

۱۱	فصل اول
۱۳	مقدمه
۱۳	چرا پایتون
۱۴	پیش‌نیازهای لازم برای دنبال کردن این کتاب
۱۵	اصول اولیه پایتون
۱۵	راه‌اندازی محیط
۱۵	آموزش نصب آناکوندا Anaconda
۱۶	استفاده از Jupyter Notebook
۱۸	نحو پایتون
۱۸	Comments
۱۹	متغیرها و انواع داده‌ها
۱۹	اعداد صحیح Integer
۲۰	اعداد ممیز شناور Floating point
۲۰	مقادیر بولی Boolean
۲۰	رشته‌ها Strings
۲۲	لیست Lists
۲۵	لغت‌نامه‌ها (dict)
۲۵	چندتایی‌ها Tuples
۲۶	شرایط و اظهارات
۲۶	دندانه‌گذاری (تورفتگی) در پایتون
۲۷	شروط در پایتون
۲۷	عبارت ELIF و عملگرهای منطقی
۲۸	نماد اختصاری
۲۹	عملگرهای پایتون
۲۹	عملگرهای حسابی (Arithmetic)
۳۰	عملگرهای تخصیص (Assignment)
۳۱	عملگرهای مقایسه (Comparison) و منطقی (Logical)
۳۱	عملگرهای عضویت (Membership)
۳۲	For برای حلقه‌ها loops
۳۲	تابع range ()
۳۳	احکام break و continue

۳۴	توابع
۳۵	محدوده متغیر
۳۷	پارامترهای تابع
۳۸	Unpacking
۳۸	نتیجه‌گیری
۳۸	نام پای <i>NumPy</i>
۳۸	کتابخانه NumPy
۳۹	وارد کردن کتابخانه NumPy
۴۰	ایجاد آرایه‌ها
۴۱	بازیابی ابعاد آرایه و dtype
۴۲	نحوه دسترسی و اصلاح عناصر یک آرایه
۴۲	عملیات ریاضی بین آرایه‌ها
۴۳	برش آرایه
۴۴	پیوند و الحاق آرایه
۴۵	مثال: محاسبه نیروهای داخلی یک تیر با استفاده از NumPy
۴۶	تعریف مسئله:
۴۶	نحوه وارد کردن کمیت‌های مسئله در پایتون
۴۷	محاسبه لنگر خمشی و برش
۴۹	نتیجه‌گیری
۴۹	سم پای (<i>SymPy</i>)
۴۹	سیستم‌های جبری رایانه‌ای
۴۹	وارد کردن کتابخانه
۵۰	نمادها
۵۱	دستور <i>subs()</i> و ارزیابی عددی
۵۲	حسابان
۵۲	حل‌کننده‌ها
۵۲	معادلات
۵۳	دستور <i>solveset</i>
۵۴	سیستم معادلات: <i>linsolve</i> و <i>nonlinsolve</i>
۵۴	پانداس (<i>Pandas</i>)

۵۵	چارچوب‌های داده Dataframes
۵۵	ایجاد یک Dataframes
۵۶	ایجاد یک Dataframe از آرایه NumPy
۵۶	تغییر شکل داده‌های یک dataframe
۵۶	sort_values
۵۷	rename
۵۷	drop
۵۸	concat
۵۹	استخراج داده از یک Dataframe
۶۰	iloc و loc
۶۰	ایجاد یک dataframe با هدرهای پیچیده
۶۱	انجام عملیات با dataframe ها
۶۲	کار با داده‌های خارجی
۶۲	بارگیری یک صفحه گسترده
۶۴	بارگذاری یک فایل csv
۶۵	صدور خروجی یک dataframe
۶۶	مثال: ترکیب بار برای یک تیر دو دهانه
۶۶	تعریف مسئله:
۷۰	ترسیم نتایج
۷۱	مت‌پلات (Matplotlib)
۷۱	بارگیری کتابخانه و وارد کردن داده‌ها
۷۲	Matplotlib چگونه کار می‌کند
۷۶	تغییر ظاهر یک نمودار
۷۶	تغییر رنگ و سبک خطوط
۷۶	ترسیم بیش از یک خط
۷۷	سبک‌های ترسیم
۷۸	پرکردن نواحی
۷۹	اضافه کردن نشانگرها
۸۰	ترسیم چند نمودار
۸۲	اصلاح علامت‌های tick mark

۸۲	 نمودارهای پراکندگی Scatter plots
۸۵	 نمودارهای ستونی یا میله‌ای Bar Plot
۸۸	 پیوست: مراجع پارامتر
۸۸	 رنگ‌های نام‌گذاری شده
۸۸	 Linetypes انواع خطوط
۸۹	 Markers نشانگرها
۹۰	 سبک‌های نمودار
۹۶	 نگاشت رنگ
۱۰۱	 فصل دوم
۱۰۳	 محاسبه خواص مقاطع
۱۰۳	 ویژگی‌های مقطع تیر فولادی
۱۰۸	 نتیجه‌گیری
۱۰۹	 اندرکنش $M-N$ در یک مقطع بتنی
۱۰۹	 Ultimate limit state (ULS) حالت حدی نهایی
۱۱۰	 مصالح و هندسه
۱۱۱	 میل‌گرد فولادی
۱۱۲	 دامن‌های کرنش
۱۱۳	 کد پایتون
۱۱۵	 نمودار تنش - کرنش بتن
۱۱۵	 نمودار تنش-کرنش فولاد
۱۱۶	 I دامن
۱۱۸	 II دامن
۱۲۱	 III دامن
۱۲۲	 IV دامن
۱۲۳	 V دامن
۱۲۴	 ترسیم نتایج
۱۲۵	 نتیجه‌گیری
۱۲۵	 پیوست: ذخیره کد در یک فایل خارجی
۱۳۰	 طراحی تیر بتنی
۱۳۱	 ابعاد و بارها

۱۳۱	واردات کتابخانه‌های لازم
۱۳۲	محاسبه توزیع‌های پوش
۱۳۳	استخراج حداکثر مقادیر از توزیع‌های پوش
۱۳۶	لنگر خمشی مقاوم مقاطع
۱۳۸	کنترل مقطع
۱۳۸	کنترل برش
۱۴۰	نتیجه‌گیری
۱۴۰	طراحی ستون فولادی
۱۴۲	مصالح
۱۴۲	سطح مقطع
۱۴۳	کمانش ستون
۱۴۷	کنترل اعضا
۱۴۸	نتیجه‌گیری
۱۴۸	صدور خروجی در <i>LaTeX</i>
۱۴۸	نصب افزونه nbextension

مقدمه

این کتاب برای چه کسانی نوشته شده است؟ به طور کلی، این کتاب برای هر کسی که مایل به یادگیری یک زبان برنامه نویسی مبتنی بر اصول پنج گانه *SOLID* و استفاده از آن در برنامه های عمرانی و مکانیکی است، هدف قرار گرفته است. با این حال، حتی اگر از رشته تحصیلی دیگری آمده باشید، در اینجا تکنیک های بین رشته ای بسیار مفیدی را خواهید یافت که می توانید در زمینه تخصصی خاص خود به کار ببرید. هر کاری که در اکسل انجام می دهید را می توان در پایتون و به روشی بسیار کارآمدتر انجام داد. اگر تا به حال یک تیر بتنی با استفاده از اکسل طراحی کرده اید، می دانید که صفحات گسترده می توانند در مدت زمان بسیار کوتاهی کاملاً شلوغ شوند. اساساً، اگر در یکی از این دسته بندی ها قرار می گیرید، این کتاب برای شما مناسب است:

- می خواهید از اکسل به چیزی سریع تر و انعطاف پذیرتر کوچ کنید.
- شما می خواهید یاد بگیرید که چگونه از پایتون برای برنامه های کاربردی مهندسی استفاده کنید.
- شما یک دانشجو هستید که به دنبال مجموعه ای از مثال های واضح در طراحی المان های سازه ای می گردید.
- شما می خواهید از پایتون در محیط کاری خود استفاده کنید، اما همچنان می خواهید اسناد و جداول اکسل را با همکاران خود به اشتراک بگذارید.
- شما می خواهید یاد بگیرید که چگونه به طور خودکار گزارش هایی با فرمت حرفه ای از محاسبات خود ایجاد کنید و آنها را در قالب *LaTeX* یا *pdf* صادر کنید.
- شما می خواهید یاد بگیرید که چگونه گراف ها و نمودارهای زیبا ترسیم کنید

چرا پایتون

پایتون یک زبان برنامه نویسی همه کاره و قدرتمند است. منبع باز و رایگان برای همه است و دارای یک جامعه بزرگ است که به طور مداوم ویژگی های جدید را در قالب کتابخانه ها اضافه می کند. این بدان معنی

است که پایتون مجموعه وسیعی از ابزارها را ارائه می‌دهد که دائماً در حال به‌روزرسانی هستند و بدون اشکال نگه داشته می‌شوند. اما مهم‌تر از همه، باوجود اینکه یک زبان برنامه‌نویسی کامل است، استفاده از آن واقعاً آسان و نحو سریع و کارآمد است، بنابراین برای دستیابی به کارهای پیچیده به چند خط کد نیاز دارید. در نهایت، یکی از مزیت‌های بزرگ پایتون، دفتر یادداشت ژوپایتر *Jupyter notebooks* است.

Jupyter محیط برنامه‌نویسی است که در سراسر این کتاب مورد استفاده قرار خواهد گرفت و یک نیروگاه واقعی است: با آن می‌توانید کد نوشتاری، جداول و تصاویر را به‌صورت متناوب بین بلوک‌های متن و متن قالب‌بندی شده به نمایش بگذارید و خیلی چیزهای دیگر. باین‌حال، تنها زمانی که فرد شروع به استفاده از زبان می‌کند، گستره واقعی برنامه‌های کاربردی آشکار می‌شود. با خواندن این کتاب متوجه خواهید شد که در کجا می‌توانید آموخته‌های خود را اعمال کنید و شروع به ایجاد برنامه‌های کاربردی خود کنید.

پیش‌نیازهای لازم برای دنبال کردن این کتاب

آنچه شما باید از طریق مثال‌های ارائه شده در فصل‌های بعدی کاملاً دنبال کنید این است:

چیزی که در فصل‌های بعدی به آن نیاز خواهید داشت این است که:

- کاربرد ابتدایی ویندوز و میکروسافت را بدانید.
- دانستن حساب دیفرانسیل و انتگرال سطح دبیرستان و جبر خطی.
- دانش ابتدایی نرم افزار *Excel*.

به‌علاوه، اگر حوزه‌های مرتبط با مهندسی زیر را بشناسید، قطعاً کمک خواهد کرد:

- مکانیک جامدات
- تئوری سازه‌ها
- تحلیل و طراحی سازه

این‌ها اکیداً ضروری نیستند، زیرا من سعی خواهم کرد تا حد امکان مطالب فنی را به‌وضوح توضیح دهم، اما مثال‌های این کتاب به‌شدت بر این موضوعات تکیه دارند.

اصول اولیه پایتون

راه اندازی محیط

قسمت اول این کتاب یک دوره آموزش سریع پایتون در نظر گرفته شده است: ابتدا تمام نرم افزارهای مورد نیاز را نصب می کنیم، سپس به سراغ اصول برنامه نویسی می رویم؛ بنابراین بسیار مهم است که همه چیزهایی که در اینجا نوشته شده است را دنبال کرده و درک کنید.

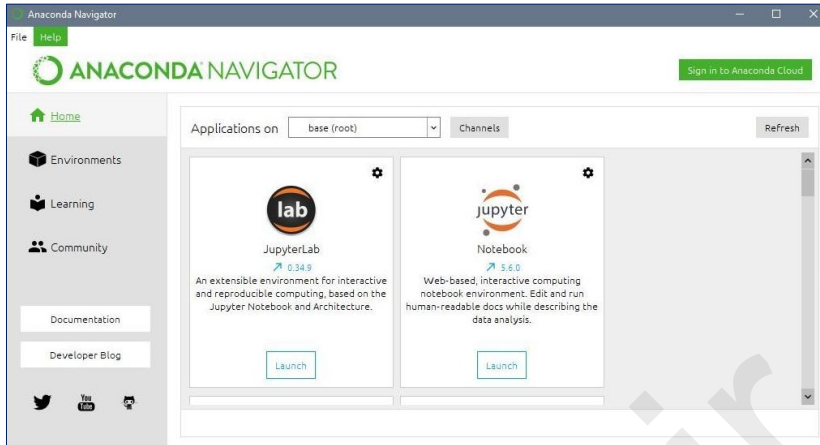
آموزش نصب آناکوندا *Anaconda*

یکی از ساده ترین راه ها برای راه اندازی و اجرای پایتون در رایانه، استفاده از توزیع آناکوندا است. آناکوندا با محبوب ترین کتابخانه های پایتون (بنابراین نیازی به نصب آن ها نخواهید داشت) و همچنین *Jupyter* که یک محیط *notebook* است که از آن برای نوشتن کد استفاده می کنیم. دستورالعمل های ارائه شده در اینجا برای مایکروسافت ویندوز در نظر گرفته شده است، اما فرایند کلی برای سایر سیستم عامل ها نیز صادق است.

به منظور نصب آناکوندا:

- مرورگر وب خود را باز کنید و به <https://www.anaconda.com/download> بروید و سیستم عامل خود را انتخاب کنید.
- نسخه ای از پایتون را که می خواهید دانلود کنید انتخاب کنید.
- در زمان نگارش این کتاب، آخرین نسخه پشتیبانی شده توسط آناکوندا پایتون ۳.۷ بود. نسخه ای را که از پایتون ۲ استفاده می کند دانلود نکنید، زیرا برخی از مثال های نوشته شده در این کتاب فقط با پایتون ۳ کار می کنند.
- هنگامی که دانلود را به پایان رساند، آن را باز کنید و دستورالعمل نصب آناکوندا را دنبال کنید. پیشنهاد می کنم مسیر پیش فرض را به عنوان پوشه مقصد *path* بگذارید.

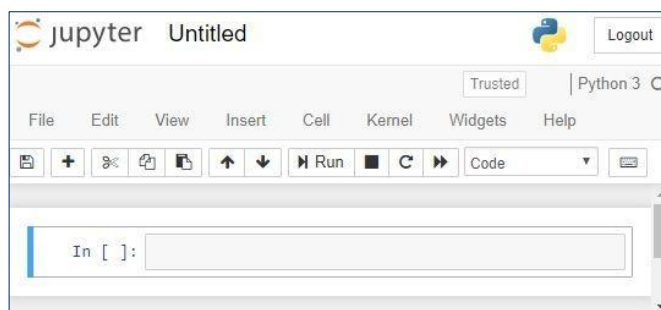
پس از اتمام نصب آناکوندا، ناور آناکوندا *Anaconda Navigator* را با تایپ کردن "*navigator Anaconda*" در نوار جستجوی رایانه خود باز کنید. با پنجره ای روبرو خواهید شد که شبیه به تصویر زیر است:



ناوبر آناکوندا اساساً یک *hub* است که از آنجا می‌توانید به محیط‌های مختلف کدنویسی پایتون دسترسی داشته باشید. کنسول *Qt* یک ویرایشگر خط فرمان است، *Spyder* یک ویرایشگر اسکریپت است و در نهایت *Jupyter* یک ویرایشگر نوت‌بوک است. این همان چیزی است که ما قرار است از آن استفاده کنیم، بنابراین روی *launch* کلیک کنید تا *Jupyter Notebook* باز شود. با این کار مرورگر پیش‌فرض شما باز می‌شود و درخت پوشه رایانه شما نمایش داده می‌شود. در نوار آدرس باید `http://localhost:8888/tree` نوشته شود. از این پنجره می‌توانید در داخل پوشه‌های رایانه خود حرکت کنید، *Jupyter Notebook* موجود را باز کنید، و پوشه‌های جدیدی ایجاد کنید.

استفاده از *Jupyter Notebook*

برای ایجاد یک *notebook* ژوپیتر *Jupyter Notebook* جدید، به پوشه دلخواه خود بروید و از سمت راست‌بالای صفحه، `new>python3` را انتخاب کنید. با این کار یک *notebook* ژوپیتر جدید در پوشه‌ای که انتخاب کرده‌اید ایجاد می‌کند. سپس مرورگر شما باید چیزی شبیه به آنچه در تصویر بعدی می‌بینید نمایش دهد:



Jupyter یک محیط *notebook* است، به این معنی که شما باید کد را در داخل سلول‌ها بنویسید. کادر در تصویر قبلی که عبارت "*In*" را نشان می‌دهد یک سلول است. اگر کادر با رنگ آبی مشخص شده باشد، به این معنی است که سلول انتخاب شده و آماده ویرایش است. قبل از شروع نوشتن کد، بیایید ابزارها و منوهای کشویی مختلفی را که *Jupyter* ارائه می‌دهد مرور کنیم تا شما ایده روشنی از نحوه عملکرد رابط کاربری داشته باشید.

- از منوی *File* می‌توانید *notebook*‌ها را باز، ذخیره، تغییر نام، صادر و ببندید.
- از منوی *Edit* می‌توانید سلول‌ها را کپی، برش، ادغام، انتقال و حذف کنید.
- از منوی *View* می‌توانید ظاهر رابط کاربری را سفارشی کنید.
- از منوی *Insert* می‌توانید یک سلول جدید در بالا یا پایین سلولی که در حال حاضر انتخاب شده است وارد کنید.
- از منوی *Cell* می‌توانید گزینه‌های مختلفی را برای اجرای سلول‌ها در *notebook* انتخاب کنید. به عنوان مثال می‌توانید یک سلول یا کل *notebook* را اجرا کنید. علاوه بر این، از این منو می‌توانید نوع سلول را تغییر دهید.
- از منوی *Kernel* می‌توانید هسته *notebook* را مدیریت کنید. هسته اساساً موتور محاسباتی است که کدهای نوشته شده در سلول‌ها را اجرا می‌کند. همچنین تمام متغیرهایی را که در کد خود اعلام می‌کنید ذخیره می‌کند.
- از منوی *Widget* می‌توانید ویجت‌های *notebook* را مدیریت کنید که در اصل افزونه‌هایی هستند که می‌توانند برای گسترش قابلیت‌های *Jupyter* نصب شوند.
- از منوی *Help* می‌توانید یک تور رابط کاربری را شروع کنید (من پیشنهاد می‌کنم آن را امتحان کنید)، کتابچه راهنمای مرجع مختلف را باز کنید و پنجره میان‌بر را نمایش دهید.

نمادهای زیر منو به شما امکان می‌دهد برخی از ابزارهای موجود در منوهای کشویی که در بالا توضیح داده شد را به سرعت انتخاب کنید. بیایید توجه خود را بر روی پنجره کشویی کوچکی متمرکز کنیم که می‌گوید "*code*": از اینجا می‌توانیم نوع سلول را از کد به *markdown* تغییر دهیم.

Markdown یک زبان تایپ است که برای فرمت کردن نوت‌بوک شما استفاده می‌شود. معمولاً یک *notebook* به خوبی فرمت شده از هر دو سلول نوع کد و سلول‌های نوع *Markdown* ساخته می‌شود. اولی شامل تمام الگوریتم‌ها و محاسبات مختلف است، دومی حاوی متنی است که توضیح می‌دهد کد چه کار می‌کند، یا نتیجه سلول‌های *{code}* را نشان می‌دهد. بیایید فعلاً نوع سلول را به *{code}* بگذاریم و داخل کادر سلول کلیک کنیم. قاب سبز می‌شود: اکنون می‌توانیم کدی در داخل آن بنویسیم. بیایید بنویسیم:

```
1 print("Hello World!")
```

```
Hello World!
```

و روی دکمه *Run* کلیک کنید. تبریک می‌گویم، شما یک برنامه پایتون نوشته‌اید! توجه داشته باشید که خروجی سلول در زیر کد نمایش داده می‌شود. یک سلول لازم نیست یک خروجی را نمایش دهد: اگر دستورالعملی برای این کار در کد وجود ندارد، سلول به‌سادگی بدون نشان‌دادن چیزی اجرا خواهد شد. اکنون که مثال کلاسیک *hello world* را از سر راه برداشتیم، می‌توانیم کارکردهایی را که پایتون به کاربرانش ارائه می‌دهد را بررسی کنیم. در چند فصل بعدی، نحوه عملکرد نحو زبان و نحوه استفاده از برخی از کتابخانه‌های موجود را خواهید آموخت. مهم است، اگر تابه‌حال از کتابخانه‌ها استفاده نکرده‌اید، مثال‌های داده شده را دنبال کنید. این فصل تمام ابزارهای لازم برای درک کد را در بخش‌های مرتبط با مهندسی بعدی کتاب در اختیار شما قرار می‌دهد.

توجه داشته باشید

از این پس، زمانی که می‌خواهید یکی از نمونه‌های ارائه شده در این کتاب را اجرا کنید، به‌سادگی کد را در یک سلول خالی بنویسید و آن را اجرا کنید. من پیشنهاد می‌کنم نمونه‌ها را خودتان بنویسید، زیرا اگر می‌خواهید بفهمید چگونه کار می‌کنند، *copy* و *paste* کردن آن‌ها چندان مؤثر نخواهد بود.

نحو پایتون

اکنون به بررسی عناصر اصلی تشکیل‌دهنده زبان پایتون می‌پردازیم.

Comments

نظرات یا *Comments* خطوطی از کد هستند که پس از اجرا برنامه نادیده گرفته می‌شوند. از آنها برای اظهارنظر و توضیح آنچه که کد انجام می‌دهد استفاده می‌شود تا همه چیز تا حد امکان واضح باشد. سلول کد بعدی نحوه استفاده از آنها را نشان می‌دهد:

```
1 # This is a single line comment
2
3 '''This is a
4 multiline comment'''
```

همان‌طور که می‌بینید پایتون دو روش برای نوشتن نظر ارائه می‌دهد. اولی از نماد `#` برای نوشتن یک نظر تک خطی استفاده می‌کند، دومی از توالی کاراکترهای `'''` برای باز و بسته کردن یک توضیح چند خطی استفاده می‌کند.

متغیرها و انواع داده‌ها

متغیرها محفظه‌هایی هستند که می‌توانید مقادیر را در آنها ذخیره کنید. انواع اصلی ساختارهای داده‌ای که در پایتون تعبیه شده‌اند (و می‌توانند بدون بارگذاری هیچ کتابخانه‌ای مورد استفاده قرار گیرند) عبارت‌اند از:

- `int` : اعداد صحیح
 - `float` : اعداد ممیز شناور
 - `bool` : مقادیر از نوع بولی (درست یا غلط)
 - `string` : دنباله‌ای از کاراکترها
 - `list` : آرایه‌های داخلی پایتون.
 - `dict` : لغت‌نامه‌ها
 - `Tuple` : چندتایی
- بیاید آنها را با جزئیات مرور کنیم.

اعداد صحیح *Integer*

این نوع داده نشان‌دهنده اعداد صحیح است، یعنی اعداد بدون مقادیر اعشاری. یک سلول کد خالی را انتخاب کنید و خطوط داده شده در قطعه کد بعدی را بنویسید:

```
1 a=1
2 b=2
3 c=a+b
4 print(c)

3
```

مقادیر `1$` و `2$` را به متغیرهای `a` و `b` نسبت می‌دهیم سپس آنها را جمع کرده و نتیجه را در متغیر `c` ذخیره می‌کنیم و در نهایت مقدار `c` را چاپ می‌کنیم. `a` و `b` مقادیر صحیح هستند و از آنجایی که `c` مجموع دو شیء از نوع `int` است، نیز یک عدد صحیح است. همان‌طور که ممکن است حدس بزنید، تابع چاپ برای نمایش مقدار هر آرگومانی که بین پرانتز قرار می‌دهید استفاده می‌شود.

توجه داشته باشید

جمع، تفریق و ضرب اعداد صحیح همیشه به یک نتیجه از نوع `int` منجر می‌شود. با این حال، تقسیم بین اعداد صحیح همیشه منجر به یک مقدار ممیز شناور می‌شود.