
آموزش گام به گام برنامه نویسی پایتون

تألیف:

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



فن آوری نوین

سرشناسه	وحیدی، جواد، ۱۳۵۲ -
عنوان و نام پدیدآور	آموزش گام به گام برنامه‌نویسی پایتون / تألیف جواد وحیدی، رمضان عباس نژادورزی.
مشخصات نشر	بابل: فن آوری نوین، ۱۳۹۶
مشخصات ظاهری	۳۸۰ ص: مصور، جدول
شابک	۹۷۸-۶۰۰-۷۲۷۲-۰۷-۷-۴۱۰۰۰ ریال
وضعیت فهرست نویسی	فیبا
یادداشت	کتابنامه.
موضوع	پیتون (زبان برنامه‌نویسی کامپیوتر)
موضوع	Python (Computer program language)
موضوع	پیتون (زبان برنامه‌نویسی کامپیوتر) -- مسائل، تمرین‌ها و غیره
موضوع	Python (Computer program language) -- Problems, exercises, etc .
شناسه افزوده	عباس نژاد ورزی، رمضان، ۱۳۴۸ -
رده بندی کنگره	۱۳۹۶ ۷۳/۷۶ QA و ۹۵۳/پ
رده بندی دیویی	۰۰۵/۱۳۳
شماره کتابشناسی ملی	۴۸۸۰۳۵۷



www.fanavarienovin.net

تلفن: ۰۱۱-۳۲۲۵۶۶۸۷

بابل، کدپستی ۴۷۱۶۷-۷۳۴۴۸

فن آوری نوین

آموزش گام به گام برنامه‌نویسی پایتون

تألیف: جواد وحیدی - رمضان عباس نژادورزی

نوبت چاپ: چاپ اول

سال چاپ: پاییز ۱۳۹۶

شمارگان: ۲۰۰

قیمت: ۴۱۰۰۰ تومان

نام چاپخانه و صحافی:

شابک: ۹۷۸-۶۰۰-۷۲۷۲-۰۷-۷-۴۱۰۰۰

نشانی ناشر: بابل، چهارراه نواب، کاظم‌بیگی، جنب مسجد منصور کاظم‌بیگی، طبقه همکف

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

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

فهرست مطالب

فصل اول: آشنایی با زبان پایتون

- ۱-۱. مقدمه‌ای بر زبان برنامه‌نویسی پایتون..... ۹
- ۱-۲. مقایسه‌ی زبان برنامه‌نویسی پایتون با سایر زبان‌های برنامه‌نویسی..... ۱۱
- ۱-۳. از پایتون برای چه اپلیکیشن‌هایی می‌توان استفاده کرد؟..... ۱۴
- ۱-۴. آموزش زبان‌های برنامه‌نویسی..... ۱۵
 - ۱-۴-۱. سطرها..... ۱۶
 - ۱-۴-۲. بلاک بندی..... ۱۷
 - ۱-۴-۳. دستورات ۱۷..... ۱۷
 - ۱-۴-۴. شناسه‌ها..... ۱۸
 - ۱-۴-۵. متغیرها..... ۱۹
 - ۱-۴-۶. انتساب چندگانه..... ۲۰
 - ۱-۵. عملگرها..... ۲۲
 - ۱-۵-۱. عملگرهای محاسباتی..... ۲۲
 - ۱-۵-۲. عملگرهای رابطه‌ای (مقایسه‌ای)..... ۲۳
 - ۱-۵-۳. عملگرهای ترکیبی..... ۲۳
 - ۱-۵-۴. عملگرهای منطقی..... ۲۳
 - ۱-۵-۵. عملگرهای بیتی..... ۲۴
 - ۱-۵-۶. اولویت عملگرها..... ۲۶
 - ۱-۶. انواع داده‌ها(اشیای آماده)..... ۲۷
 - ۱-۶-۱. انواع داده‌های عددی..... ۲۷
 - ۱-۶-۲. رشته..... ۳۲
 - ۱-۷. تبدیل نوع..... ۳۳
 - ۱-۸. تابع print()..... ۳۴
 - ۱-۹. تایپ، ذخیره و اجرای برنامه در پایتون..... ۳۵
 - ۱-۱۰. خواندن داده..... ۳۸
 - ۱-۱۱. مسائل حل شده..... ۴۷

فصل دوم: ساختار تصمیم و تکرار

- ۲-۱. ساختارهای تصمیم‌گیری..... ۵۴
- ۲-۲. ساختارهای تکرار..... ۶۰
 - ۲-۲-۱. دستور for..... ۶۰
 - ۲-۲-۲. دستور while..... ۶۶
 - ۲-۳. دستور break..... ۶۸

- ۶۹ ۲-۴. دستور continue
- ۷۲ ۲-۵. مسائل حل شده

فصل سوم: توابع

- ۸۸ ۳-۱. انواع توابع
- ۸۸ ۳-۲. توابعی که برنامه نویس می نویسد
- ۸۸ ۳-۲-۱. نوشتن تابع
- ۹۰ ۳-۲-۲. فراخوانی تابع
- ۹۴ ۳-۳. ارسال پارامترها
- ۹۴ ۳-۳-۱. ارسال پارامتر از طریق مقدار
- ۹۹ ۳-۴. آرگومان های تابع
- ۹۹ ۳-۴-۱. آرگومان های اجباری
- ۹۹ ۳-۴-۲. آرگومان های کلمه کلیدی
- ۱۰۰ ۳-۴-۳. آرگومان های با مقدار پیش فرض
- ۱۰۲ ۳-۴-۴. تعریف و فراخوانی توابع با تعداد پارامترهای متغیر
- ۱۰۴ ۳-۵. توابع بی نام
- ۱۰۶ ۳-۶. طول عمر و محدود حضور متغیرها
- ۱۱۵ ۳-۷. مسائل حل شده

فصل چهارم: آرایه ها و بسته NUMPY

- ۱۳۲ ۴-۱. آرایه های یک بعدی
- ۱۳۲ ۴-۱-۱. تعریف آرایه
- ۱۳۶ ۴-۱-۲. دسترسی به عناصر آرایه
- ۱۳۷ ۴-۱-۳. مقداردهی به عناصر آرایه
- ۱۳۸ ۴-۱-۴. نمایش عناصر آرایه
- ۱۳۹ ۴-۲. تولید اعداد تصادفی
- ۱۴۲ ۴-۳. مرتب سازی تعویضی
- ۱۴۴ ۴-۴. جست و جوی مقدار در آرایه
- ۱۴۴ ۴-۴-۱. جست و جوی خطی (ترتیبی)
- ۱۴۴ ۴-۴-۲. جست و جوی دودویی
- ۱۴۸ ۴-۵. بسته NumPy

۱۴۹NumPy آرایه‌های
۱۶۶Numpy ریاضیات در
۱۶۹آرایه‌های دوبعدی (ماتریس)
۱۷۰تعریف آرایه دوبعدی
۱۷۰مقداردهی به عناصر آرایه دوبعدی
۱۷۱نمایش مقادیر عناصر آرایه دوبعدی
۱۷۱کار با ماتریس از طریق numpy
۱۷۶جبر خطی در numpy
۱۸۰مسائل حل شده

فصل پنجم: رشته‌ها

۱۹۲عملگرهای رشته‌ای
۲۰۰متدهای کار بر روی رشته
۲۰۵کاربرد سه کتیشن ("")
۲۰۵نمایش کاراکتر \
۲۰۸رشته‌های یونیکد
۲۰۹مسائل حل شده

فصل ششم: لیست‌ها، چندتایی‌ها، دیکشنری‌ها و مجموعه‌ها

۲۲۶لیست‌ها
۲۲۷عملگرهای کار بر روی لیست
۲۳۸متدهای کار با لیست
۲۴۰تاپل (چندتایی)
۲۴۳دیکشنری
۲۴۶عملیات بر روی دیکشنری
۲۵۱کاربرد for برای پیمایش دیکشنری
۲۵۶مجموعه‌ها
۲۵۷عملگرهای مجموعه
۲۶۱متدهای مجموعه
۲۶۶نوع frozenset
۲۶۸مسائل حل شده

فصل هفتم: کلاس‌ها و وراثت

۲۷۳	۱-۷. کلاس‌ها
۲۷۴	۱-۱-۷. تعریف کلاس‌ها
۲۷۵	۲-۱-۷. نمونه‌سازی کلاس‌ها
۲۷۵	۲-۲-۷. اعضای کلاس
۲۷۶	۱-۲-۷. دسترسی به اعضای کلاس
۲۷۶	۲-۲-۷. انواع اعضای کلاس
۲۷۸	۳-۷. سازنده‌ها و مخرب‌ها
۲۸۶	۴-۷. وراثت و چندریختی
۲۸۷	۵-۷. وراثت
۲۸۹	۶-۷. تشخیص کلاس پایه و مشتق
۲۸۹	۷-۷. تعریف کلاس مشتق
۲۹۰	۸-۷. سازنده‌ها و مخرب‌ها در کلاس‌های مشتق
۲۹۳	۹-۷. پیاده‌سازی مجدد عملگرها
۳۰۱	۱۰-۷. مسائل حل شده

فصل هشتم: فایل‌ها

۳۱۳	۱-۸. باز کردن فایل
۳۳۸	پيوست: مسائل تکمیلی
۳۸۰	منابع:

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 به قدر معلمش بداند دنیا را متوقف می‌کند.

این تویی که پیش می‌رانی...

عصر حاضر از نظر بسیاری از روشنفکران انقلاب ارتباطات و اطلاعات نام گرفته است. در واقع کلیه کارهای روزمره‌ی انسان با سیستم های ارتباطی نوین و فناوری اطلاعات گره خورده است. لذا اهمیت علم کامپیوتر به عنوان نمود اصلی فناوری اطلاعات و ارتباطات بر کسی پوشیده نیست. در دنیای مدرن امروز تسلط و آشنایی با نحوه کار با سیستم های کامپیوتری به عنوان یک سطح سواد مطلوب در جامعه شناخته می شود که اگر کسی آن را نداشته باشد باید بهای عدم آشنایی با دنیای کامپیوتر در زندگی روزمره خویش با مراجعه به کافی نت ها، درخواست های الکترونیکی و موارد دیگر بپردازد.

بدون شک، برنامه نویسی یکی از مهم ترین مهارت هایی است که امروزه نه تنها برای فارغ التحصیلان و دانشجویان رشته مهندسی کامپیوتر، بلکه برای سایر رشته ها و زمینه ها نیز به شکل ضروری، مورد نیاز است. هر ساله، بر تعداد موقعیت های شغلی که مرتبط با برنامه نویسی هستند و یا پیش نیاز اصلی آنها برنامه نویسی است، افزوده می شود. از طرفی، بخشی جدا نشدنی از فرآیند تحقیق و پژوهش در اکثر رشته های 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 همه منظوره، سطح بالا، شیء‌گرا و مفسر 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 نوشته شده و کوتاهی و بازدهی نسبی بالای آن است. کلمات کلیدی و اصلی این زبان به صورت حداقلی تهیه شده‌اند و در مقابل کتابخانه‌هایی که در اختیار کاربر است بسیار وسیع هستند.

علی‌رغم استقبال گسترده دانش پژوهان عرصه کامپیوتر از زبان پایتون، در زمینه آموزش این زبان قدرتمند کتاب‌های زیادی به زبان فارسی منتشر نشده است که کتاب حاضر برای پر کردن خلاء موجود در زمینه نوشته شده و توسط انتشارات فن آوری‌های نوین به زیور چاپ آراسته شده است. امید است اثر حاضر همانند سایر آثار مولفین مورد استقبال قرار گیرد.

از تمامی اساتید و دانشجویان عزیز تقاضا داریم، هر گونه اشکال، ابهام در متن کتاب، پیشنهاد و انتقادات را به آدرس پست الکترونیک **fanavarienovin@gmail.com** ارسال نمایند.

مؤلفین

fanavarienovin@gmail.com

نام کتاب	لینک خرید چاپی	لینک خرید الکترونیکی	لینک فایل نمونه
گنجینه سوالات C# (حل مسائل - مرجع کامل)	لینک خرید کتاب چاپی	لینک خرید فایل الکترونیکی	
مفاهیم شی گرای و پیاده سازی آن‌ها با زبان های C# ، جاوا ، C++ و پایتون	لینک خرید کتاب چاپی	لینک خرید فایل الکترونیکی	
مسئله مسیریابی وسایل نقلیه (تئوری و کاربردها)	لینک خرید کتاب چاپی	لینک خرید فایل الکترونیکی	
روسازی راه		لینک خرید فایل الکترونیکی	
آموزش گام به گام برنامه نویسی C# - From Application	لینک خرید کتاب چاپی	لینک خرید فایل الکترونیکی	
آموزش گام به گام برنامه نویسی پایتون	لینک خرید کتاب چاپی	لینک خرید فایل الکترونیکی	لینک دانلود فایل نمونه
حل مسائل پایتون (حل ۶۵۰ برنامه - مرجع کامل)	لینک خرید کتاب چاپی	لینک خرید فایل الکترونیکی	لینک دانلود فایل نمونه
مبانی رایانه و برنامه نویسی به زبان C++	لینک خرید کتاب چاپی	لینک خرید فایل الکترونیکی	لینک دانلود فایل نمونه
۶۰۰ برنامه C++ با حل کامل آنها (حل مسائل - C++ مرجع کامل)	لینک خرید کتاب چاپی	لینک خرید فایل الکترونیکی	لینک دانلود فایل نمونه
حل مسائل جاوا (حل ۶۰۰ برنامه - مرجع کامل)	لینک خرید کتاب چاپی	لینک خرید فایل الکترونیکی	لینک دانلود فایل نمونه
طراحی سیستم های شی گرا با زبان C #	لینک خرید کتاب چاپی	لینک خرید فایل الکترونیکی	لینک دانلود فایل نمونه
۶۵۰ برنامه C # با حل کامل آنها (حل مسائل - مرجع کامل).	لینک خرید کتاب چاپی	لینک خرید فایل الکترونیکی	لینک دانلود فایل نمونه
دانلود کتاب آموزش گام به گام برنامه نویسی بانک اطلاعاتی با C#	لینک خرید کتاب چاپی	لینک خرید فایل الکترونیکی	لینک دانلود فایل نمونه
دانلود کتاب حل مسائل C#	لینک خرید کتاب چاپی	لینک خرید فایل الکترونیکی	لینک دانلود فایل نمونه
گرافیک رایانه ای با زبان برنامه نویسی C#	لینک خرید کتاب چاپی	لینک خرید فایل الکترونیکی	لینک دانلود فایل نمونه
آشنایی با مبانی امنیت شبکه (امنیت اطلاعات)	لینک خرید کتاب چاپی	لینک خرید فایل الکترونیکی	
اصول طراحی پایگاه داده ها	لینک خرید کتاب چاپی	لینک خرید فایل الکترونیکی	لینک دانلود فایل نمونه
حل مسائل C++ (آزمایشگاه کامپیوتر مرجع کامل)	لینک خرید کتاب چاپی	لینک خرید فایل الکترونیکی	لینک دانلود فایل نمونه
آموزش گام به گام LINQ با C#	لینک خرید کتاب چاپی	لینک خرید فایل الکترونیکی	
ساختن داده ها با C++	لینک خرید کتاب چاپی	لینک خرید فایل الکترونیکی	لینک دانلود فایل نمونه
مدیریت استراتژیک فناوری اطلاعات	لینک خرید کتاب چاپی	لینک خرید فایل الکترونیکی	لینک دانلود فایل نمونه
درس و کنکور پایگاه داده پیشرفته	لینک خرید کتاب چاپی	لینک خرید فایل الکترونیکی	لینک دانلود فایل نمونه
فیزیک الکترونیسته	لینک خرید کتاب چاپی	لینک خرید فایل الکترونیکی	
تجارت الکترونیکی	لینک خرید کتاب چاپی	لینک خرید فایل الکترونیکی	لینک دانلود فایل نمونه
راهنمای کاربردی کاربری OPNET برای شبکه های شبیه سازی کامپیوتر	لینک خرید کتاب چاپی	لینک خرید فایل الکترونیکی	
درس و کنکور سیستم عامل پیشرفته	لینک خرید کتاب چاپی	لینک خرید فایل الکترونیکی	لینک دانلود فایل نمونه
شبکه های کامپیوتری با رویکرد کاربردی، آزمایشگاه شبیه سازی شبکه	لینک خرید کتاب چاپی	لینک خرید فایل الکترونیکی	
آزمایشگاه پایگاه داده با ۲۰۱۲ SQL Server	لینک خرید کتاب چاپی	لینک خرید فایل الکترونیکی	لینک دانلود فایل نمونه
کاربرد رایانه در مدیریت و حسابداری	لینک خرید کتاب چاپی	لینک خرید فایل الکترونیکی	لینک دانلود فایل نمونه
آموزش گام به گام برنامه نویسی بانک اطلاعاتی با ویژوال بیسیگنت	لینک خرید کتاب چاپی	لینک خرید فایل الکترونیکی	لینک دانلود فایل نمونه

	لینک خرید فایل الکترونیکی	لینک خرید کتاب چاپی	آموزش گام به گام برنامه نویسی به زبان ++C
	لینک خرید فایل الکترونیکی		حل مسائل پاسکال
لینک دانلود فایل نمونه	لینک خرید فایل الکترونیکی	لینک خرید کتاب چاپی	حل مسائل ++C
	لینک خرید فایل الکترونیکی	لینک خرید کتاب چاپی	دانش ثروتمند شدن
	لینک خرید فایل الکترونیکی	لینک خرید کتاب چاپی	برنامه سازی پیشرفته به زبان #C
	لینک خرید فایل الکترونیکی	لینک خرید کتاب چاپی	الگوریتم ها و محاسبات موازی
	لینک خرید فایل الکترونیکی	لینک خرید کتاب چاپی	دانلود کتاب حل مسائل C
	لینک خرید فایل الکترونیکی	لینک خرید کتاب چاپی	قدرت فکر
	لینک خرید فایل الکترونیکی		راهبردهای الگوریتمی
	لینک خرید فایل الکترونیکی	لینک خرید کتاب چاپی	ارتعاش فکر
	لینک خرید فایل الکترونیکی	لینک خرید کتاب چاپی	طراحی سیستم های تحمل پذیر خطا

فصل ۱ آشنایی با زبان پایتون

۱-۱. مقدمه‌ای بر زبان برنامه نویسی پایتون

پایتون یکی از معدود زبان‌های برنامه‌نویسی است که می‌توان ادعا کرد ساختاری ساده و قدرتمند دارد، از این رو، 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 برنامه‌نویسان 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 قرار گرفته است که نشان‌دهنده‌ی رشد محبوبیت این زبان در میان برنامه‌نویسان 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 در راه یادگیری بگذاریم. بنابراین، برای آشنایی بیش‌تر با زبان پایتون، در ادامه به معرفی برخی از ویژگی‌ها و قابلیت‌های آن می‌پردازیم:

۱. **سادگی و صراحت**^۱: پایتون یک زبان ساده و کمینه‌گرا است. وقتی نگاهی به سورس‌کد یک برنامه‌ی 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 به جای 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.

۲. **منحنی یادگیری کم شیب**^۲: 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 نه تنها خوانایی آن را افزایش داده است، بلکه با حذف پیچیدگی‌ها سهولت یادگیری آن را نیز بیش‌تر کرده است.

^۱. Simplicity

^۲. Low Learning Curve

۳. رایگان و متن باز بودن^۱: 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.
۴. سطح بالا بودن^۲: پایتون از جمله زبان‌های قدرتمند سطح بالا است که برنامه‌نویس را درگیر جزئیات سطح پایین مثل مدیریت حافظه یا کار با ثبات‌ها (Registers) و غیره نمی‌کند.
۵. قابل حمل بودن^۳: ماهیت متن باز پایتون موجب شده است که این زبان با پلتفرم‌های مختلف سازگار باشد. بنابر اعلام رسمی سایت پایتون، در حال حاضر این زبان روی ۲۱ پلتفرم از جمله 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.
۶. زبانی مفسری^۴: برخلاف زبان‌های کامپایلری مانند C یا جاوا، زبان برنامه‌نویسی پایتون یک زبان مفسری است و سورس کد برنامه‌های نوشته‌شده به این زبان با استفاده از یک مفسر اجرا می‌شود که همین موضوع قابل حمل بودن آن را افزایش می‌دهد.
۷. شیء‌گرایی^۵: پایتون در مقایسه با زبان‌هایی مانند جاوا یا ++C، روش قدرتمندتر و ساده‌تری را برای اجرا برنامه‌های شیء‌گرا به کار می‌گیرد.
۸. توسعه پذیری^۶: یکی از مشکلات زبان مفسری پایتون سرعت پایین 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.
۹. تعبیه پذیری^۷: علاوه بر این که 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 نشان داد و به این ترتیب قابلیت‌های اسکریپتی به سورس کد مدنظر اضافه نمود.
۱۰. کتابخانه‌ی گسترده: پایتون از یک کتابخانه‌ی استاندارد غنی بهره می‌برد و در کنار این کتابخانه‌ی وسیع، کتابخانه‌های سایر توسعه‌دهندگان نیز به سرعت در حال توسعه می‌باشند که در مجموع ابزارهای مناسبی را برای ایجاد اسناد، رابط‌های گرافیکی کاربر (GUI)، مرورگرهای وب، رمزنگاری، هوش مصنوعی، پست الکترونیکی، بازی‌سازی، داده کاوی، ایجاد و مدیریت وب‌سایت، و بسیاری کاربردهای دیگر در اختیار برنامه‌نویسان قرار می‌دهد.

^۱. Free & Open Source

^۲. High-level

^۳. Portable

^۴. Interpreted

^۵. Object Oriented

^۶. Extensible

^۷. Embeddable

۱.۱. همه منظوره بودن: پایتون یک زبان برنامه نویسی با طیف گسترده‌ای از کاربردها است که در حوزه‌های مختلف و متنوع کاربرد داشته است که از جمله مهم‌ترین کاربردهای آن در طی سالیان گذشته می‌توان به موارد زیر اشاره کرد:

✚ موتور جستجوگر گوگل و موتور گرافیکی یوتیوب

✚ ساخت برنامه‌های کاربردی علمی در سازمان فضایی ناسا، Fermilab

✚ بخشی از سرویس ایمیل یاهو

✚ تست سخت‌افزار در Cisco، Intel، IBM

✚ ابزارهای نصب لینوکس در نسخه‌ی Redhat

✚ سرویس ابری Dropbox

✚ و بسیاری کاربردهای دیگر نظیر طراحی سایت‌های دینامیک، تولید نرم‌افزارهای دسکتاپ، انیمیشن سازی، بازی سازی، شبکه، امنیت، پایگاه داده، داده کاوی، ساخت برنامه‌های محاسباتی و کاربردی در رشته‌های مختلف نظیر ریاضی، فیزیکی، آمار، زیست و غیره.

در نهایت می‌توان گفت که پایتون ابزاری مهیج و قدرتمند در اختیار برنامه نویسان است که کار با آن ساده و سرگرم کننده می‌باشد و تسلط بر آن کاربران را وارد دنیایی شگفت‌انگیز و بی‌نهایت می‌کند که هر کس می‌تواند متناسب با توانایی‌هایش از امکانات آن برای حل مسائل خود بهره‌مند شود.

۲-۱. مقایسه‌ی زبان برنامه نویسی پایتون با سایر زبان‌های برنامه نویسی

با دانستن این که مقایسه زبان‌های برنامه نویسی با یکدیگر اصلاً کار درستی نیست - چرا که هر زبانی را بهر کاری ساخته‌اند و هر زبان دارای نقاط قوت و ضعف خاص خود است - با این حال، برخی از کاربران همواره دوست دارند تا بدانند زبانی که قرار است فراگیرند در مقایسه با سایر زبان‌های برنامه نویسی هم‌رده‌اش، در چه جایگاهی قرار دارد. بنابراین، در ادامه به مقایسه‌ای کوتاه از زبان پایتون با سایر زبان‌های برنامه نویسی 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 برنامه نویسی 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 به هر حال هر کدام از این دو زبان دارای نقاط قوت و ضعفی هستند). زبان برنامه نویسی پایتون در مقایسه با سی شارپ، از نقاط قوت زیر برخوردار است:

✚ یادگیری آسان‌تر

✚ کد نویسی کم‌تر

✚ متن‌باز و جامعه‌ی توسعه‌ی گسترده

✚ پشتیبانی چندمنظوره بهتر (Multiplatform)

✚ امکان استفاده‌ی راحت از چندین محیط توسعه‌ی نرم‌افزار مختلف

✚ قابلیت توسعه‌ی راحت‌تر با استفاده از زبان‌های C، C++ یا جاوا

✚ پشتیبانی بیش‌تر عملی / مهندسی

۲. **مزیت‌های زبان پایتون نسبت به زبان جاوا:** سالیان درازی را برنامه‌نویسان سراسر دنیا منتظر ماندند تا به زبانی دست یابند که یک‌بار کد نویسی کنند و هرکجا که خواستند آن را اجرا کنند تا این که 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:

✚ یادگیری به‌مراتب راحت‌تر

✚ کد نویسی به‌مراتب کم‌تر

✚ متغیرهایی باقابلیت ذخیره‌سازی انواع داده‌ها

✚ سرعت توسعه‌ی اپلیکیشن به‌مراتب بیش‌تر از جاوا

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

✚ یادگیری سریع‌تر

✚ خوانایی بیش‌تر

✚ تعامل بهتر با زبان جاوا

✚ سازگاری بهتر و بیش‌تر با پلتفرم‌های مختلف

✚ 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 قابل توجهی 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 به عنوان یک برنامه‌نویس مبتدی می‌توانیم بسته به نیازی که برای یادگیری برنامه‌نویسی داریم دست به انتخاب درستی بزنیم.

معمولاً برنامه‌نویسان حرفه‌ای سعی می‌کنند که به بیش از یک زبان برنامه‌نویسی تسلط پیدا کنند تا متناسب با نقاط قوتی که هر زبان برنامه‌نویسی دارد - مثلاً یک زبان 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 - به خصوص زبان برنامه‌نویسی پایتون که در این دوره‌ی آموزشی مدنظر است - به برنامه‌نویسان کمک می‌کند تا با دید بازتری اقدام به استفاده از آن زبان برنامه‌نویسی نمایند. در ادامه قصد داریم به نکاتی پردازیم 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 داده شده است تا دردی از دردهای سایر برنامه‌نویسان 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 به صرف وقت زیاد، سرعت توسعه‌ی کم، راندمان اندک و بسیاری مشکلات دیگر اشاره کرد.

🚩 **نیاز به تعداد خطوط کد کمتر:** سورس کد برنامه‌های نوشته شده 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.

🚩 **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. به عبارت دیگر، برخی از زبان‌های برنامه‌نویسی هستند که مطالعه‌ی سورس کد آن‌ها به مراتب دشوارتر از سایر زبان‌ها است و خبر خوشحال‌کننده این که املا‌ی زبان برنامه‌نویسی پایتون - اگر نگوئیم راحت‌ترین - یکی از راحت‌ترین املاها در میان زبان‌های برنامه‌نویسی است، چرا که تا حد بسیار زیادی شبیه به زبان انگلیسی است!

🚩 **یادگیری سریع:** پیش از این 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 عجیب و غریب 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:

✚ **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 توسعه‌ی اپلیکیشن با این زبان نسبت به سایر رقبا به‌مراتب بیش‌تر است و توسعه‌دهندگان زودتر می‌توانند خروجی کدهای خود را ببینند.

✚ **ساخت وب اپلیکیشن:** اگرچه زبان جاوا اسکریپت به‌عنوان معروف‌ترین زبان برنامه‌نویسی وب اپلیکیشن‌ها 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 اپلیکیشن‌های سریع‌تری طراحی کنند.

✚ **طراحی اپلیکیشن‌های محاسباتی، علمی و مهندسی:** کتابخانه‌های طراحی شده برای پایتون این امکان را به توسعه‌دهندگان می‌دهد تا به‌راحتی و به‌سرعت بتوانند اپلیکیشن‌های محاسباتی، علمی و مهندسی طراحی کنند که از جمله‌ی مهم‌ترین این کتابخانه‌ها می‌توان به NumPy و SciPy اشاره کرد. با بسته NumPy در فصل ۴ آشنا می‌شویم.

✚ **کار با XML:** زبان XML یکی از زبان‌هایی است که برای ذخیره‌سازی داده‌های تحت وب مورد استفاده 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.

✚ **ارتباط با پایگاه داده:** امروزه کسب و کارهای 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 - پردازد.

✚ **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

درعین حال فریم‌ورک‌های بسیاری برای این زبان طراحی شده‌اند 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.

۱. مانند هر زبان طبیعی ابتدا باید علائم تشکیل‌دهنده زبان را آموخت. به‌عنوان مثال، زبان فارسی از علائم الف تا ی، ارقام ۰ تا ۹ و علائم خاص مانند !، ؟ و غیره تشکیل شده است. هر کدام از این علائم (نمادها) مفهوم خاصی دارند. زبان پایتون، نیز از علائم a تا z، A تا Z، 0 تا 9، علائم ویژه نظیر {، }، [،]، / تشکیل شده است. ابتدا، باید مفاهیم هر یک از علائم را در زبان پایتون آموخت.

۲. 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. به‌عنوان مثال، کلمات "بابا"، "آب" و "داد"، در زبان فارسی مفهوم خاصی دارند. ولی کلمات "تپانم" و "بکیاپ" مفهوم خاصی ندارند. به کلماتی که در زبان دارای مفهوم خاص هستند، کلمات کلیدی می‌گویند. در زبان پایتون کلمات کلیدی نظیر if، for، else، while، وجود دارند. در آموزش یک زبان ابتدا باید کلمات کلیدی آن را شناخت و معنی و کاربرد هر کدام از آن‌ها را آموخت.

۳. 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. به‌عنوان مثال، print برای چاپ اطلاعات به کار می‌رود که به‌صورت زیر استفاده می‌گردد:

print (اطلاعات)

۴. 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.

۵. 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 به صورت تعدادی سطر مشاهده می کنند. در پایتون دو نوع سطر وجود دارند. ۱. سطرهای فیزیکی^۱، سطرهایی هستند که توسط ویرایشگرهای متن شماره گذاری می شوند و به سادگی توسط کاربر قابل تشخیص می باشند. ۲. سطرهای منطقی^۲، برداشت مفسر از اجرای برنامه است. هر سطر بیان گر یک دستور پایتون است. به عنوان مثال، دستورات زیر را در نظر بگیرید:

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

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

```
Fanavarienovin.net
```

گاهی اوقات هر سطر فیزیکی می تواند شامل چند سطر منطقی باشد. در این حالت، باید بین سطرها، کاراکتر ”؛“ قرار داد. به عنوان مثال، دستورات زیر را ببینید:

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

با اجرای این دستورات نیز خروجی زیر نمایش داده می شود:

```
Fanavarienovin.net
```

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

^۱. Physical Lines

^۲. Logical Lines

```
>>> 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 "»" استفاده می‌شود. با اجرای این دستورات خروجی زیر نمایش داده می‌شود:

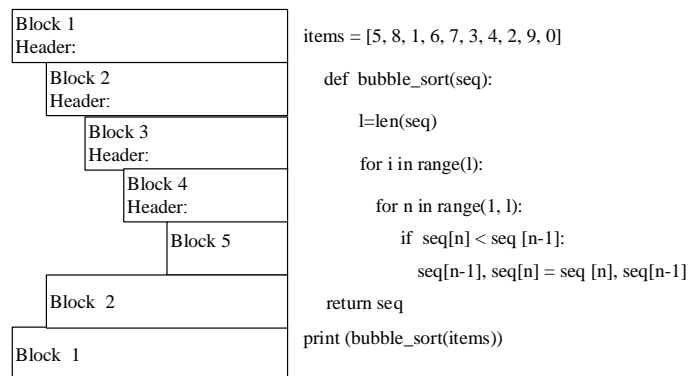
```
Python is a good programing language
```

تعداد کاراکترهای هر سطر فیزیکی نباید از ۷۹ کاراکتر بیش تر شود.

سطرهای خالی^۱، 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^۲ سطرها استفاده می‌شود. در واقع، 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 بلاک‌بندی 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

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. Blank Lines

^۲. Indentation

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

۲. دستورات مرکب؛ 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 چند بخشی هر بخش شامل یک سرآیند^۳ و یک بدنه است. هر سرآیند با یک کلمه کلیدی شروع شده و با ":" خاتمه می‌یابد. همان طوری که قبلاً بیان گردید، بدنه باید پس از سرآیند و با رعایت سطح تورفتگی بیش تر تایپ شود تا علاوه بر افزایش خوانایی برنامه، بلاک‌های هر بخش برنامه را نیز مشخص کنند.

۴-۴-۱. شناسه‌ها

همان طوری 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:

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

در جدول ۱-۲ 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، _y). اما شناسه‌های خصوصی کلاس با دو کاراکتر __ شروع می‌شوند (نظیر شناسه‌های __name، __type و غیره). در ادامه به این شناسه‌ها بیش تر می‌پردازیم.

^۱ Simple Statements

^۲ Compound Statements

^۳ Header

^۴ private

جدول ۱-۲ برخی از نام‌های غیر مجاز	
نام	دلیل
9nine	استفاده از رقم 9 در ابتدای نام غیر مجاز است.
get.length	استفاده از کاراکتر "." در نام شناسه غیر مجاز است.
Var 2	استفاده از کاراکتر فضای خالی در نام شناسه مجاز نیست.
fan@gmail	استفاده از کاراکتر "@" در نام شناسه غیر مجاز است.
\$money	استفاده از کاراکتر "\$" در نام شناسه مجاز نیست.

۵-۴-۱. متغیرها

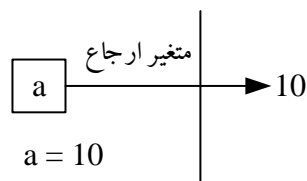
برای نگه‌داری هر ماده‌ی لازم است که یک ظرف استفاده شود. به‌عنوان مثال، در خانه برای نگه‌داری مواد غذایی، ظروف مختلفی وجود دارند که هر کدام برای نگه‌داری مواد خاصی به کار می‌روند. ظرف نگه‌داری داده در برنامه‌نویسی **متغیر** نام دارد. متغیر، نامی است که برای یک مکان از حافظه که ممکن است در طول اجرای برنامه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

۱. نام‌گذاری متغیرها، برای نام‌گذاری متغیر از قوانین نام‌گذاری شناسه‌ها استفاده می‌کنیم.
۲. مقداردهی متغیرها، پس از این 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 شیء مشخص در حافظه اشاره نمی‌کند. 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.

به‌عنوان مثال، دستورات زیر را ببینید:

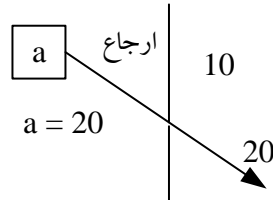
```
>>> a = 10
>>> a = 20
>>> b = a
```

دستور اول، یک شیء از نوع عدد صحیح با مقدار ۱۰ را 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):

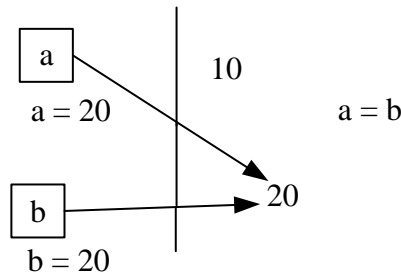


^۱. object

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 به عنوان یک شیء دیگر در مکان جدید تعریف می‌کند و یک پیوند بین شیء a اشاره می‌کند، برای شیء b برقرار می‌کند (مانند شکل زیر):

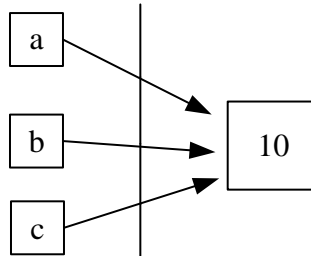


۶-۴-۱. انتساب چندگانه

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. به‌عنوان مثال، دستور زیر را ببینید.

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

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



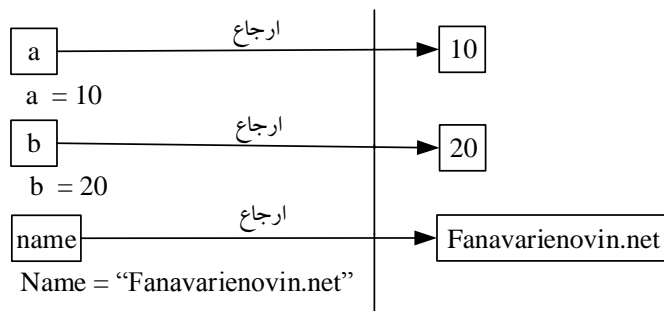
همان‌طور که در این دستور مشاهده کردید، تمام این اشیاء (a ، b و c) به شیء ۱۰ اشاره می‌کنند، اما، گاهی اوقات نیاز است، با یک دستور چند متغیر تعریف کنیم که به چند شیء متفاوت اشاره کنند. برای این منظور، تنها از یک عملگر "=" استفاده می‌شود و قبل از عملگر "="، بین متغیرها، " قرار

آشنایی با زبان پایتون ۲۱

می‌گیرد. بعد از علامت "=" بین اشیاء (مقادیر) نیز، (کاما) قرار می‌گیرد. به عنوان مثال، دستور زیر را ببینید:

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

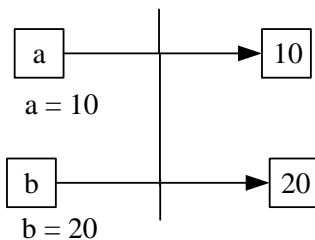
این دستور ابتدا سه شیء به نام‌های ۱۰، ۲۰ و "Fanavarienovin.net" ایجاد می‌کند، سپس سه شیء دیگر به نام‌های a، b و name ایجاد می‌نماید و a را به ۱۰، b را به ۲۰ و name را به "Fanavarienovin.net" پیوند می‌دهد (شکل زیر):



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

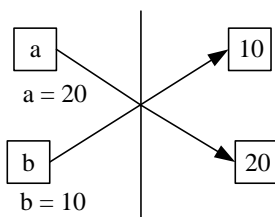
```
>>> a = 10
>>> b = 20
>>> a, b = b, a
```

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 به b اشاره می‌کند را به a و مکانی که a اشاره می‌کند را به b تخصیص

می‌دهد.



۵-۱. عملگرها

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

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

۵-۱-۱. عملگرهای محاسباتی

این عملگرها برای انجام محاسبات بر روی داده‌های عددی به کار می‌روند (جدول ۳-۱)، از جمله 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	$12 + 3$	۱۵	عملوند اول را با عملوند دوم جمع می‌کند.
-	تفریق	$13,5 - 3$	۱۰,۵	عملوند دوم را از عملوند اول کم می‌کند.
*	ضرب	$12 * 2,5$	۳۰	عملوند اول را در عملوند دوم ضرب می‌کند.
/	تقسیم	$13 / 2$	۶,۵	عملوند اول را بر عملوند دوم تقسیم می‌کند.
%	باقی‌مانده تقسیم صحیح	$13 \% 5$	۳	باقی‌مانده تقسیم صحیح عملوند اول بر عملوند دوم را محاسبه می‌کند.
//	تقسیم صحیح	$10 // 3$	۳	تقسیم صحیح عملوند اول بر عملوند دوم را محاسبه می‌کند.
**	توان	$2 ** 9$	۵۱۲	عملوند اول را به توان عملوند دوم می‌رساند.

➦ **عملگر %**، برای محاسبه باقی‌مانده تقسیم صحیح به کار می‌رود. به عنوان مثال، دستورات زیر را

ببینید:

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

در دستور دوم، مقدار متغیرهای x و y دست نخورده باقی می‌مانند.

یعنی، مقدار x و y تغییر نمی‌کند.

➦ **عملگر ****، برای محاسبه توان به کار می‌رود. به عنوان مثال، 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.

^۱.operators

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

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

این دستورات، ۲۳ تقسیم بر ۴ را به صورت گرد شده انجام داده، در z 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. نتیجه 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 نتیجه درست داشته باشند. اما نتیجه عملگر or (یا منطقی) هنگامی نادرست است که هر دو 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 x و true 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 ایجاد شده 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 ۴-۱ 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.

است، وگرنه نتیجه نادرست است.				
اگر عملوند اول کوچک‌تر یا مساوی عملوند دوم باشد، نتیجه درست است، وگرنه نتیجه نادرست خواهد شد.	False	$5 <= 3$	کوچک‌تر یا مساوی	$<=$
اگر عملوند اول مخالف عملوند دوم باشد، نتیجه درست است، وگرنه، نتیجه نادرست خواهد بود.	True	$2 != 5$	نامساوی	$>$ یا $!=$
اگر عملوند اول مساوی عملوند دوم باشد، نتیجه درست است، وگرنه نتیجه نادرست خواهد شد.	False	$2 == 3$	تساوی	$==$

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

جدول ۶-۱ 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). به‌عنوان مثال، دستورات زیر را ببینید:

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

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

پس Z برابر با 2 می‌شود.

آشنایی با زبان پایتون ۲۵

۲. عملگر، "یا" بیتی را انجام می دهد. این عملگر، دو عملوند را بیت به بیت باهم "یا بیتی" نموده (نتیجه یا بیتی زمانی صفر است که هر دو بیت ۰ باشند). به عنوان مثال، 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 \wedge ، xor (یا انحصاری) بیتی را انجام می دهد. این عملگر دو عملوند را بیت به بیت (بیت های متناظر) را باهم یا انحصاری می کند (نتیجه یا انحصاری زمانی یک است که دو بیت مخالف یکدیگر باشند). به عنوان مثال، دستورات زیر را ببینید:

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

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

۴. عملگر \sim ، نقیض بیتی است. این عملگر قبل از یک عملوند قرار گرفته، تمام بیت های 1 آن را به 0 و تمام بیت های 0 را به 1 تبدیل می کند. به عنوان مثال، دستورات زیر را مشاهده کنید:

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

```
a= 00001010
b=11110101
```

۵. عملگر \ll ، شیفت به چپ را انجام می دهد. این عملگر بین دو عملوند قرار گرفته و مقدار عملوند سمت چپ را به تعداد عملوند سمت راست به سمت چپ شیفت می دهد. به عنوان مثال، دستورات زیر را ببینید:

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

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

همان طور که مشاهده می شود، z برابر با ۱۶ است. یعنی، با هر شیفت به چپ، مقدار a در ۲ ضرب می شود و در z قرار می گیرد. پس مقدار a در ۸ ضرب شده ($2^3 * 2$) تا ۱۶ به دست آمده، در z قرار می گیرد.

۶. عملگر \gg ، شیفت به سمت راست را انجام می دهد. این عملگر بین دو عملوند قرار گرفته و مقدار عملوند اول را به تعداد عملوند دوم به سمت راست شیفت می دهد. به عنوان مثال، دستورات را مشاهده کنید:

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

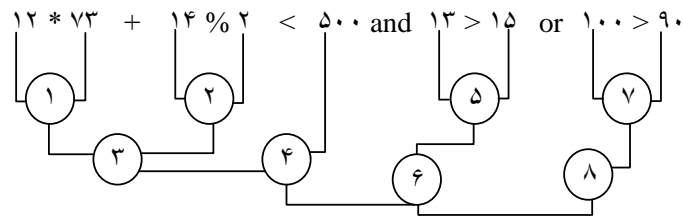
```
a = 00001100
b = 00000011
```

همان طور که در این دستورت مشاهده کردید، با هر شیفت به چپ عدد تقسیم بر ۲ می شود، مقدار ۱۲ تقسیم بر ۴ شده و مقدار ۳ (یعنی، 00000011) به دست آمده است.

۶-۵-۱. اولویت عملگرها

فرض کنید عبارت مقابل را داشته باشید:
 $2 + 3 * 5$
 نتیجه این عبارت می تواند یکی از مقادیر ۲۵ یا ۱۷ باشد. چون اگر عملگر جمع اول انجام شود، نتیجه عبارت برابر $2 + 3$ یعنی ۵ و سپس $5 * 5$ (برابر با ۲۵) خواهد بود. اما، اگر عملگر ضرب اول انجام شود. یعنی $3 * 5$ ابتدا انجام شود، نتیجه برابر با $15 + 2$ (۱۷) خواهد بود. چون، اولویت عملگر ضرب بیش تر از عملگر جمع می باشد، پس نتیجه ۱۷ خواهد شد. اگر عبارت از ترکیبی از عملگرهای مختلف تشکیل شده باشد اولویت عملگرها را طبق جدول ۷-۱ می توان تعیین کرد. یعنی، بالاترین اولویت را عملگر () و پایین ترین اولویت را عملگر and، or و not دارد.

تقدم انجام عملگرها در عبارت زیر را تعیین کنید؟



جدول ۷-۱ اولویت عملگرها	
اولویت	ردیف
()	1
**	2
~	3
+ و - (علامت مثبت و منفی)	4
*, /, %, //	5
+ و - (به علاوه و منها) جمع و تفریق	6
<< و >>	7
&, ^,	8
=, !=, <>	9
>=, >, <=, <	10
+=, -=, %=, /=, *=, **=, =	11
is not, is	12
in not, in	13
not, and, or	14

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

جدول ۸-۱ فرآیند انجام عبارت نمونه.			
مرحل	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

۶-۱. انواع داده‌ها (اشیای آماده)

پایتون هر نوع داده را توسط یک کلاس ارائه می‌کند. بنابراین، هر داده نمونه‌ای^۱ یا یک شی^۲ 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

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

۶-۱-۱. 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:

- داده‌های صحیح با محدودیت اندازه که `int` نامیده می‌شوند.
- 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.

^۱. Instance

^۲. Object