استفاده کنید. در تابع print()، می‌توانید هر دنباله‌ای از عباراتی را بی‌آورید. این عبارات با کاما (,) از هم جدا می‌شوند. در هنگام استفاده از تابع print() به نکات زیر دقت کنید:

۱. اگر تابع print() را بدون آرگومان استفاده کنید، یک سطر خالی چاپ خواهد شد.
۲. با هر بار اجرای تابع print()، یک سطر چاپ خواهد شد.
۳. اگر آرگومان تابع print()، رشته‌ای باشد، عین رشته را در خروجی نمایش می‌دهد.
۴. اگر در آرگومان تابع print()، یک عبارت آورده شود، نتیجه عبارت در خروجی نمایش داده می‌شود.
۵. اگر در آرگومان تابع print()، نام یک متغیر آورده شود، مقدار متغیر در خروجی نمایش داده می‌شود.

به‌عنوان‌مثال، دستورات زیر را ببینید:

```
>>> x, y = 3, 5
>>> print(x, " + ",  y, " = ", x + y)
```

دستور اول، مقدار ۳ را در x و مقدار ۵ را در y قرار می‌دهد، دستور دوم، ابتدا مقدار x (یعنی ۳)، سپس علامت "+"، در ادامه مقدار y یعنی (۵)، در پایان علامت = و نتیجه عبارت x + y (یعنی ۸) را نمایش می‌دهد؛ یعنی، خروجی زیر:

```
3 + 5 = 8
```

درواقع هر چیزی که در آرگومان تابع print() استفاده می‌شود، برای نمایش به نوع رشته تبدیل می‌گردد، به‌عنوان‌مثال، اگر متغیر n عددی صحیح باشد که به مقدار ۱۰ ارجاع می‌دهد، امّا وقتی به‌عنوان آرگومان print() استفاده می‌گردد، درنهایت مقدار ۱۰ به یک‌رشته تبدیل می‌شود. بااین‌وجود، باید دقت کنید که متغیر n همچنان به یک عدد صحیح ارجاع می‌دهد. به‌عنوان‌مثال، دستورات زیر را ببینید:

```
>>> n = 10
>>> print("n is" + n)
```

با اجرای این دستور انتظار داریم که عبارت زیر نمایش داده شود:

```
n is 10
```

درصورتی‌که با اجرای این دستور خطای زیر صادر می‌گردد:

```
Traceback (most recent call last:(
  File "<pyshell#11>", line 1, in <module>
    print("n is" + n )
TypeError: Can't convert 'int' object to str implicitly
```

چون n از نوع عددی است. پس باید به نوع رشته تبدیل شود یا دستور به‌صورت زیر به کار رود:

```
>>> print("n is", n)
```

اکنون خروجی زیر نمایش داده می‌شود :

```
n is 10
```

برای تبدیل n به نوع رشته‌ای می‌توانید از تابع str() استفاده کنید(مانند دستور زیر):

```
>>> print("n is " + str(n))
```

با اجرای این دستور، خروجی زیر نمایش داده می‌شود:

```
n is 10
```

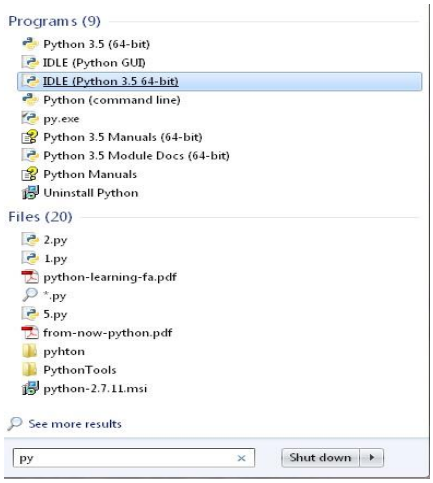
۹–۱. تایپ، ذخیره و اجرای برنامه در پایتون

تاکنون، دستورات را به‌صورت تکی نوشته و اجرا کردیم و نتایج را مشاهده نمودیم. در پایتون امکانی وجود دارد تا بتوانید دستورات را به‌صورت یکجا تایپ کرده و اجرا نمایید. برای این منظور، به ویراستاری نیاز دارید تا برنامه را در آن تایپ کنید. سپس آن را اجرا کنید. در نسخه‌های مختلف پایتون، ویراستاری آماده‌شده است که در آن می‌توانید برنامه‌تان را تایپ و اجرا کنید.

به‌عنوان‌مثال، در پایتون نسخه ۳ به بعد فرآیند اجرا و ویرایش مانند مثال ۱ – ۱ است.

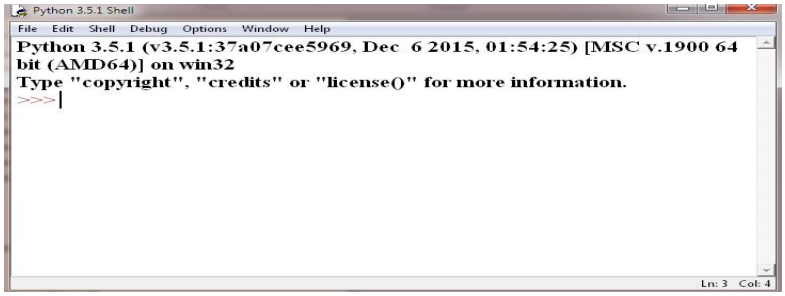
مثال ۱–۱. برنامه‌ای که مراحل تایپ، ذخیره و اجرای یک برنامه ساده را نشان می‌دهد.

۱. در منوی Start، در باکس Search، عبارت py را تایپ کرده تا لیست برنامه‌هایی که با py شروع می‌شوند، ظاهر شود (شکل۲–۱).



شکل۲–۱ لیست برنامه‌هایی که با py شروع می‌شوند.

۲. برنامه(IDLE (python 3.5 64-bit را اجرا کنید تا شکل ۳ -۱ ظاهر شود.

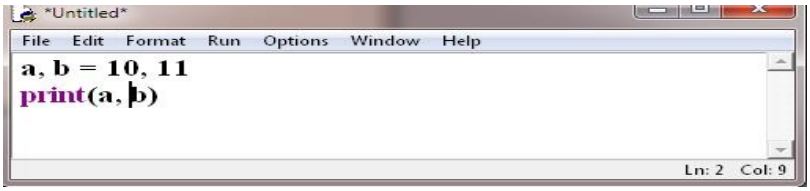


شکل ۳-۱ Python 3.5.1 Shell.

۳. گزینه File / New File (یا کلیدهای ترکیبی Ctrl+N) را فشار دهید تا فایل جدیدی ایجاد شود (شکل۴-۱).

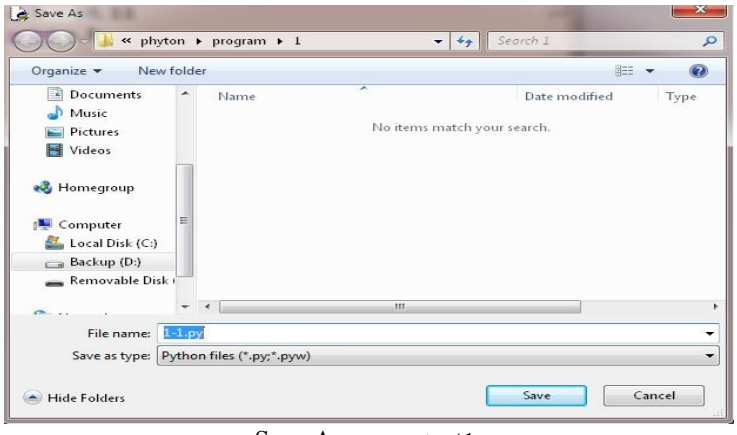
شکل ٤-۱ایجاد فایل جدید Python.

۴. اکنون دستورات برنامه‌تان را تایپ کنید (مانند شکل۵-۱).



شکل ٥ - ۱دستورات نمونه برای اجرا.

۵. گزینه File / Save As (یا کلیدهای ترکیبی Ctrl+ Shift+ S) را اجرا کنید تا پنجره Save As ظاهر شود (شکل ۶ -۱).



شکل ٦-۱پنجره Save As.