



هنگامی که پایتون را برای توسعه نرم‌افزار انتخاب می‌کنید در واقع یک اکوسیستم بزرگ از زبان برنامه‌نویسی را انتخاب کرده‌اید که تمامی شیوه‌های برنامه‌نویسی مورد نیاز را شامل می‌شود. اما علاوه بر کتابخانه‌ها، برای همه چیز، از توسعه GUI تا یادگیری ماشین، شما می‌توانید از چندین مجری (runtime) پایتون نیز استفاده کنید که بهتر است بدانید کدامیک در چه زمانی کارایی بیشتری دارد. در این مقاله یک نگاه کوتاه به توزیع‌های رایج پایتون مانند پیاده‌سازی استاندارد (CPython)، نسخه‌های بهینه‌شده برای سرعت (PyPy)، برای استفاده خاص (Anaconda، ActivePython) و یا مجری‌هایی برای زبان‌های متفاوت (Jython و IronPython) خواهیم داشت.

سی‌پایتون (CPython)

سی‌پایتون مرجع اصلی پیاده‌سازی پایتون است، نسخه اصلی پایتون که نسخه‌های دیگر از روی آن نوشته شده‌اند. سی‌پایتون همانطور که از اسمش مشخص است در زبان C نوشته شده است و توسط گروه اصلی توسعه دهندگان پایتون ساخته می‌شود.

موارد استفاده سی‌پایتون

از آنجا که سی‌پایتون پیاده‌سازی مرجع پایتون است، از لحاظ بهینه‌سازی، وسواس زیادی روی آن وجود دارد. توسعه‌دهندگان پایتون می‌خواهند سی‌پایتون به عنوان وسیع‌ترین و کامل‌ترین پیاده‌سازی پایتون در دسترس باشد. اگر می‌خواهید برنامه‌های شما بیشترین انطباق را با استانداردهای پایتون داشته باشد سی‌پایتون بهترین انتخاب است. به شما این امکان را می‌دهد که به تمامی جزئیات پایتون دسترسی داشته باشید که البته کمی کار برنامه‌نویسی را مشکل‌تر می‌کند. برای مثال با استفاده از سی‌پایتون برای تنظیمات یک محیط مجازی باید کارهای زیادی انجام دهید، در صورتی که توزیع‌های دیگر پایتون خصوصاً آناکوندا کنترل بهتری بر فضای کاری دارند.

محدودیت‌های سی‌پایتون

سی‌پایتون، بهینه‌سازی عملکردی در نسخه‌های دیگر پایتون را ندارد. هیچ کامپایلر پویایی (just in time) برای آن تعریف نشده است، هیچ کتابخانه‌ایی برای سرعت بخشی به کار وجود ندارد، همچنین هیچ افزونه جانبی برای بالا بردن بازدهی وجود ندارد. این‌ها چیزهایی هستند که می‌توانید خودتان اضافه کنید. با این حال سی‌پایتون به عنوان یک مرجع پیاده‌سازی، بیشترین اطمینان را برای حداکثر سازگاری دارد اما این بدین معنی است که هر بهینه‌سازی عملکردی، باید توسط برنامه‌نویس انجام شود. علاوه بر این، سی‌پایتون تنها مجموعه‌ای از ابزارهای پایه برای کار با پایتون را فراهم می‌کند. به عنوان مثال می‌تواند بسته‌ها را از بسته پایتون (PyPI) دریافت و نصب کند. مدیر بسته‌های پایتون (PIP) حتی می‌تواند بسته‌هایی که توسط توسعه دهندگان ارائه می‌شود را نصب کند ولی هر چیزی که خارج از بسته پایتون باشد را نصب نخواهد کرد.



پایتون در خدمت علم داده
8 نکته کلیدی و مهم پایتون که خیلی از توسعه‌دهندگان اطلاعی از آن ندارند

آناکوندا پایتون (Anaconda Python)

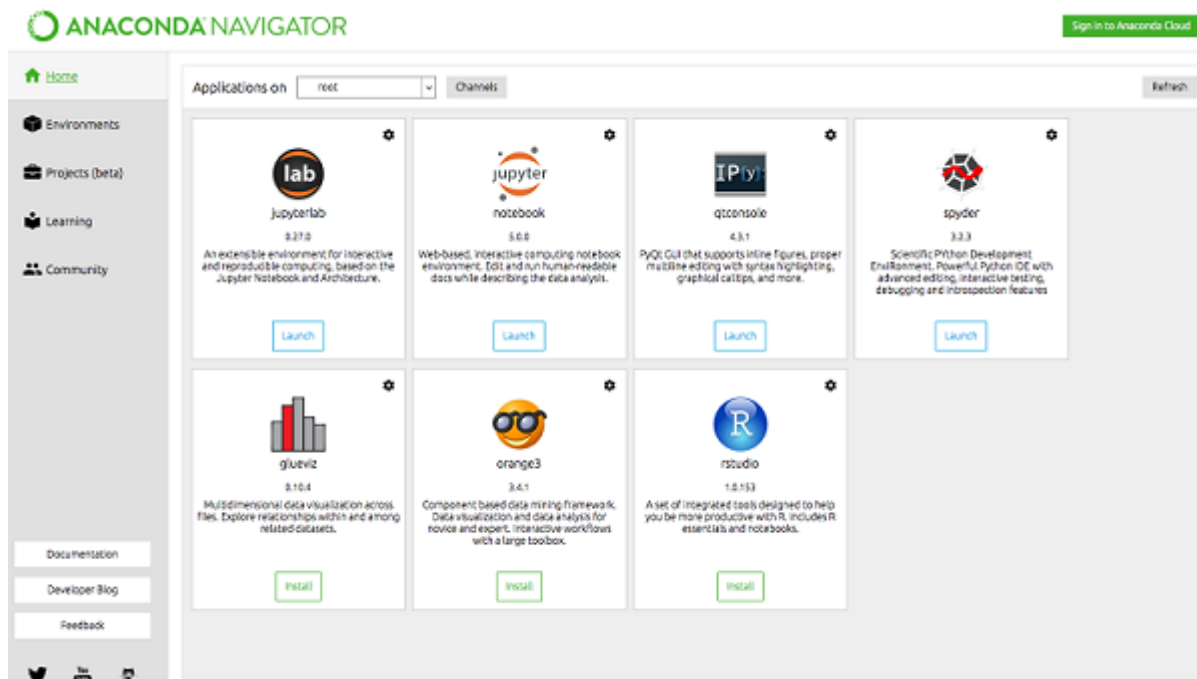
آناکوندا توسط شرکت تجاری آناکوندا برای توسعه دهندگان پایتون طراحی شده که نیاز به یک توزیع از پایتون با پشتیبانی تجاری برای شرکت‌ها دارند. یعنی شرکت‌هایی که از پایتون استفاده تجاری می‌کنند باید بابت هزینه پرداختی پشتیبانی دریافت کنند که این کار توسط شرکت آناکوندا انجام می‌شود. موارد استفاده اصلی برای آناکوندا ریاضی، آمار، مهندسی، تجزیه و تحلیل داده‌ها، یادگیری ماشین و برنامه‌های این چنینی است.

موارد استفاده آناکوندا

شرکت آناکوندا بسیاری از کتابخانه‌های رایج مورد استفاده در کارهای تجاری و علمی مانند Python-SciPy، NumPy، Numba و غیره را می‌سازد و بسیاری از آنها را از طریق یک سیستم مدیریت بسته‌های سفارشی قابل دسترس می‌سازد. آناکوندا در ادغام این قطعات مانند دیگر توزیع‌ها نیست. زمانی که نصب می‌شود یک اپلیکیشن ارائه می‌دهد که کار در محیط آناکوندا را از طریق یک رابط گرافیکی بسیار راحت می‌کند. یافتن اجزای مناسب، به‌روز نگه داشتن آنها و استفاده از آنها با آناکوندا بسیار ساده‌تر از سی‌پایتون است. یکی دیگر از مزایای آناکوندا، اجزای خارج از اکوسیستم پایتون است یعنی زمانی که یک بسته خاص مورد نیاز است. مدیر بسته‌های آناکوندا که conda نامیده می‌شود و به طور خاص برای آناکوندا ایجاد شده است، نصب بسته‌های پایتون و نرم افزارهای جانبی را انجام می‌دهد.

محدودیت‌های آناکوندا

آناکوندا شامل بسیاری از کتابخانه‌های مفید است و می‌تواند حتی با چند کلیک موارد دیگری را نصب کند. به همین دلیل اندازه نصب آناکوندا بسیار بزرگتر از سی‌پایتون است. نصب اولیه سی‌پایتون حدود 100 مگابایت است در حالی که تأسیسات **آناکوندا** می‌توانند به اندازه چند گیگابایت رشد کنند. یکی از راه‌های کاهش اندازه **آناکوندا** این است که مینی‌کوندا را نصب کنید، نسخه‌ای از **آناکوندا** که شامل کمترین قطعات مورد نیاز برای بالا بردن و اجرا **آناکوندا** است. سپس می‌توانید بسته‌های مورد نیاز خود را نصب کنید و با چشم ببینید که هر بسته چه فضایی را اشغال می‌کند.



1000
 00000
 0000000
 00
 000000

اکتیوپایتون (ActivePython)

مانند آناکوندا، اکتیوپایتون نیز توسط یک شرکت تجاری عرضه شده است. اکتیوپایتون می‌تواند چندین زبان مختلف را در محیط یکپارچه Komodo ارائه دهد.

موارد استفاده اکتیوپایتون

اکتیوپایتون بیشتر برای کاربردهای سازمانی و کسانی که با داده‌ها سروکار دارند خوب است، کسانی که دوست دارند از پایتون استفاده کنند ولی نمی‌خواهند وقت زیادی برای نصب و اسمبل کردن آن صرف کنند. اکتیوپایتون از یک نصب ساده و معمول پایتون استفاده می‌کند اما همچنان صدها کتابخانه به صورت بسته‌های آماده نصب دارد و همچنین کتابخانه‌هایی به صورت برنامه‌های جانبی که می‌تواند بعداً اضافه شود.

محدودیت‌های اکتیوپایتون

گاهی اوقات مشکلاتی در نصب و مدیریت پکیج‌های خارجی مشاهده می‌شود. همچنین اگر بخواهید یک نسخه جدیدتر از یک پروژه ایجاد کنید باید اکتیوپایتون را به طور کامل ارتقا دهید؛ یعنی آن را حذف کرده و ورژن جدیدتر را دوباره نصب کنید. در محیط‌هایی که در آن توسعه به یک نسخه خاص از پروژه گره خورده است، این مسئله کمتر دیده می‌شود اما در جاهایی که توسعه‌های آتی در پروژه مد نظر است مشکل ایجاد می‌کند.

پای پای (PyPy)

پای پای یک جایگزین کوچکتر بجای سی‌پایتون به شمار می‌رود. پای پای از یک کامپایلر پویا برای سرعت بخشیدن به برنامه‌های پایتون استفاده می‌کند. استفاده از Just in time می‌تواند سرعت اجرا را به صورت چشمگیری افزایش دهد.

موارد استفاده PyPy

یک شکایت رایج در مورد پایتون (در حالت کلی) و به ویژه سی‌پایتون، سرعت آن است. به صورت پیش‌فرض سرعت پایتون از C بسیار کمتر است اما پای پای با محیط پویای کامپایلر خود به طور میانگین 7.7 برابر سرعت اجرا نسبت به سی‌پایتون دارد. در گاهی اوقات دیده شده این سرعت تا 50 برابر نیز می‌رسد. استفاده از این ابزار نیازی به هیچ کمکی از سوی تیم پایتون ندارد و شما به راحتی می‌توانید برنامه‌های سی‌پایتون را با سرعت بالاتر در پای پای اجرا کنید.

محدودیت‌های PyPy

پای پای همیشه با خود **پایتون** (نه نسخه‌های دیگر) بهترین عملکرد را دارد. این عملکرد در مورد محیط سی پایتون چندان خوب نیست. این مورد توسط توسعه‌دهندگان **پایتون** نادیده گرفته شده است و باعث شده که پای پای بیشتر با بسته‌های خود **پایتون** که به زبان C هستند سازگاری داشته باشد. به طور خلاصه، پشتیبانی از برنامه‌های افزودنی C که جزو کتابخانه‌های **پایتون** نیستند هنوز محدود است، و باعث می‌شود بسیار کمتر از آن استفاده کنند.

دیگر مشکل پای پای اندازه آن در زمان اجرا است. هسته اجرایی **سی پایتون** بدون در نظر گرفتن کتابخانه‌ها حدود 4 مگابایت است و این در مورد پای پای به 32 مگابایت می‌رسد. توجه داشته باشید که **پای پای** به مدت طولانی بر اساس **پایتون** X.2 توسعه یافته است. بنابراین نسخه‌های دیگر مانند X.3 فقط برای ویندوز 32 بیتی و به صورت تست ساخته شده و هنوز به طور کامل عملیاتی نشده است (پای پای در نسخه 64 بیتی برای **پایتون** X.2 و X.3 برای لینوکس و مک موجود است).

جایتون (Jython)

ماشین مجازی جاوا (JVM) مجری بسیاری از زبان‌های دیگر در کنار جاوا نیز هست. مانند Groovy, Scala, Clojure, Kotlin.

موارد استفاده جایتون

جایتون کدهای **پایتون** X.2 را به بایت‌کدهای (bytecode) ماشین مجازی جاوا تبدیل می‌کند و برنامه حاصل در این محیط اجرا می‌شود و این موضوع باعث شده در بعضی موارد جایتون سریعتر از همتای سی پایتون اجرا شود. بزرگترین مزیت جایتون ارائه قابلیت همکاری مستقیم با بقیه اکوسیستم جاوا است. جاوا حتی بیشتر از پایتون استفاده می‌شود. اجرای **پایتون** در ماشین مجازی جاوا به توسعه‌دهندگان پایتون اجازه می‌دهد تا به یک منبع عظیم از کتابخانه‌ها و چارچوب‌هایی که در حالت عادی قادر به استفاده از آن نیستند، وارد شوند. به همین ترتیب، جایتون نیز به توسعه دهندگان جاوا اجازه می‌دهد از کتابخانه‌های پایتون استفاده کنند.

محدودیت‌های جایتون

بزرگترین اشکال جایتون این است که تنها شاخه X.2 **پایتون** را پشتیبانی می‌کند. پشتیبانی از **پایتون** X.3 در حال توسعه است اما تا کنون چیزی منتشر نشده است. نکته دیگر این است که همانطور که **جایتون**، **پایتون** را به ماشین مجازی جاوا تبدیل می‌کند این کدها قابل اجرا در اندروید نیستند. به عبارت دیگر جایتون مناسب برای برنامه‌نویسان اندروید نمی‌باشد.

آیرون پایتون (IronPython)

همانطور که جایتون پیاده‌سازی **پایتون** در محیط مجازی جاوا است، **آیرون پایتون** پیاده‌سازی آن در .NET runtime و یا محیط مجری زبان مشترک (CLR) است. آیرون پایتون از محیط مجری زبان پویا (DLR) که جزئی از مجری زبان مشترک است استفاده می‌کند تا اجازه دهد برنامه‌های **پایتون** با همان پویایی در **سی پایتون** اجرا شوند.

مزایای آیرون پایتون

مثل جایتون، **آیرون پایتون** نیز مانند یک پل عمل می‌کند و استفاده از دنیای بزرگ دات‌نت بزرگترین مزیت **آیرون پایتون** به حساب می‌آید. همچنین می‌توان کدهای **آیرون پایتون** را به زبان‌های دیگر تبدیل و اجرا کرد. با این حال توجه داشته باشید که زبان واسط مایکروسافت (MSIL) نمی‌تواند به صورت مستقیم از دیگر زبان‌های دات‌نت استفاده کند زیرا با مشخصات عمومی زبان سازگار نیست. به عبارت ساده فرضاً نمی‌توانید کد سی‌شارپ را در محیط بیسیک اجرا کنید. اما آیرون پایتون تا حدودی این قابلیت‌ها را دارد.

محدودیت‌های آیرون پایتون

آیرون پایتون نیز در حال حاضر از **پایتون** X.2 پشتیبانی می‌کند. با این حال برای اجرای **پایتون** X.3 کارهایی در حال انجام است.

وین پایتون (WinPython)

همانطور که از نامش برمی آید، **وین پایتون** یک توزیع پایتون است که به طور خاص برای کاربران مایکروسافت ویندوز ایجاد شده است. نسخه‌های پیشین **سی پایتون** برای ویندوز به خوبی طراحی نشده بود و برای کاربران ویندوز دشوار بود از مزایای کامل اکوسیستم پایتون استفاده کنند. نسخه ویندوز **سی پایتون** در طول زمان بهبود یافته است، اما هنوز چیزهای بسیاری در وین پایتون هست که در **سی پایتون** پیدا نمی‌شود.

مطلب پیشنهادی



نام کتاب: مقدمه ای بر برنامه نویسی با پایتون

مزایای وین پایتون

جاذبه اصلی **وین پایتون** این است که این یک نسخه مستقل از پایتون است. بعلاوه نیاز به نصب در دستگاه ندارد و فقط باید از درون پوشه، برنامه را اجرا کنید. به عبارت دیگر این برنامه پرتابل است. این باعث می‌شود که **وین پایتون** در مواردی که نرم‌افزار بر روی یک سیستم مشخص نصب نمی‌شود، مفید باشد و یا زمانی که لازم است چندین نسخه از پایتون همزمان و بدون تداخل اجرا شود. وین پایتون همچنین مجموعه‌ای از بسته‌های پایتون مانند NumPy, Pandas, SciPy, Matplotlib و غیره را شامل می‌شود. بنابراین می‌توان آنها را بدون نیاز به مراحل نصب اضافی بلافاصله مورد استفاده قرار داد. همچنین شامل یک کامپایلر C++/C است، زیرا بسیاری از ماشین‌های اجرایی ویندوز این کامپایلرها را ندارند ولی بسیاری از برنامه‌های پایتون به آنها نیاز دارند و از آن استفاده می‌کنند.

محدودیت‌های وین پایتون

یکی از معایب **وین پایتون** این است که بسیاری از بسته‌ها و کتابخانه‌های آن به طور پیش فرض برای برخی از پروژه‌ها استفاده نشود اما همواره در برنامه **وین پایتون** موجود هستند و حجم برنامه را بالا می‌برند. برای اصلاح آن، سازندگان وین پایتون یک نسخه «صفر» از هر ورژن وین پایتون ارائه می‌دهند که حاوی حداقل داده برای نصب محصول می‌باشد. بعداً می‌توانید بسته‌های بیشتر را با ابزار مدیریت بسته‌های پایتون (PIP) یا ابزار مدیریت بسته‌های وین پایتون (WPPM) اضافه کنید.

منبع:

infoworld

تاریخ انتشار:

27 خرداد 1398

نشانی منبع:

<https://www.shabakeh-mag.com/information-feature/15444/%D9%85%D8%B9%D8%B1%D9%81%DB%8C-%D8%AE%D8%A7%D9%86%D9%88%D8%A7%D8%AF%D9%87-%D9%BE%D8%A7%DB%8C%D8%AA%D9%88%D9%86%E2%80%8C%D9%87%D8%A7-%D8%A2%D9%86%D8%A7%DA%A9%D9%88%D9%86%D8%AF%D8%A7%D8%8C-pypy-%D8%8Cpython-%D9%88-%D8%AF%DB%8C%DA%AF%D8%B1-%D8%AA%D9%88%D8%B2%DB%8C%D8%B9%E2%80%8C%D9%87%D8%A7%DB%8C-%D9%BE%D8%A7%DB%8C%D8%AA%D9%88%D9%86>